

**A parsing explanation of linear movement asymmetry
in natural language**

by

Carl G. Alphonse

B.Sc., McGill University, 1988

B.A., McGill University, 1990

M.Sc., University of British Columbia, 1992

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE STUDIES

(Department of Computer Science)

We accept this thesis as conforming
to the required standard

The University of British Columbia

April 2000

© Carl G. Alphonse, 2000

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of COMPUTER SCIENCE

The University of British Columbia
Vancouver, Canada

Date April 25, 2000

Abstract

Natural language sentences are made up of words. Groups of words which can act together as a unit are called constituents. *Movement* is a syntactic process which displaces a constituent from its canonical position. The moved constituent is called a *filler*. The position from which a filler is displaced is termed a *gap*. A gap is a particular type of *empty category*.

Interestingly, a filler always appears to the left of its associated gap. My thesis is that this linearity constraint on filler-gap dependencies is not a syntactic constraint but rather a product of the parsing mechanism.

I develop a parsing algorithm which can process filler-gap dependencies only if the filler is identified before the gap. What makes this possible in general is that I treat the problem of licensing and identification of empty categories during on-line processing as a case of ambiguity resolution. I develop a fine-grained typology of nominals (both overt and empty) which supports the use of underspecified representations. In turn, this permits the parser to resolve empty category ambiguities incrementally as parsing proceeds.

The main results of the dissertation are as follows. The parsing algorithm presented herein offers an explanation for the curious fact that overt movement is leftward. It also explains why, in a language like Italian, postverbal DPs are preferentially interpreted as objects rather than subjects when they appear after optionally transitive verbs. There are also several predictions which arise from this work. The operation of the algorithm predicts that a prenominal relative clause is possible only in a language which permits personal pronouns to act as relative pronouns. There is also predicted to be a filled-gap effect with structures (such as left dislocation in English) for which there is an ambiguity as to whether movement has taken place. Finally, this research supports the view that the human language processing mechanism constructs a single underspecified representation, and resolves ambiguities incrementally as the parse progresses.

Contents

Abstract	ii
Contents	iii
List of Figures	viii
Acknowledgements	ix
Dedication	x
1 Introduction	1
1.1 Displacement	1
1.2 Movement	3
1.3 Overview of dissertation	5
2 Background	7
2.1 Introduction	7
2.2 Syntax	9
2.2.1 Fundamental concepts	10
2.2.2 Projections and extended projections	14
2.2.3 Principle of full interpretation	20

2.2.4	Thematic roles	20
2.2.5	Case marking	23
2.2.6	The basics of movement	26
2.2.7	Operators and A-bar movement	28
2.2.8	Head movement	30
2.2.9	Binding theory and empty categories	31
2.2.10	Properties of movement	33
2.3	Psycholinguistic background and assumptions	35
2.3.1	Empirical results from parsing experiments	36
2.3.2	Theories of filler-gap processing	40
2.4	A licensing parsing algorithm	44
2.4.1	Determinism	44
2.4.2	Licensing and parsing	48
3	The problem	58
3.1	Introduction	58
3.2	Refuting rightward movement	60
3.2.1	Right node raising	60
3.2.2	Extraposition	62
3.2.3	Heavy NP Shift	63
3.3	Stating thesis	69
4	Possible solutions	70
4.1	Constraining the movement operation	71
4.2	Constraining the target of movement	72
4.2.1	Context free grammars	73

4.2.2	\bar{X} theory	75
4.2.3	ID/LP rules	77
4.2.4	Summary	81
4.3	Constraining the structure to order mapping	81
4.3.1	Representing word order differences	82
4.3.2	Deriving different word orders	84
4.4	Constraining the processing of movement	88
4.4.1	A gap-driven model	89
4.4.2	The Active Filler Strategy	91
4.4.3	Summary	92
5	A parsing solution	94
5.1	Nominal subcategories	95
5.1.1	θ marking	97
5.1.2	Case marking	99
5.1.3	Operator status	100
5.1.4	A typology of nominal chains	114
5.2	Parsing revisited	119
5.2.1	Underspecification and augmentation	119
5.2.2	Attachment	125
5.2.3	Extension	129
5.2.4	Gap-driven processing	132
5.3	Processing empty categories	140
5.3.1	Creation of trace empty categories	141
5.3.2	Postulation of chain-heading empty categories	141
5.3.3	Licensing features and empty categories	142

5.4	The parsing algorithm and an implementation	143
5.4.1	The algorithm	143
5.4.2	An implementation	143
5.5	An example	147
5.6	Processing vs. grammatical exclusion	154
6	Testing the thesis	156
6.1	Overt movements	158
6.1.1	Overt <i>wh</i> movement	158
6.1.2	Overt relative pronoun	172
6.1.3	Topicalization	179
6.2	Covert movements	182
6.2.1	Licensed in specifier of C position	183
6.2.2	Licensed in argument position	190
6.3	Impossible movements	199
6.4	Unexplained phenomena	206
6.5	Summary	207
7	Consequences	209
7.1	Post-verbal subject interpretations	210
7.2	Relative clauses	222
7.3	Non-movements	233
7.4	Left dislocation	236
7.5	Garden paths and <i>culs-de-sac</i>	238
8	Coda	242
8.1	Summary	242

8.2	Kayne's Linear Correspondence Axiom (LCA)	244
8.3	Hawkins' Early Immediate Constituents Principle	247
8.4	Future work	249
8.4.1	Linguistic issues	249
8.4.2	Psycholinguistic issues	255
8.4.3	Computational issues	255
Bibliography		257
Index		265

List of Figures

2.1	The relationship of competence and performance.	8
2.2	An idealized example of priming	38
2.3	A simple licensing parsing algorithm	50
5.1	The parser's view of chain-heading empty categories	124
5.2	The final version of the algorithm	144
5.3	The relationship of my contribution (inside dashed line) to a standard licensing parser	145

Acknowledgements

I gratefully acknowledge the contributions of my committee members: David Poole, Henry Davis, Richard Rosenberg and Michael Rochemont. They were all instrumental in guiding me along my journey. They also all endured more than their fair share of misguided questions and ill conceived ideas with great aplomb. While I have learned a great deal from each of them, I wish now that I had paid better attention: they taught me more than I learned.

I also wish to thank Hamida Demirdache, Rose-Marie Déchaine and Martina Wiltschko for sharing their linguistic wisdom, and the UBC Computer Science department, UBC and NSERC for financial support. Lyn Frazier provided invaluable feedback on psycholinguistic issues.

I am thankful to my parents for having instilled in me a desire to learn, a love of reading, and for teaching me to say “please” and “thank-you”.

My dear, sweet, loving wife Averill and daughter Olivia have been infinitely understanding and supportive throughout my graduate work, and our new addition Dahlia, who arrived less than two weeks after the defense, encouraged me with big smiles throughout the revisions. I cannot imagine having undertaken, let alone completed, this dissertation without them.

CARL G. ALPHONCE

*The University of British Columbia
April 2000*

To my family: Averill, Olivia and Dahlia.

Chapter 1

Introduction

1.1 Displacement

A property of natural language is that elements,

... appear in the sensory input in positions “displaced” from those in which they are interpreted, under the most principled assumptions about interpretation. This is an irreducible fact about human language, expressed somehow in every contemporary theory of language, however the facts about displacement may be formulated.

(Chomsky 1995, p. 222)

For example, in (1) below the word *what* appears sentence initially. *What* is interpreted as the object of the verb *buy*. The canonical object position is indicated by “___”. In this example, *what* is displaced from the canonical object position.

- (1) What did Ardelia buy ___?

A language may make use of many different mechanisms to effect the displacement of a constituent from one position to another. Each mechanism may

impart different properties on the displaced constituent. Let D refer to the set of all cases of displacement which occur in a given language, regardless of the mechanism used to effect the displacement. The cluster of properties listed in (2a–f), which are discussed in chapter 2, defines a subset M of D .

- (2) M is the set of those displacements that:
- a. involves a syntactic constituent
 - b. exhibits island effects
 - c. obeys the proper binding condition
 - d. displays empty category principle effects
 - e. can be long-range
 - f. licenses parasitic gaps

These properties, which are syntactic (structural) in nature, make no reference to linear order. Nonetheless, this cluster of properties correlates systematically with leftward displacement. All known cases of rightward displacement have been shown to lack this cluster of properties. In other words,¹

- (3) a. $x \in M \Rightarrow x$ involves leftward displacement
 b. x involves rightward displacement $\Rightarrow x \notin M$

The central concern of this dissertation is to explain the existence of these implications. My thesis is that the manner in which a parser incrementally builds well-formed structures during on-line processing forces $x \in M$ to involve leftward displacement.

¹(3a) and (3b) are not logically equivalent, because the negation of “ x involves leftward displacement” is not “ x involves rightward displacement” since x might not involve displacement at all.

1.2 Movement

The discussion in this dissertation is couched in the terminology of the Principles and Parameters tradition of syntactic theory. Cases of displacement with properties (2a–f) are analysed in terms of a syntactic operation called *movement*. A priori, movement can occur both to the left and to the right. In examples (4)–(12) the displaced constituent is shown in **boldface** font. (4)–(8) show leftward displacement, while (9)–(12) show rightward displacement (which, it will be argued in chapter 3, only superficially instantiates rightward movement).

Leftward displacement

(4) *Topicalization*

Mary, John thinks that Bill likes ____.

(5) *Question formation*

Who does John think that Bill likes ____.

(6) *Relative clause formation*

I read the book **which** Wolfgang thinks that Ardelia likes ____.

(7) *Raising*

John is likely ____ to fail the exam.

(8) *Passive*

Mary was recognized ____ for her achievements.

Rightward displacement (apparent rightward movement)

(9) *Right node raising*

Mary accepted the claim that Bill questioned ____ and John rejected outright

— that story Susan concocted about UFO's.

- (10) *Extraposition (from NP)*

The mayor ejected the man — from the room **who disrupted the last meeting.**

- (11) *Heavy NP Shift*

The mayor proposed — to the council **a seven percent property tax reduction.**

- (12) *Presentational "there" insertion*

There arrived as the mayor spoke **a protester who disrupted the last meeting.**

It is well-established that rightward "movement" is more constrained than leftward movement in its operation (Ross 1986, Grosu 1973, Culicover and Rochemont 1990, Rochemont 1992).² A longstanding question in the linguistics literature is *why* rightward "movement" behaves differently from leftward movement. Put differently, the question is why linear order should play a role in constraining movement.³ Much work has been devoted to correctly characterizing the locality constraints on rightward "movement", without adequately addressing the issue of why there is a difference between leftward and rightward "movement" to begin with.

²(Ross 1986) was originally presented as the author's Ph.D. dissertation (MIT, 1967), and is also available as (Ross 1967).

³In fact, virtually all principles making reference to linear order have been reformulated in terms of hierarchical structure only. For example, the empirical ground covered by the Leftness Condition (Chomsky 1976, Higginbotham 1980) can be accounted for in strictly hierarchical terms as in, for example, (Koopman and Sportiche 1982, Safir 1984).

A posteriori, cases of rightward “movement” do not display the characteristic properties of movement, as listed in (2a-f). If rightward “movement” is not movement after all, then it is no mystery that it behaves differently from leftward movement. Accepting this, we are left with another mystery: why is movement licit only when it is leftward? Rochemont (1992) suggests that this is not a syntactic phenomenon at all, but rather derives from processing considerations. Following Rochemont’s lead, I propose a parsing-based explanation for this linear asymmetry in the operation of movement.

The main results of the dissertation are as follows. The parsing algorithm presented herein offers an explanation for the curious fact that overt movement is leftward. It also explains why, in a language like Italian, postverbal DPs are preferentially interpreted as objects rather than subjects when they appear after optionally transitive verbs. There are also several predictions which arise from this work. The operation of the algorithm predicts that a prenominal relative clause is possible only in a language which permits personal pronouns to act as relative pronouns. There is also predicted to be a filled-gap effect with structures (such as left dislocation in English) for which there is an ambiguity as to whether movement has taken place. Finally, this research supports the view that the human language processing mechanism constructs a single underspecified representation, and resolves ambiguities incrementally as the parse progresses.

1.3 Overview of dissertation

The dissertation proceeds as follows. Chapter 2 presents the relevant background material for the dissertation. It covers syntax, psycholinguistics and parsing. I introduce the basic parsing model which serves as a framework for presenting my

proposed algorithm. Chapter 3 reviews evidence in favour of the position that overt rightward “movement” does not exist. Chapter 4 discusses possible accounts for the leftwardness constraint on overt movement. My parsing model is developed in chapter 5. This chapter also contains a brief discussion of the implementation of the proposed parsing algorithm. Chapter 6 tests my thesis, showing in detail how the proposed parsing model accounts for the leftwardness constraint on movement. Chapter 7 explores consequences of the proposed parsing algorithm. Finally, chapter 8 concludes the dissertation with a summary and directions for future work.

Chapter 2

Background

2.1 Introduction

Linguistic competence refers to a speaker's knowledge of language. Linguistic performance refers to a speaker's use of their knowledge of language. In this dissertation I will use these terms in a restricted sense. Competence refers to a speaker's grammar (G), while performance refers to the speaker's use of grammar (parsing, or $P(G)$).

The set of sentences accepted by a speaker's competence, $\mathcal{L}(G)$, is not the same as the set of sentences accepted by a speaker's performance, $\mathcal{L}(P(G))$. There are many sentences which are considered grammatical and yet are difficult to parse. Among these are centre-embedded sentences, as in (13), and garden-path sentences, as in (14).

(13) The rat the cat the dog chased bit died.

(14) The raft floated under the bridge ran aground.

There are also sentences which are not considered grammatical which we can parse quite readily. Although (15) lacks the proper agreement marking on the verb, it is

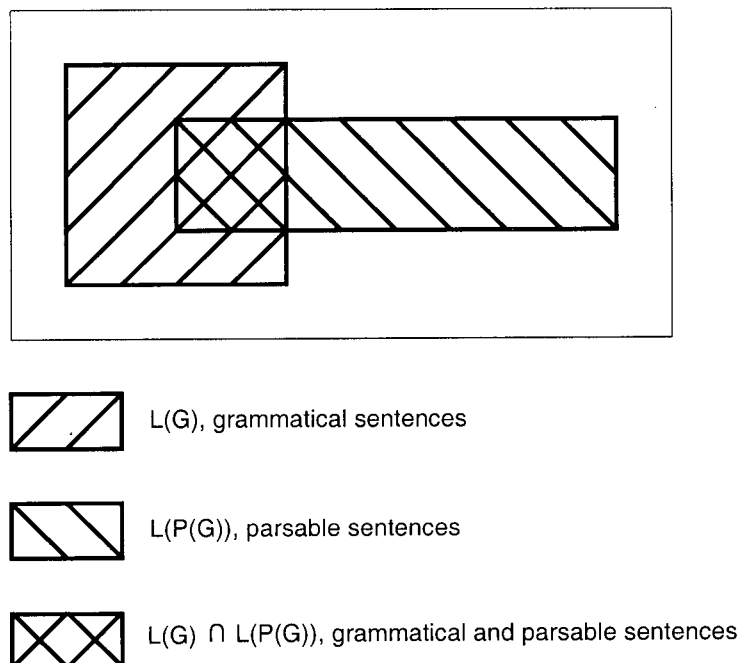


Figure 2.1: The relationship of competence and performance.

not difficult to parse or interpret this sentence.

- (15) The girls and John reads many books each week.

Furthermore, in actual conversation sentence fragments occur, as well as sentences with incorrect tense, agreement, and reference. While these sentences may sound odd out of context, they are parsable and understandable.

The relationship of $\mathcal{L}(G)$ and $\mathcal{L}(P(G))$ is shown in figure 2.1, where $\mathcal{L}(G) \not\subseteq \mathcal{L}(P(G))$, $\mathcal{L}(P(G)) \not\subseteq \mathcal{L}(G)$, and $\mathcal{L}(G) \cap \mathcal{L}(P(G)) \neq \emptyset$.

A theory of syntax characterizes the grammar G . Evidence for the structure of G comes primarily from native speaker intuitions about the well-formedness of strings of words. A parsing theory characterizes $P(G)$. Evidence for the structure of $P(G)$ comes primarily from psycholinguistic experimentation. A theory of $P(G)$ builds on a theory of G , either directly or indirectly. In other words, a theory

of natural language parsing cannot be divorced from a theory of natural language grammar.

This chapter presents background information in syntax, psycholinguistics, and the licensing approach to parsing. Section 2.2 presents a sketch of a standard Principles and Parameters theory of syntax, and is addressed to readers unfamiliar with this syntactic framework. Section 2.3 presents different approaches to the processing of filler-gap structures from a psycholinguistic perspective, and is addressed to readers unfamiliar with the psycholinguistic literature. Section 2.4 presents the basics of the licensing approach to parsing. The parsing algorithm which I present in chapter 5 builds on this material.

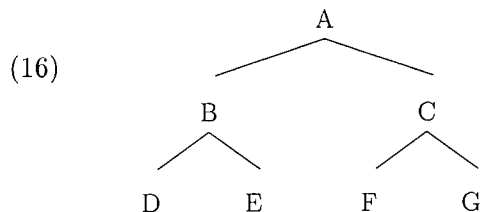
2.2 Syntax

In this section I present a brief overview of those aspects of Principles and Parameters theory which I make direct use of in the remainder of the dissertation. Haegeman (1991) and van Riemsdijk and Williams (1986) both provide comprehensive introductions to syntactic theory within this tradition, while Chomsky and Lasnik (1992) presents a more up-to-date view of syntactic research within the Principles and Parameters tradition. The review is carried out in conjunction with a demonstration of how a parsing algorithm might operate using this information.

Before delving into the main subject matter of this section I present some common notational conventions. In the linguistics literature strings which are considered ungrammatical (strongly deviant) are marked with ‘*’, while examples which are questionable (mildly deviant) are marked with ‘?’. Examples which are considered fully grammatical are typically not annotated with any special symbol. In those cases where it is important to emphasize the grammaticality of the example,

the mark ‘√’ will be used. I will also mark strings which are unprocessable with ‘⊗’. A string marked with only ‘⊗’ is fully grammatical, but unprocessable. A string which is marked with ‘⊗’ and ‘*’ is to be rejected on two grounds: it is both unprocessable and ungrammatical.

The syntactic analysis of a string is typically shown by a tree structure (the parse tree). As an alternative to showing the tree structure, it is often more convenient to show the structure using a labeled bracket representation. For example, the tree shown in (16) can be represented as in (17).



(17) [A [B D E] [C F G]]

Parentheses ‘(’ and ‘)’ are used in examples to show optional material. (18) indicates that the given string is well-formed whether *that* is present or absent.

(18) Wolfgang thinks (that) Ardelia bought the book.

While all the examples in this dissertation are in written form, the focus of interest is spoken language. Words that carry heavy stress in spoken language are written using UPPER CASE LETTERS.

2.2.1 Fundamental concepts

Natural language utterances have structure beyond the linear order of words. Some groupings of words can act as a unit while others cannot. A phrase structure tree

is used to represent this hierarchical structure. Consider the toy grammar shown in (19).¹

- (19) a. $\text{Det} \rightarrow \textit{the}$
b. $\text{N} \rightarrow \textit{cat}$
c. $\text{N} \rightarrow \textit{dog}$
d. $\text{V} \rightarrow \textit{chases}$
e. $\text{S} \rightarrow \text{NP VP}$
f. $\text{NP} \rightarrow \text{Det N}$
g. $\text{VP} \rightarrow \text{V NP}$

This grammar accepts the four strings of words in (20) as sentences. These four sentences share the same basic structure according to the grammar of (19), as shown

¹The phrase structure component of a Principles and Parameters grammar is a context free grammar (CFG). A CFG is formally a 4-tuple (N, T, R, S) where,

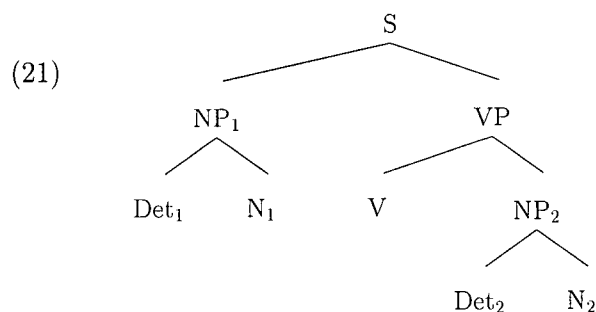
- (i) N is a set of non-terminal symbols,
- (ii) T is a set of terminal symbols,
- (iii) R is a set of rules, where $R \subseteq N \times (N \cup T)^*$,
- (iv) $S \in N$ is called the start symbol.

A rule $r \in R$ is written in the form $A \rightarrow \sigma$, where $A \in N$ and $\sigma \in (N \cup T)^*$. A is called the *left hand side (LHS)* of the rule, while σ is referred to as the *right hand side (RHS)* of the rule.

A grammar is often given informally by writing only the rules of the grammar. The rules implicitly specify the remainder of the grammar. For example, the set of rules given in (19) specify the grammar which more formally is expressed as the 4-tuple $(\{\textit{Det}, \textit{N}, \textit{V}, \textit{S}, \textit{NP}, \textit{VP}\}, \{\textit{the}, \textit{cat}, \textit{dog}, \textit{chases}\}, \{(\textit{Det}, \textit{the}), (\textit{N}, \textit{cat}), (\textit{N}, \textit{dog}), (\textit{V}, \textit{chases}), (\textit{S}, \textit{NP}, \textit{VP}), (\textit{NP}, \textit{Det}, \textit{N}), (\textit{VP}, \textit{V}, \textit{NP})\}, \textit{S})$.

in (21).²

- (20) a. the cat chases the cat
 b. the cat chases the dog
 c. the dog chases the cat
 d. the dog chases the dog

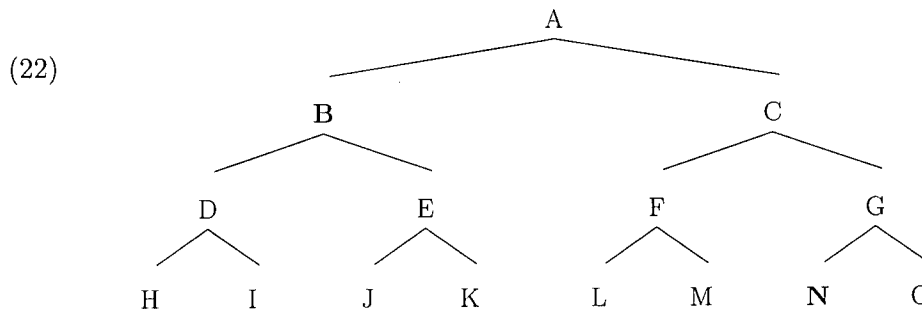


A *constituent* is any subtree of a tree. [VP V [NP₂ Det₂ N₂]] is a constituent of (21), as is N₁. There is no constituent of (21) which includes V and Det₂ but excludes N₂, because any subtree which includes V and Det₂ necessarily also includes N₂.

Basic word order

This dissertation is concerned with the directionality (leftward or rightward) of a syntactic process called *movement*. Movement relates two positions in a phrase structure tree. Consider the two positions labeled **B** and **N** in the tree in example (22). B is to the left of N while N is to the right of B. Whether movement is construed to be leftward or rightward depends on which of the two positions is taken to be the source of movement and which is the target.

²The subscripts on node labels in this tree structure serve only to distinguish between nodes with identical labels. These subscripts do not form part of the structure.



In other words, since *leftward* and *rightward* are relative rather than absolute terms, the reference point for these terms must be fixed. Furthermore, it must be fixed in a non-arbitrary and uniform manner. It is not acceptable to fix the source and target positions for movement on a case by case basis.

Hence, the canonical order of words (the “basic word order”) from which other word orders are derived must be fixed. This permits reference to the directionality of movement without either ambiguity or stipulation.

Basic word order is expressed in terms of the relative order of the Subject (S), the Verb (V), and the Object (O) of the sentence.

Greenberg proposed a typology based on the positioning of the subject, object and verb, resulting in a six-way division of languages into: SVO, SOV, VSO, OVS, VOS and OSV.

... the term ‘basic order’ is typically identified with the order that occurs in stylistically neutral, independent, indicative clauses with full noun phrase (NP) participants, where the subject is definite, agentive and human, the object is a definite semantic patient, and the verb represents an action, not a state or an event. In other words, it is the ordering of constituents in prototypical transitive clauses ...

(Siewierska 1988, p. 8)

As a close first approximation the reference point for determining the directionality of movement will be the order of a transitive verb and its *subject* and *object* arguments in a prototypical transitive clause. The sentence in (23) is an example of a prototypical transitive clause in English.

(23) The boy chased the emu.

The subject of this sentence is *the boy*, the verb is *chased* and the object is *the emu*. According to the criterion given the basic word order of English is Subject-Verb-Object, or SVO.

2.2.2 Projections and extended projections

I assume that the task of a parser is to build a syntactic representation of its input. In this section I present the phrase structure framework which I assume. The constraints discussed in this section govern the construction of local phrase structure representations. These local representations are joined to satisfy *licensing requirements*, as discussed in section 2.2.3.

Projections

I follow Frank (1992) in adopting Grimshaw's (1991) system of extended projections for phrase structure description. This system is an extension of the standard \bar{X} theory of phrase structure. Kornai and Pullum (1990) express \bar{X} theory as a set of constraints on context-free rewrite systems, as in (24).

- (24) a. LEXICALITY: phrasal categories are projections of lexical categories
 b. SUCCESSION: nonterminal X^n rewrites as X^{n-1} (n and $n - 1$ indicate bar-levels)

- c. UNIFORMITY: all maximal projections have the same bar level
- d. MAXIMALITY: the RHS of a rule contains maximal projections and exactly one head
- e. CENTRALITY: the start symbol of a grammar is a maximal projection
- f. OPTIONALITY: non-heads in the RHS of rules are optional
- g. PERIPHERALITY: the head is either first or last in the RHS of all rules

I adopt two other standard assumptions,

- (25) a. BINARY BRANCHING: the RHS of a rule contains at most one non-head
- b. HEIGHT: the maximal number of bar levels is 2

These constraints can be expressed succinctly in terms of the rule schema in (26), where order is irrelevant on the right hand side.

$$(26) \quad X^n \rightarrow X^{n-1} (Y^m), \text{ where } n \in \{1, 2\}, m \in \{0, 1, 2\}.$$

X and Y are variables which range over all grammatical categories. For the purposes of this dissertation it is enough to consider the following categories:

- (27) a. noun (N)
- b. determiner (D)
- c. preposition (P)
- d. verb (V)
- e. inflection (I)
- f. complementizer (C)

²Not all categories are included in this list. For instance I include neither adjectives nor adverbs.

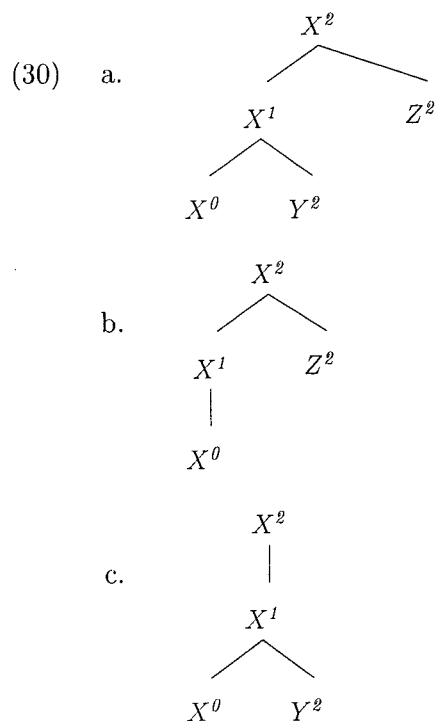
The relational terms *complement* and *specifier* are defined in (28) and (29).

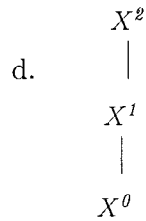
(28) **Definition** Y^m is the *complement* of X^0 if and only if Y^m is sister to $X^0, m \in \{0, 1, 2\}$.

(29) **Definition** Y^m is the *specifier* of X^0 if and only if Y^m is sister to $X^1, m \in \{0, 1, 2\}$.

By abuse of terminology the complement (or specifier) of a head is sometimes referred to as the complement (or specifier) of the maximal projection of the head.

Ignoring the linear order of constituents, these constraints limit the possible phrase structure configurations to those shown in (30a–d).

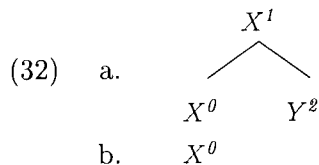




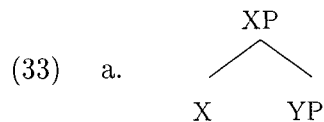
Frank's (1992) model abandons the uniformity constraint, but adds the constraint that projection only occurs if required. I call this constraint the non-vacuous projection constraint,

- (31) NON-VACUOUS PROJECTION: phrase structure is only projected if it is required

Exactly what constitutes a requirement for projection is discussed below. The basic idea is that an X^2 level is projected if and only if a specifier is required, and an X^1 level is projected if and only if either a specifier or a complement is required, or both. Instead of (30c-d) we thus have (32a-b) respectively.



It is common practice to write XP in place of X^2 , \bar{X} in place of X^1 , and X in place of X^0 . I will also use XP to denote the maximal actual projection of a head, whether or not the maximal projection is of bar level 2. Context will determine whether XP denotes a phrase of bar level 2, or simply the maximal projection of a head, of bar level 0, 1 or 2. Using these conventions (32a-b) can be written as (33a-b).



- b. XP

Extended projections

The set of categories in (27) is divided into lexical and non-lexical (or functional) categories:³

(34) lexical

- a. noun (N)
- b. verb (V)

(35) functional

- a. determiner (D)
- b. preposition (P)
- c. inflection (I)
- d. complementizer (C)

The lexical categories are more primitively characterized in terms of the binary-value features **n** and **v**:⁴

(36) Featural decomposition of lexical category labels

	+n	-n
+v	A	V
-v	N	P

³Prepositions (postpositions) are not uniformly functional or lexical. Some prepositions (postpositions) are lexical while others are functional. I do not address this issue here.

⁴For the purposes of this typology, I include the lexical categories adjective (A) and (lexical) pre/post position (P).

Grimshaw proposes that functional categories be included in this typology. Members of a functional category always select the same complement category. This selection by functional categories of a fixed category of complement is referred to as *functional selection*.

(37) Selection of complement category by functional categories

category	complement
C	I
I	V
P	D
D	N

Grimshaw extends the standard typology with a feature “F”, whose allowable values are drawn from the set $\{0,1,2\}$, as shown in (38). In parallel to the bar levels of \bar{X} theory, these are functional levels. Each lexical category supports up to two functional projections, much like a head supports up to two \bar{X} projections (the single bar level is projected to host a complement, the double bar level is projected to host a specifier).

(38) Featural decomposition of category labels

F	V	N	Category
2	+	-	C
1	+	-	I
0	+	-	V
2	-	+	P
1	-	+	D
0	-	+	N

2.2.3 Principle of full interpretation

An important principle in much recent syntactic theory is the *Principle of Full Interpretation* (PFI) (Chomsky 1986). The PFI states that every element in a well-formed syntactic representation must contribute something to that representation. More formally the PFI requires that every element in a well-formed syntactic representation be *licensed*.

Abney (1987) and Abney and Cole (1986) propose that the need to satisfy *licensing requirements* be used as a basis for a parsing algorithm. A licensing parser builds structure by determining how each element is licensed in relation to other constituents of that structure. Building on Abney's (1987) licensing parsing model, Frank (1992) explicates a *Generalized Licensing* parsing framework. My parsing algorithm, which I discuss in detail in chapter 5, builds on Franks' Generalized Licensing proposal. In the next several sections I explain how various syntactic licensing mechanisms are used by a licensing parser to build structure. I begin with a discussion of thematic role assignment.

2.2.4 Thematic roles

Every argument of a predicate is assigned a *thematic role* (or θ role). A constituent's θ role indicates the conceptual role that it plays in the event described by the predicate. For example, in (39) we know that *the boy* is doing the chasing and that *the emu* is being chased. *The boy* is called the **agent** of the action, while *the emu* is the **patient** of the action. **Agent** and **patient** are θ roles.

(39) The boy chased the emu.

Subject and **object** are *grammatical functions*.⁵ In (39) *the boy* is the subject and *the emu* is the object. Though subjects are often agents and objects are often patients, this need not be the case. In (40) the grammatical function of *the emu* is different from what it is in (39), but the θ role of *the emu* is the same in both cases. Determining the θ roles of the arguments in a sentence is a fundamental step in interpreting it.

(40) The emu was chased (by the boy).

The θ criterion is a syntactic constraint which requires that each argument bear exactly one θ role, and that every θ role present in a structure be assigned to an argument. The thematic structure of a verb is specified in a θ grid, which forms part of a verb's lexical entry. For example, the θ grid of the verb *chase* is,⁶

(41) *chase*: (Agent, Patient)

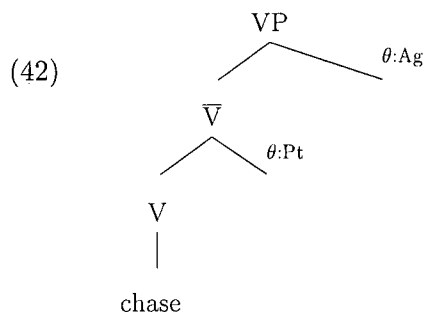
This lexically-specified information is projected into the following phrase structure fragment.⁷ For each θ role which a verb assigns, a phrase structure position is projected.⁸

⁵Unlike some other grammatical frameworks, the Principles and Parameters framework takes *grammatical function* to be a derivative notion rather than a primitive of the theory. A *grammatical function* in Principles and Parameters theories denotes a position in a tree.

⁶The content of the θ role is of no significance to the syntax. Only the fact that a θ role is or is not assigned is relevant. Unless it is important to distinguish different θ -roles, I will henceforth not mention the content of θ -roles.

⁷See (Koopman and Sportiche 1991, Kuroda 1988) for arguments in support of VP-internal subject structures.

⁸In this example the labels "Ag" and "Pt" are retained for expository purposes only. These labels do not form part of the structure under consideration.



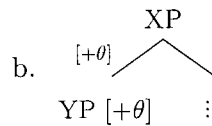
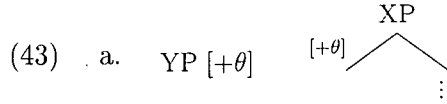
A licensing parser interprets the θ criterion procedurally as an instruction to find arguments to satisfy the θ roles which a predicate assigns. As each token of input is processed the parser attempts to satisfy θ role assignment requirements. This idea can be generalized by abstracting away from any specific licensing feature to yield the driving force of a licensing parser: the desire to satisfy licensing requirements. In this chapter I present Frank's (1992) model of licensing requirement satisfaction.⁹ In Frank's (1992) model each constituent is annotated with two sets of licensing specifications. The first specification indicates what licensing features the constituent assigns. This is the GIVES specification. The second specification indicates what licensing features the constituent is in need of. This is the NEEDS specification.

Whether a given element can satisfy a particular licensing feature assignment requirement is determined by the compatibility of the assigner's licensing feature with the licensing requirements of the assignee. In order to discharge a licensing feature assignment the assignee must be specified as having an identical licensing feature requirement.¹⁰ For example, if a phrase is to discharge a predicate's θ role

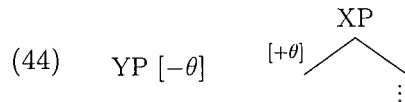
⁹In chapter 5 I propose a slight modification.

¹⁰I depart here somewhat from Frank's model. GIVES specifications are triples $\langle \text{Feature}, \text{Value}, \text{Direction} \rangle$. I assume that the value specification is either +, meaning that the feature is assigned, or -, meaning that the feature is not assigned. I will also not discuss

the phrase must be specified as needing a θ role. Example (43a) shows a case where YP is specified as having a θ role requirement, and XP assigns a θ role to its specifier. YP can be attached into the specifier position, thereby discharging the θ role, as in (43b).



(44) shows a case where attachment of YP into XP cannot proceed because of a mismatch in the licensing feature assignment of the position $[+\theta]$ and the licensing feature requirement of YP $[-\theta]$. XP assigns a θ role to its specifier, but YP must not be assigned a θ role. XP and YP therefore have contradictory values for the θ feature.



2.2.5 Case marking

Languages mark the relationship between verbs and their arguments in various ways. Languages like English rely on word order. Head-marking languages employ morphology on the predicate (head) to indicate the grammatical function of arguments. Argument-marking morphology appearing on a head is often called agreement. Dependent-marking languages employ morphology on the arguments the directionality component. NEEDS specifications are pairs $\langle \text{Feature}, \text{Value} \rangle$. Again, I will assume that the value specification is either +, meaning that licensing by this features is required, or -, meaning that licensing by this feature must not take place.

(dependents) to indicate their grammatical function. Argument-marking morphology appearing on dependents is called *case marking*.

Languages do not fall neatly into exactly one category. English shows some overt case marking and some overt agreement. Word order remains the primary mechanism for determining grammatical function, however.

Case marking is reflected morphologically in the English pronominal system. Hence, while the morphological form of full noun phrases does not vary with their grammatical function, that of pronouns does.¹¹

- (45) a. The woman helped the little boy cross the street.
b. The little boy helped the woman cross the street.

- (46) a. She helped him cross the street.
 3sg-Nom 3sg-Acc
b. He helped her cross the street.
 3sg-Nom 3sg-Acc

A fundamental assumption of Case theory is that all lexically overt noun phrases bear Case marking,¹² even if it is not overtly realized as morphological case. The *Case filter* is a syntactic constraint which states that every lexically overt noun phrase appearing in a phrase structure tree must bear (abstract) Case in order for the structure to be well formed.

¹¹The abbreviations *3sg-Nom* and *3sg-Acc* are interpreted as follows. *3* refers to third person, *sg* refers to singular number, *Nom* refers to nominative case, and *Acc* refers to accusative case.

¹²This is known as *abstract Case*. The word “Case” is capitalized when used in this sense. The word “case” is not capitalized when used in the sense of overt morphology which reflects abstract Case.

(47) *Case Filter*

Every phonetically realized noun phrase must be assigned (abstract) Case. As with θ marking it is the presence or absence of Case marking which is relevant, not any particular Case (e.g. nominative or accusative). In Principles and Parameters theory, there are several levels of syntactic representation: D-Structure (DS), S-Structure (SS) and Logical Form (LF). The Case filter applies at S-Structure.¹³

Adopting the definition of Lasnik and Uriagereka (1988),

(48) α assigns Case to β if

- a. α is a Case assigner;
- b. α governs β ;
- c. α is adjacent to β

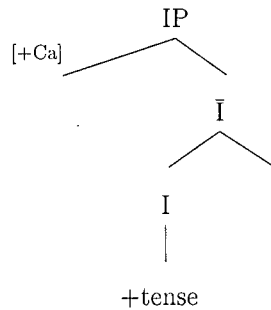
For instance, nominative Case is assigned by tensed inflection to its specifier position (see 49a), and accusative Case is assigned by a verb to its complement (see 49b).^{14,15}

¹³The parsing algorithm presented in this thesis builds a Logical Form representation directly. Representations corresponding to DS and SS are not constructed. This does not pose a problem with respect to the application of the Case filter. In section 2.2.6 movement and chains are introduced. Movement is a function mapping one level of representation into another, and chains at a given level of representation record the movement history. The Case filter can be expressed as a condition on chains, rather than as a condition applying to a certain level of representation.

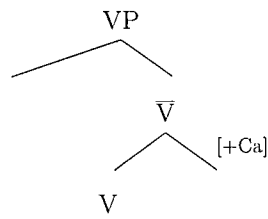
¹⁴I present a traditional version of Case theory. Case and agreement theories have recently undergone many radical revisions (see (Chomsky 1995) and references cited therein). For the purposes of this dissertation the traditional Case theory is sufficient, and has the virtue of simplicity. I do not believe anything crucial to this dissertation hinges on the details of Case theory.

¹⁵Especially in tree diagrams I use the abbreviation “Ca” to refer to the Case marking

- (49) a. Nominative Case assignment by tensed inflection



- b. Accusative Case assignment by a verb to its complement



2.2.6 The basics of movement

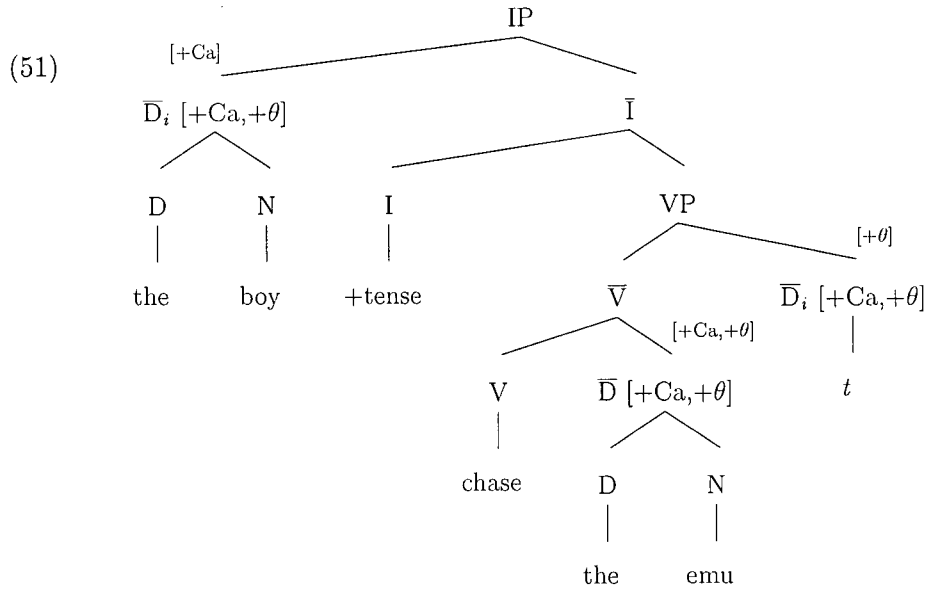
Movement is a syntactic operation which associates a constituent with more than one position in a syntactic structure in order to satisfy licensing requirements. There are two sorts of movement: movement of a phrasal category and movement of a head. Movement of a phrasal category is further subdivided into *A movement* and *Ā movement*. *A movement* is discussed immediately below, *Ā movement* in section 2.2.7, and head movement in section 2.2.8.

Consider as an example our prototypical English transitive sentence (23), repeated as (50).

(50) The boy chased the emu.

licensing feature.

Combining the assumptions about θ role assignment and Case marking, we have the following structure for (50):

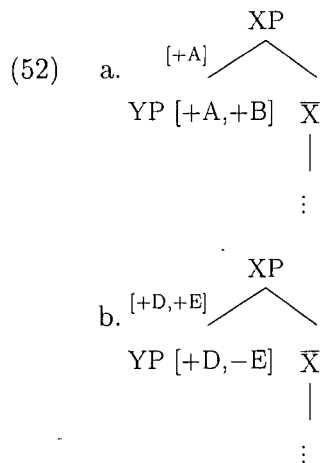


Notice that while the verbal complement is both Case and θ marked, the verbal specifier is only assigned a θ role. Since the Case filter requires that phonetically overt noun phrases be Case marked, a noun phrase like *the boy* can only be interpreted as the subject argument of the verb if it is associated with two positions in the syntactic representation: the specifier of V and the specifier of I. Movement takes place from the specifier of V to the specifier of I. The movement operation leaves behind a phonetically silent copy of the moved constituent called a *trace*. Traces and their antecedents are annotated with an identical index. A moved constituent (the antecedent) and its (zero or more) traces together form a *chain*. An unmoved constituent forms a singleton chain. A trace is usually denoted by *t*.

The chain formed by the two constituents $[\overline{\text{D}} \text{ the boy}]$ and $[\overline{\text{D}} t]$ is called an *A chain* (for *Argument chain*), and is formed by *A movement*. An A chain is defined to be a chain whose *foot* (lowest position) is θ marked and whose *head* (highest

position) is Case marked.

Precisely because a constituent can move it need not have all of its licensing requirements met in a single position. A position must have all of its licensing requirements discharged by the constituent which occupies it, however. Exactly one chain may pass through a position. The situation in (52a) is permitted, since YP can move to have its [+B] licensing requirement fulfilled. The situation in (52b) is illicit, since the assignment of the licensing feature [+E] cannot take place. This assignment cannot take place because the YP has a [−E] specification. A negative (−) specification indicates that the element bearing this specification must not be assigned such a licensing feature.



In section 2.4 I consider in detail how the parsing of a simple example of movement proceeds.

2.2.7 Operators and A-bar movement

Recall the prototypical English transitive sentence (23), repeated as (53).

(53) The boy chased the emu.

Consider now the following sentences.

- (54) a. Who did the boy chase?
 b. Who does the kangaroo think that the boy chased?

In both of these sentences *the boy* is interpreted as the agent of *chase*, while *who* is interpreted as its patient. According to Principles and Parameters theory *who* is interpreted as the patient of *chase* because it has undergone movement to its surface position from the object position of *chase*.

Unlike example (50) it is not Case marking which motivates movement of *who* here, since *chase* both Case and θ marks its complement. A question word, such as *who*, is a type of *operator*.¹⁶ An operator is like a quantifier in predicate logic. An operator must bind a variable. An operator also has a structurally defined scope. An operator can only bind a variable which is within its scope. Operators have a licensing need, which I represent as [+Op]. Movement is motivated by the need to satisfy the [+Op] licensing requirement.¹⁷

Following Browning (1987), I define the specifier of C position to be an operator position. Translated into licensing terms, this means that the [+Op] licensing requirement of operators is satisfiable in the specifier of C position.

The structure of (54a) is shown in (57). There are two chains in this structure. The chain consisting of [\overline{D} the boy] and its trace is an A chain. The chain consisting of [D who] and its trace forms an \overline{A} chain (for *non-argument chain*).

- (55) **Definition** α is an *A position* if and only if (i) a θ role is assigned in α , or
 (ii) α is the specifier of I.

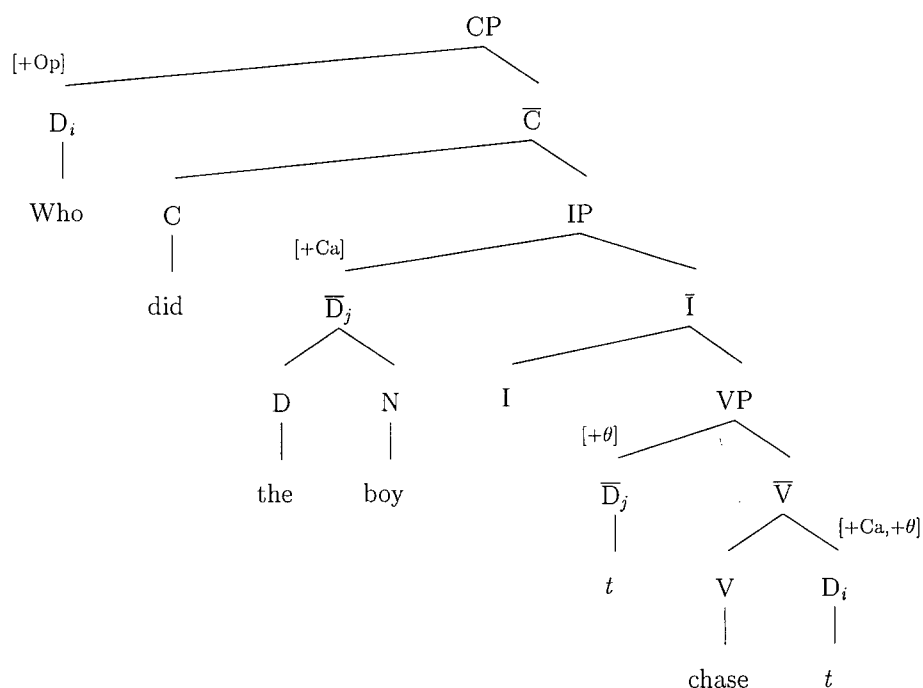
¹⁶The term *operator* is given a precise definition in chapter 5. An intuitive understanding of an operator is sufficient at this point.

¹⁷Again, this will be revisited in chapter 5.

(56) **Definition** α is an \bar{A} position if and only if α is not an A position.

I will define an \bar{A} chain to be a chain whose head is in an \bar{A} position.

(57) Who did the boy chase?



2.2.8 Head movement

A third type of displacement often analysed in terms of movement is called *head movement*. Displacement of a head has different properties than either A or \bar{A} movement. For example, head movement affects only a *head* (an X^0 category), and not any larger syntactic constituent. Head movement is also subject to very strict locality constraints. Because the properties of head movement are quite distinct from those of either A or \bar{A} movement, I do not discuss head movement further.

2.2.9 Binding theory and empty categories

The last piece of syntactic theory I need to introduce is binding theory. Broadly speaking, binding theory is concerned with coindexation and coreference. Before defining what it means for a constituent to be bound I must introduce some structural relations on trees.

Assuming that α , β , and γ are nodes of a tree τ ,

(58) **Definition** α *immediately dominates* β iff β is a daughter of α .

(59) **Definition** α *dominates* β iff either

- (i) α immediately dominates β or
- (ii) γ immediately dominates β and α dominates γ .

(60) **Definition** α *c-commands* β iff

- (i) every γ which dominates α also dominates β , and
- (ii) α does not dominate β .

We can now define binding:

(61) **Definition** α *binds* β iff

- (i) α c-commands β and
- (ii) α and β are coindexed.

Coindexation indicates obligatory coreference.

Reflexives (e.g. *herself*) and reciprocals (e.g. *each other*) must, in English, be bound by an antecedent within a local domain (the embedded clause *that Mary admires herself*), as is shown in (62). The anaphor *herself* must be construed as coreferential with *Mary*. Because *John* is not a possible antecedent for *herself* (62b) is ruled out.

- (62) a. John thinks that Mary admires herself.
 b. * Mary thinks that John admires herself.

Pronouns, on the other hand, must not have antecedents in their local domains. Hence, in (63a) the pronoun *her* cannot be construed as coreferential with *Mary*. *Her* can be construed as coreferential with *Jane*, but this is not necessary. A pronoun can take its antecedent from outside the sentence, as (63b) shows.

- (63) a. Jane thinks that Mary admires her.
 b. John thinks that Mary admires her.

Noun phrases which are not reflexive, reciprocal or pronominal are called *R-expressions*. R-expressions must not be bound.

These informally stated *binding conditions* form the core of the binding theory. They are stated more formally in (64)–(65), as taken from Lasnik and Uriagereka (1988).

- (64) The binding conditions
 a. An anaphor must be bound in its governing category.
 b. A pronominal must be free in its governing category.
 c. An R-expression must be free.

- (65) **Definition** α is the *governing category* for β if and only if α is the minimal NP or S containing β and a governor for β .

Noun phrases are categorized according to their binding theory properties; the typology is based on the binary valued features *anaphoric* and *pronominal*. What is interesting from the point of view of this dissertation is that the *binding theory typology* of noun phrases applies not only to overt noun phrases but also to phonetically empty noun phrases, including the traces of movement.

(66) Typology of overt noun phrases

	+ anaphoric	– anaphoric
+ pronominal		pronouns
– pronominal	reflexives, reciprocals	R-expressions

Binding condition A requires that a [+anaphoric] category be bound in its governing category. Binding condition B requires that a [+pronominal] category be free (i.e. not bound) in its governing category. Since the governing category is the same for [+anaphoric] and [+pronominal] noun phrases, a [+pronominal,+anaphoric] noun phrase is, according to the binding conditions, subject to contradictory requirements: it must at the same time be bound and free in its governing category. A noun phrase can satisfy this requirement only by not having a governing category at all. Such a noun phrase will satisfy conditions A and B of the binding theory vacuously. Since it is ungoverned, such a noun phrase cannot be assigned Case. Hence, there cannot be a [+pronominal,+anaphoric] noun phrase which is overt.

(67) Typology of empty categories based on binding theory properties

	+ anaphoric	– anaphoric
+ pronominal	<i>PRO</i>	<i>pro</i>
– pronominal	A trace	\bar{A} trace

2.2.10 Properties of movement

Consider again the diagnostic properties of movement given in (2a–f) in chapter 1, and repeated here as (68a–f).

(68) $x \in M \leftrightarrow$ the displacement

- a. involves a syntactic constituent

- b. exhibits island effects
- c. obeys the proper binding condition
- d. displays empty category principle effects
- e. can be long-range
- f. licenses parasitic gaps

The purpose of this section is to briefly review these properties. The reader who is not familiar with these properties should consult one of the in-depth sources mentioned at the beginning of the chapter.

The first property simply says that only syntactic constituents undergo movement. For example, while (69a) is licit, (69b) is not.

- (69) a. *Which red block* did the robot move ____?
 b. * *Which block* did the robot move ____*red*____?

The second property expresses that there exist syntactic boundaries across which movement cannot operate. For example, movement cannot take place out of a relative clause. In (70) *who* cannot be moved from the relative clause *that Mary likes* ____.

- (70) * Who did John give the dog that bit ____a treat.

The third property, having to do with the *Proper Binding Condition* (PBC), is that the target position of movement must c-command the source position. In other words, movement must be upwards rather than downwards. This is illustrated in (71).

- (71) a. Bill wonders what the dog will catch ____?
 b. * ____ wonders who the dog will catch the ball?

The fourth property refers to the *Empty Category Principle* (ECP). The ECP requires that the trace of movement be *properly governed*.

- (72) **Definition** α is *properly governed* by β if and only if β governs α and either
- (i) β assigns a θ role to α or (ii) β is co-indexed with α .

The effect of the ECP is to place stricter locality conditions on movements whose source is not a θ marked position. In this manner the ECP accounts for the fact that objects are less restricted than either subjects or adjuncts in their ability to undergo movement.

The fifth property states that movement can, in the general case, be iterated across clauses.

The final property refers to the ability of a movement construction to license a so-called parasitic gap. A parasitic gap is a phonetically empty position which is licensed only when there is an existing operator–trace movement dependency.

2.3 Psycholinguistic background and assumptions

In this section I review some basic results from psycholinguistic experimentation, and theories which have been advanced to account for these results. These results yield insights into how people recover a syntactic structure from natural language input.

The parsing algorithm described in chapter 5 is constructed so that it models human sentence processing with respect to these results. Modeling human performance, rather than simply building a parsing algorithm which “works”,

- further constrains the algorithm,
- makes predictions about how people parse sentences,

- provides independent motivation for my explanation for the directionality constraint on movement

2.3.1 Empirical results from parsing experiments

Shadowing

In what is known as a *shadowing* experiment, Marslen-Wilson (1985) demonstrated that people process natural language input in a syntactic fashion during on-line processing, with minimal delays. The subjects in the experiment were presented (auditorily) with natural language input. The task of the subject was to repeat the input as soon as possible. Subjects were able to “shadow” the input with a minimal delay (250–500 milliseconds) accurately. When presented with ungrammatical input (input with syntactic errors in it), the subjects repeated a grammatical version of the input, again with a minimal delay. This shows that natural language input is processed syntactically during on-line processing.

Garden paths

Garden path sentences¹⁸ are sentences with the following properties:

- they have a locally ambiguous region
- they are globally unambiguous
- the human parser preferentially resolves the local ambiguity in a manner at odds with the globally coherent analysis

¹⁸Pritchett (1992) discusses garden path sentences at length, and contains numerous citations to work in this area.

The existence of syntactic garden path sentences (i.e. garden path sentences in which the local ambiguity is syntactic in nature) offers strong support for the position that the human parser pursues only one possible analysis of its input at a time. If multiple analyses are pursued, no garden paths are expected because both the globally untenable and the globally correct resolution of the local ambiguity are considered simultaneously. Once the incorrect alternative is seen to be untenable it is dropped from consideration. No processing difficulties are predicted to ensue.

Priming

In the context of psycholinguistic experimentation, *priming* refers to the heightened activation of (or increased ease of access to) a lexical item.

The reaction time to a probe (a word which is somehow related to the target, either phonetically or semantically) is measured at various points during the presentation of a sentence. The measurement is made during different trials, with different subjects, and different occurrences of the same sentence.

The reaction time to the probe decreases just after the target is introduced into the sentence, and then gradually rises to its pre-target level. The interesting result of these experiments is that the reaction time decreases again just after the extraction site of the filler is presented.

In other words, the activation of a filler looks something like that shown schematically in figure 2.2, where the probe word is related to *penguin*. The probe word in the idealized example of figure 2.2 is more easily accessible not only after the initial appearance of *penguin*, but also after the theorized extraction site.

Priming evidence provides strong evidence for the existence of a relationship between an antecedent and its extraction site. Priming evidence does not support

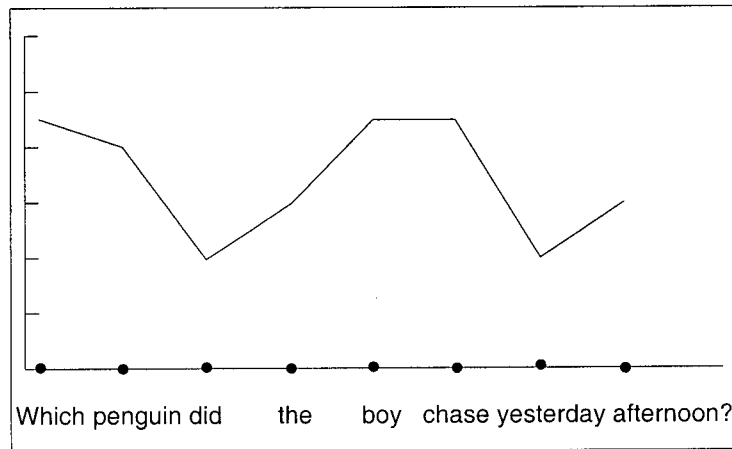


Figure 2.2: An idealized example of priming

an antecedent–trace analysis of the relationship over some other analysis, but it clearly supports the existence of a relationship of some sort.

The filled-gap effect

Both Crain and Fodor (Crain and Fodor 1985) and Stowe (Stowe 1986) report what has come to be known as the *filled-gap effect*. The filled-gap effect is a slight processing slowdown which occurs in a situation where a potential gap location turns out not to be the actual gap location.

[Crain and Fodor (1985)] compared information questions with the corresponding declarative sentence, as in (5).

- (5) a. Who had the little girl expected us to sing those stupid French songs
for ____ at Christmas?
- b. The little girl had expected us to sing those stupid French songs for
Cheryl at Christmas?

Subjects were required to read these sentences one word at a time, controlling the pace of presentation themselves (self-paced reading). The general pattern

of comprehension times was the same for the question and declarative versions of the sentences, but the reading times at and immediately following the object noun phrase ('us' in (5)) were longer for the WH-question than for the corresponding declarative. The most obvious explanation for this difference is that people expect a gap rather than a noun phrase in the object position, when there is an uninterpreted WH-filler in the sentence. The appearance of the object disconfirms this expectation, which causes longer reading times in this area of the sentence.

(Stowe 1986, p.229)

Crain and Fodor (1985) demonstrated the existence of the filled-gap effect only with the verbal object position. Stowe (1986) replicated Crain and Fodor's findings for the verbal object position, but also tested subject position and prepositional object position.

Crain and Fodor (1985) have shown that at an object noun phrase in a WH-question, processing is substantially more difficult than in the corresponding declarative sentence. This indicates that people expect a gap to appear in the object position and are surprised when an overt object noun phrase appears in this position. [...] the same effect obtains when prepositional phrases in the verb phrase are compared. On the other hand no such processing difficulty occurs at a subject noun phrase.

(Stowe 1986, p. 243)

Stowe tested sentences like those shown below in (73). She found that while there was an increase in processing difficulty at *us* in (73b) as compared to (73a), there was no such increase at *Ruth* in either (73c) or (73b) when compared to (73a).

- (73) a. My brother wanted to know if Ruth will bring us home to Mom at Christmas.

- b. My brother wanted to know who Ruth will bring us home to *e* at Christmas.
- c. My brother wanted to know who Ruth will bring *e* home to Mom at Christmas.

The import of the filled-gap effect for parsing is that it provides clues as to how syntactic structure is built. The next section presents theories developed to account for the filled-gap effect. I also present my account of the filled-gap effect, and the lack of a filled-gap effect with subjects (in English).

2.3.2 Theories of filler-gap processing

Movement dependencies are referred to in the psycholinguistic literature as a type of *filler-gap dependency*. There are many factors which play a role in the parsing of filler-gap dependencies. Several different strategies for processing such dependencies have been proposed in the literature (Fodor 1978, Stowe 1984, Frazier 1987b, Frazier and Flores D'Arcais 1989). These proposals may be divided into those which are *filler-driven* and those which are *gap-driven*, and into so-called *first resort* and *last resort* models.

Gap-driven vs. filler-driven

The terms *gap-driven* and *filler-driven* refer to a parser's strategy for locating the endpoints of chains. A gap-driven parser identifies potential chains by their feet (where there is a gap). A filler-driven parser identifies potential chains by their heads (where there is a filler).

Gap-driven The standard characterization of gap-driven processing is that a gap-driven parser treats a sentence with a filler-gap dependency exactly like a sentence without such a dependency, until normal processing breaks down. Precisely when normal processing breaks down depends on a number of factors, but the basic idea is that if the parser expects an overt NP but does not find one, then and only then will a trace be postulated. This approach to filler-gap processing can be represented as ranking the option of an overt NP over that of a trace (empty NP):

(74) Gap-driven processing

$$\begin{bmatrix} \text{overt NP} \\ \text{empty NP} \end{bmatrix}$$

In other words, for a gap-driven parser the normal state of affairs is to encounter overt material rather than gaps. When processing the sentence in example (75), a gap-driven parser recognizes the existence of a filler-gap dependency between *who* and the gap (represented by “—”) only when the gap is identified. The parser does not do anything special when it encounters *who*.

(75) Who did John meet — yesterday?

Filler-driven A filler-driven parser does not continue with normal processing until a gap is reached. A filler-driven parser identifies dislocated phrases, and uses them as triggers for chain building (filler-gap dependency construction).

A filler-driven parser attempts to resolve a filler-gap dependency as soon as possible. In other words, a filler-driven parser attempts to postulate a trace in a noun phrase position before looking for a suitable overt noun phrase to attach into the position. Only if no suitable filler is available to support a trace does the parser look for an overt noun phrase to fill the position. A filler-driven parser thus adopts

the opposite ranking.

(76) Filler-driven processing

$$\begin{bmatrix} \text{empty NP} \\ \text{overt NP} \end{bmatrix}$$

When processing the sentence in example (75), a filler-driven parser recognizes the existence of a filler-gap dependency between *who* and some as yet to be identified gap as soon as it encounters *who*.

These two approaches can be thought of as expressing a preference for taking an identified noun phrase *position* as either the head of a chain (the gap-driven option) or as the foot of a chain (the filler-driven option).

First resort vs. last resort

While many noun phrase positions are obligatory, some are optional. Obligatory noun phrase positions which are phonetically unfilled are called *doubtless gaps*, while optional ones are called *doubtful gaps*.¹⁹ A doubtless gap must be filled by an empty category. A doubtful gap represents either a nonexistent position (this is the case, for example, with an optionally transitive verb which is being used in its *intransitive* sense), or an empty category (consider, for example, the case of an optionally transitive verb which is being used in its transitive sense). A *true doubtful gap* is a doubtful gap which must be filled by an empty category, while a *false doubtful gap* is a nonexistent position.

Consider the following example:

(77) What did Ardelia read to the students from?

¹⁹The terminology *doubtful* and *doubtless* is due to Fodor (1978).

When the verb *read* is encountered, it is not clear in which sense it is being used: should *what* be interpreted as the object of *read* or not? Hence, *read* is deemed to be followed by a doubtful gap. In the final analysis it is clear that *what* must be interpreted as the object of the preposition *from*. At this point the “object position” of *read* can be classified as a false doubtful gap.

There are two basic strategies for parsing doubtful gaps, a first resort and a last resort strategy.²⁰ These are represented by the rankings in (78).

- (78) a. first resort strategy $\begin{bmatrix} \text{NP} \\ \epsilon \end{bmatrix}$
- b. last resort strategy $\begin{bmatrix} \epsilon \\ \text{NP} \end{bmatrix}$

First resort Employing a first resort strategy, a parser prefers to interpret doubtful gaps as true gaps. Such a parser will detect all doubtful gaps.

Last resort Employing a last resort strategy, a parser prefers to interpret doubtful gaps as non-gaps. Such a parser will detect no doubtful gaps, unless forced to do so. For example, given the optionally transitive verb *walk*, the string *I walked* may

²⁰Fodor (1978) argues that neither of these strategies is always correct. Instead, she proposes a lexical expectation model in which the preference is stated in the lexical entry of each optionally transitive verb. The strategy used depends on the preference expressed. A first resort strategy is used with optionally transitive verbs which are more likely to be transitive than intransitive. On the other hand a last resort strategy is used with optionally transitive verbs which are more likely to be intransitive than transitive.

It is possible to incorporate a lexical expectation model into the parsing algorithm presented in this thesis. Doing so does not alter the predictions made in any significant way.

be interpreted as an intransitive clause. If *the dog* is encountered next in the input, the transitive interpretation of *walk* must be chosen.

2.4 A licensing parsing algorithm

In this section I present a basic licensing parsing algorithm. A *licensing parser* builds syntactic representations by satisfying licensing requirements. Licensing requirements are well-formedness conditions on grammatical structure. A licensing parser interprets these declarative constraints in a procedural manner. In other words, a licensing parser uses licensing conditions to guide structure building.

I begin with a discussion of determinism. The hypothesis that parsing of natural language proceeds in a deterministic manner accounts for the garden path phenomenon.

2.4.1 Determinism

I adopt the deterministic model of parsing first introduced by Marcus (1980).

... the Determinism Hypothesis must mean something more than the claim that language can be parsed by a deterministic machine. As noted above, any computational mechanism that physically exists is deterministic in the automata theoretic sense, and thus any process which is specified by an algorithm for such a machine must be similarly deterministic. From this it follows that any parser, whether it simulates a nondeterministic machine or not, must itself be deterministic.

Rather than attempting to formulate any rigorous, general explanation of what it means to “not simulate a nondeterministic machine”, I will focus instead on several specific properties of the grammar interpreter which will be presented below. [...]

First, all syntactic substructures created by the grammar interpreter are permanent. This eliminates the possibility of simulating determinism by “backtracking”, i.e. by undoing the actions that were done while pursuing a guess that turns out to be incorrect. In terms of the structures that the interpreter creates and manipulates, this will mean that once a parse node is created, it cannot be destroyed; that once a node is labeled with a given grammatical feature, that feature cannot be removed; and that once a node is attached to another node as its daughter, that attachment cannot be broken.

Second, all syntactic substructures created by the grammar interpreter for a given input must be output as part of the syntactic structure assigned to that input. [...]

Finally, the internal state of the mechanism must be constrained in such a way that no temporary syntactic structures are encoded within the internal state of the machine.

(Marcus 1980, p. 12)

According to Marcus’ characterization, a parser is deterministic only if it pursues exactly one analysis at a time. A non-deterministic parser can investigate multiple analyses at the same time. The existence of garden path sentences is a strong empirical argument against nondeterministic parsers as models of human sentence parsing.²¹ Consider the example in (79).

(79) The boat floated down the river sank

²¹There are other parsing models which also offer accounts of the existence of garden path sentences. For example, a ranked-parallel parser (Gibson, Hickock and Schütze 1994) pursues analyses which are below a certain “cost” threshold. In a given situation there may be one or more than one such analysis. Since a fine-grained investigation of garden-path behaviour is not my focus, I do not consider these alternatives.

This sentence is not globally ambiguous. There is only one analysis of this sentence: *floated down the river* is a reduced relative clause modifying *the boat*; *sank* is the main verb of the sentence. Notice, however, that the word *floated* is ambiguous between a past tense verb and a past participle. A plausible analysis (in fact, the initially preferred analysis) of *the boat floated down the river* is that this is a main clause. The appearance of *sank* rules out this interpretation. Since the presence or absence of the word *sank* rules out one or the other of the two interpretations of *floated*, the point of resolution of this sentence's ambiguity occurs after the word *river*.

A deterministic parser must resolve the ambiguity of *floated* when it arises. The point of resolution is not accessible to the parser when the ambiguity arises (otherwise there would be no ambiguity). Hence, a deterministic parser will sometimes make a mistake. When it does, a garden path results.

A nondeterministic parser pursues both possible analyses to the point of disambiguation. After the point of disambiguation it pursues only the globally viable analysis.

Marcus's (1980) thesis was that English could be parsed deterministically. Marcus's claim was that garden path sentences were problematic precisely because they could not be parsed deterministically.

Interestingly, not all local ambiguities give rise to garden path effects. Marcus considers the examples in (80a-b).

- (80) a. Have the boys take the exam today.
b. Have the boys taken the exam today?

These two sentences must be analysed differently. (80a) is an imperative construction, while (80b) is a question. The ambiguity is not resolvable until the form of

the verb *take* is encountered. While this seems to be a candidate garden path sentence pair, people have no problems parsing example pairs such as these. In order to properly parse sentence pairs such as (80a–b) Marcus endowed his parser with a *lookahead* facility. Before describing what a lookahead facility is, recall that Marcus makes the claim that English can be parsed using a strictly deterministic device.

It is possible for a deterministic machine to simulate a nondeterministic machine. For example, a deterministic parser can pursue one analysis until it becomes untenable, and then try another analysis. This is simulation of nondeterminism through backtracking. Another possibility is for a deterministic parser to delay making a crucial decision until disambiguating information is available. This is simulation of nondeterminism through lookahead. Simulation of nondeterminism is successful only if backtracking or lookahead is unbounded. If a bound is placed on how much lookahead can be used the simulation ceases to be nondeterministic. The Marcus parser makes use of *bounded lookahead* in order to parse examples such as (80a–b). These sentences are locally but not globally ambiguous, and do not give rise to garden paths. Because the Marcus parser uses a strictly bounded lookahead facility, it is still deterministic.

Lookahead is measured in terms of the number of extra input tokens that a parser can look at before making a decision about the current one. Pritchett (1992) shows that this crude measure makes incorrect predictions regarding garden paths. Pritchett shows that although the Marcus parser correctly processes (81a) as a garden path (it cannot parse it within its lookahead facility), it is able to process (81b) without difficulty. People find (81b) is just as severe a garden path as (81a).

- (81) a. The boat floated down the river sank.

- b. The boat floated quickly sank.

Lookahead is only one way to delay decisions, however. A parser can delay making certain decisions by building underspecified representations. This is the approach taken by, for example, Marcus, Hindle and Fleck (1983), Pritchett (1992) and Gorrell (1995). The parsing algorithm I present in chapter 5 does not make use of a lookahead facility, but instead relies on building underspecified representations to delay certain types of decisions. The delay in decision-making permitted by the use of underspecification is limited by the fact that the parser must add to the partial specification at each decision point. Moreover, I require that what is partially specified can stand on its own: it must not be dependent on the presence of some other yet-to-be-encountered element in the input. This is explored in more detail in chapter 5.

2.4.2 Licensing and parsing

The *Principle of Full Interpretation* (PFI) requires that every element present at the LF level of syntactic representation must be semantically interpretable.²² Abney and Cole (1986) and Abney (1987) introduced the idea of using licensing conditions to drive parsing. Their idea is that a parser can use knowledge of how constituents are licensed with respect to each other in order to build structural representations without the use of explicit phrase structure rules. This work was motivated by the move in the Principles and Parameters framework away from phrase structure rules to general constraints on structural well-formedness.

In this section I present a simplified version of a licensing parsing algorithm

²²The PFI also requires that every element present in the PF representation be interpretable in the “articulatory/perceptual” interface, but this is not relevant for my purposes.

developed by Frank (1992). This simplified algorithm serves as the starting point for presenting my algorithm in chapter 5.

Frank (1992) presents a recent exploration of *licensing parsing*. Frank's licensing parsing model departs from Abney's in some interesting respects. Abney's licensing parser builds two structures. It first constructs a *licensing structure* from which the final phrase structure representation is derived. Frank's parser dispenses with this intermediate level of structure. Instead, it builds a phrase structure representation directly.

Abney constrained the use of licensing conditions such that an element could be licensed by exactly one licensing mechanism. θ marking was one Abney's licensing mechanisms, but Case marking was not. Since Case marking and θ marking often go hand in hand, Abney could only allow one of these to be considered a licensing mechanism. Frank's conception of licensing conditions is more liberal. There is no requirement that a constituent be licensed in a unique manner, and so both Case marking and θ marking are used as licensing mechanisms.

Informally, the algorithm works as follows. The parser first reads a word from the input. The parser next looks up the word in its lexicon. In the lexicon the parser finds such information as the word's grammatical category and its licensing properties. On the basis of this information the parser projects a phrase with the word as its head. The parser next tries to join this phrase and any structure it has previously built to form a larger parse tree. I call this step *attachment*. Finally, the parser examines the structure to see if there are any chains which can be built. I call this last step *extension*. This sequence of steps is repeated until the input stream is empty.

I now describe the algorithm, shown more formally in figure 2.3, in more

Tokenize Undo head movement while tokenizing the input.

Repeat

Reception Read a token from the input stream.

Projection Once an input item has been recognized the parser projects structure on the basis of the item's lexical properties.

Attachment The resulting structure needs to be attached into the Current Partial Phrase Marker (CPPM). Attachment is done to satisfy constraints on the assignment of licensing features, such as Case or θ marking. These features cannot be left unassigned.

Movement *Movement* associates a constituent with more than one position in the phrase structure tree. A constituent is extended into a gap position to discharge a feature which must be assigned.

A trace of the extended constituent is inserted into the gap position.

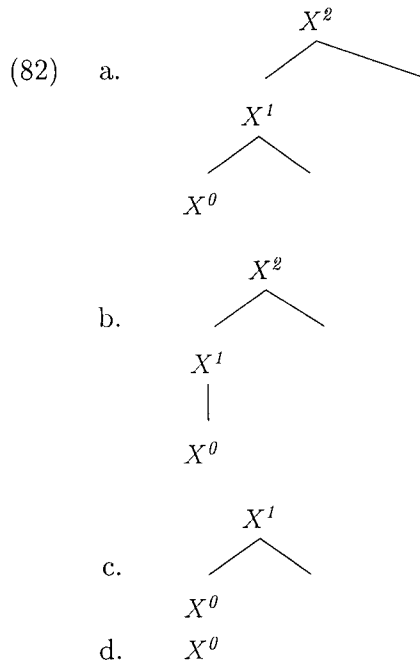
Until the input stream is empty.

Figure 2.3: A simple licensing parsing algorithm

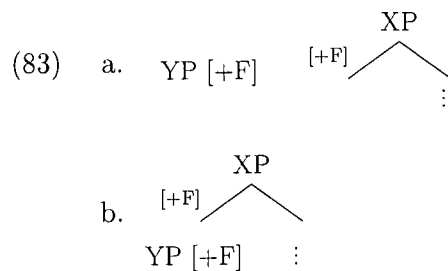
detail. The first point to make is that I am not concerned with head movement in this thesis. I make use of a standard technique for dealing with simple cases of head movement – the input stream is preprocessed by a tokenizer which undoes the effects of head movement. Thus a tensed verb (e.g. *bought*) is tokenized as tense (*past*) and the verb root (*buy*).

Structure is projected as follows. If the head of the phrase assigns a licensing feature to its specifier position, then a specifier is projected. If the head assigns a licensing feature to its complement position, then a complement is projected. If the head does not license either a complement or a specifier, then no structure beyond the head is projected. (82a) is the structure projected from a head which licenses both a specifier and a complement. (82b) is the structure projected from a head which licenses a specifier but no complement. (82c) is the structure projected from a head which licenses a complement but no specifier. (82d) is the structure projected

from a head which licenses neither a specifier nor a complement.

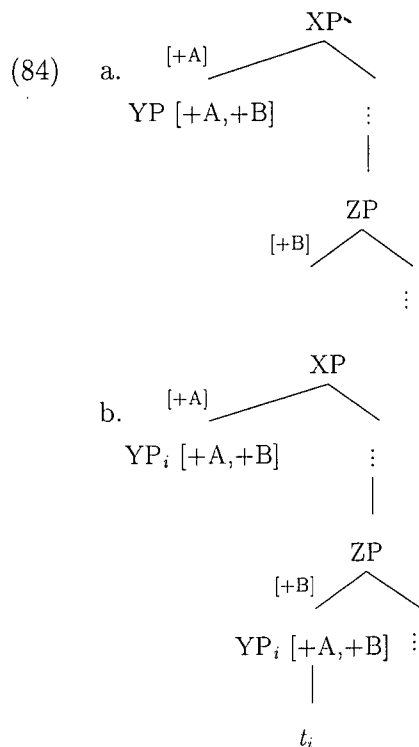


The basic mechanism of attachment is discussed in section 2.2.4. (83a) shows the initial configuration for successful attachment. There is a position which has a licensing specification, $[+F]$, and a constituent which can discharge that licensing feature. The result of attachment is shown in (83b).



The motivation for extension (or chain formation) is similar to that of attachment, namely to satisfy licensing requirements. The main difference in the two processes is that in the case of extension there is no unattached constituent which can discharge the licensing requirement of an empty position. Instead there is a

constituent already in the structure which can satisfy the licensing requirement of the position. (84a) shows an initial configuration for extension. Extension involves creating a coindexed trace of the original constituent, and attaching the trace into the empty position. (84b) shows the result of extension.



Parsing proceeds in order to transfer licensing features from positions to constituents. Movement permits a constituent to satisfy its licensing requirements in more than one position. Notice that the extension process is triggered to satisfy the need of a licensor to discharge licensing features. It is not triggered by the need of a licensee to receive licensing features. In other terms, extension is gap-driven, not filler-driven.

Parsing: an example

Let us now work through the parsing of the sentence in (85a) in detail. The example is one in which the subject DP is assumed to be assigned Case marking in the specifier of IP position, and θ marking in a VP-internal position. The result of tokenizing (85a) is shown in (85b).

- (85) a. Ardelia kissed John.
b. Ardelia +tns kiss John

Round 1

Reception Read **Ardelia** from the input.

Projection Project a DP with licensing requirements $[+Ca, +\theta, -Op]$ from the D^0 Ardelia. Notice that since Ardelia is not an operator, it is explicitly annotated with the licensing specification $[-Op]$.

Attachment There is nothing to attach.

Extension There are no empty positions to consider.

DP $[+Ca, +\theta, -Op]$
|
Ardelia

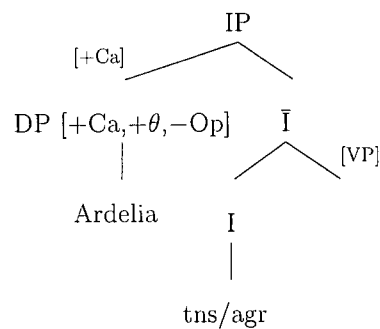
Round 2

Reception Read **+tns** from the input.

Projection Project an IP with both a specifier and a complement from the I^0 head **+tns**. The specifier is licensed by Case assignment. The complement is licensed by functional selection of a category with features $[-N, +V, F \text{ level } 0]$ or, more succinctly, a VP.

Attachment Attach DP as specifier of IP to allow discharge of the [+Ca] licensing feature.

Extension There is no constituent of type $[-N, +V, F \text{ level } 0]$ c-commanding the sole gap of the structure, the complement of I position. Hence, nothing can be extended into this position.

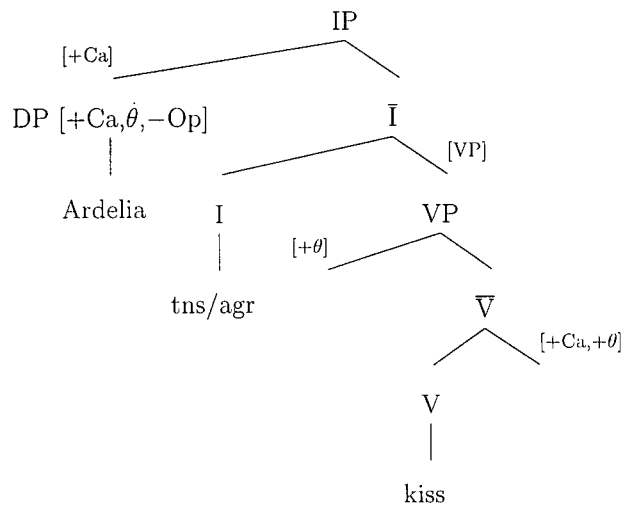


Round 3

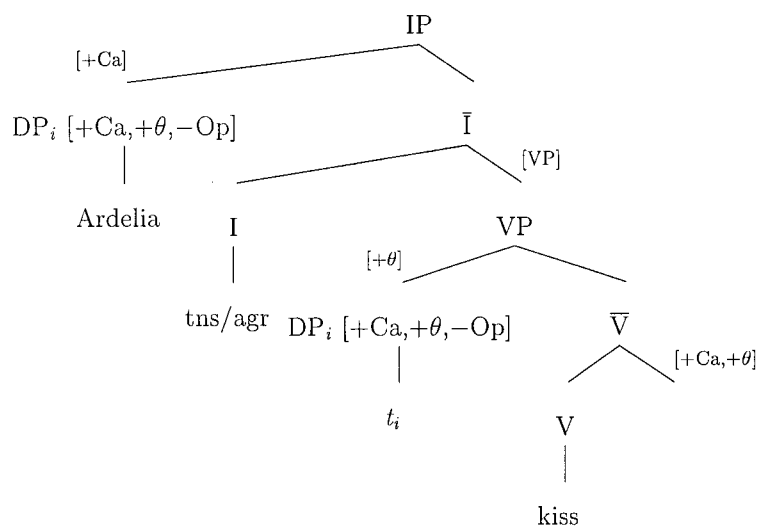
Reception Read **kiss** from the input.

Projection Project a VP from the V^0 head **kiss**. The specifier is licensed by θ assignment, and the complement by both Case and θ assignment.

Attachment Attach VP as complement of IP to satisfy the functional selectional requirement of the IP.



Extension Consider now all potential gap sites within the newly attached VP structure. There are two empty positions, the VP-internal subject position (with licensing feature $[+\theta]$) and the object position (with licensing features $[+\text{Ca}, +\theta]$). There is no constituent of IP which is compatible with the object position, since no constituent has an unsatisfied need of Case marking. The DP *Ardelia* can, however, be extended into the VP-internal subject position. The θ assignment requirement of the verb is thereby satisfied.



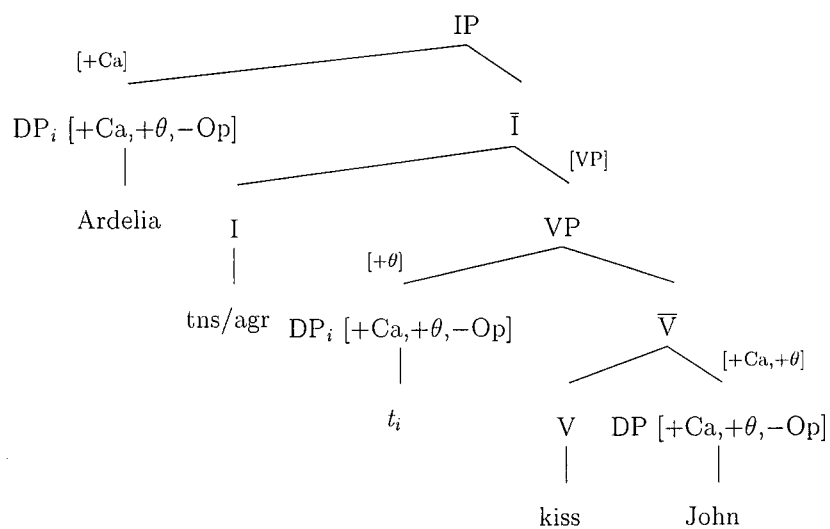
Round 4

Reception Read **John** from the input.

Projection Project a DP with licensing requirements $[+Ca, +\theta, -Op]$ from the D^0 John.

Attachment Attach this DP as the object argument of V, thereby discharging both the Case and θ assignments into this position.

Extension There are no empty positions to consider.



Since all the input has been consumed, and we are left with a fully licensed structure, parsing is finished and successful.

Chapter 3

The problem

In chapter 1 I claimed that all cases of rightward displacement fail to exhibit the cluster of properties which is diagnostic of movement. In this chapter I review the literature which presents evidence in support of this.

3.1 Introduction

It has long been assumed that leftward and rightward displacement are both cases of movement. However, it is also well known that rightward movement is more constrained than its leftward counterpart. Ross (1967) hypothesizes that rightward movement operates within the confines of a single clause. Leftward movement by contrast is free to operate across clause boundaries. Grosu (1973) christens this clause boundedness constraint on rightward movements the Right Roof Constraint (RRC).

- (86) Node A of a phrase marker *commands* node B if neither node dominates the other, and if node B is dominated by the first node S above A. (Ross 1986, p. 201)

- (87) In all rules whose structural index is of the form ... A Y, and whose structural change specifies that A is to be adjoined to the right of Y, A must command Y. (Ross 1986, p. 203)

Culicover and Rochement (1990) argue that various cases of extraposition from NP are not amenable to a movement analysis. While shrinking the domain of operation for rightward movement, from a clause to a phrase, Culicover and Rochement also circumscribe more tightly the empirical ground which any rightward movement analysis must cover. Rochement (1992) formulates this phrase-boundedness constraint as the Rightward Movement Constraint (RMC).

- (88) *Rightward Movement Constraint*

If X^0 governs t , t the trace of rightward movement, then X^0 governs the head of the chain containing t . (Rochement 1992, p. 388)

Recent arguments of Hawkins (1994) and Postal (1994), to be reviewed in section 3.2.3, suggest that there are in fact no cases of rightward displacement which can be analysed as instances of rightward movement. I assume that this is the case. In other words, I assume that the implication of (3), repeated here as (89), is true.¹ My aim in this dissertation is to explain why such an implication should be true.

- (89) a. $x \in M \Rightarrow x$ involves leftward displacement
 b. x involves rightward displacement $\Rightarrow x \notin M$

¹Recall that D is the set of all cases of displacement in a language, and that $M \subseteq D$ is the set of those cases of displacement which satisfy the cluster of properties given in (2a-f) in chapter 1.

3.2 Refuting rightward movement

The purpose of this section is to demonstrate that each case of rightward displacement lacks the cluster of properties characteristic of movement.

3.2.1 Right node raising

Right Node Raising (RNR), shown in (90)–(91), seems to displace material to the right.

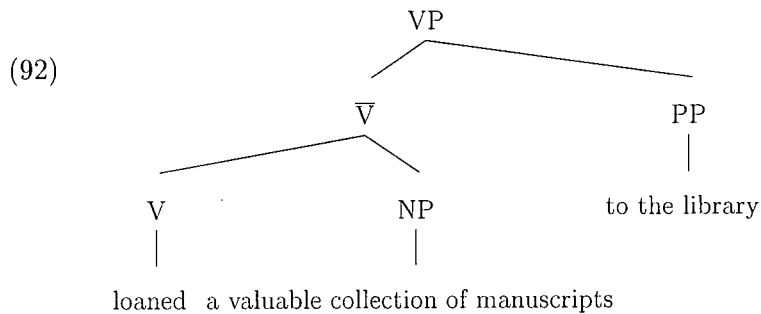
(90) (Abbot 1976, 639)

- a. Smith loaned a valuable collection of manuscripts to the library, and his widow later donated a valuable collection of manuscripts to the library.
- b. Smith loaned —, and his widow later donated —, a valuable collection of manuscripts to the library.

(91) (Rochemont 1992, 393)

John bought and Mary sold everything of value.

A defining property of movement is that it applies to a constituent. Abbot (1976) argues that RNR cannot be derived by movement because non-constituents are able to undergo RNR. This is shown by (90b), in which the non-constituent *a valuable collection of manuscripts to the library* has undergone RNR. The relevant structure is shown in (92).



In further support of the non-movement character of RNR, Levine (1985) shows that RNR does not respect island constraints. If RNR is analysed as movement, then (93) should be ruled out since *to Bill* is extracted out of a complex NP (the relative clause *who had given a set of steak knives*).

(93) (Levine 1985, 492)

John gave a briefcase, and Harry knows someone who had given a set of steak knives, to Bill.

Levine also constructs an argument based on principle C of the binding theory. Principle C requires that full noun phrases must not be coindexed with any c-commanding noun phrase. Under a movement analysis of RNR, the raised NP will not be c-commanded by any coreferential element in either argument. If so, it is predicted that “a pronoun anywhere in either conjunct can be coreferential with any NP in the Raised node.” (Levine 1985, 495). Levine gives examples like those in (94) to show that the prediction is not borne out. If RNR were derived by movement, then the judgements for (94a-b) would be reverse.

- (94) a. * Leon expected her_i — and she_i was encouraged by others — to publish Mary_i's book.
- b. Leon expected Mary_i — and she_i was encouraged by others — to publish her_i book.

Levine concludes that,

[S]uch evidence further reinforces the conclusion that the so-called Raised element in RNR is actually present in all conjuncts in a way that makes it sensitive to island constraints, pronominalization constraints, and so on — that, in fact, it is present in some sense in the phrase structure sites from which it appears to have been displaced.

(Levine 1985, 496)

In other words, no analysis of RNR as involving movement can be maintained.

3.2.2 Extraposition

Rochemont (1992) shows that extraposition,² a construction which appears to involve rightward movement, does not display the diagnostics of movement. An extraposed constituent is interpreted with respect to an antecedent. The locality constraints imposed on the relationship between the antecedent and the extraposed constituent are different from those of movement. Culicover and Rochemont (1990) formulate the Complement Principle to account for the distribution of extraposed constituents.

(95) Complement Principle (Rochemont and Culicover 1990, p. 35)

β is a potential complement of α ($\alpha, \beta = X^{\max}$) only if α and β are in a government relation.

Whereas movement from subjects is in general bad, as shown in (96b), extraposition from subjects is perfectly well-formed, as in (96c).

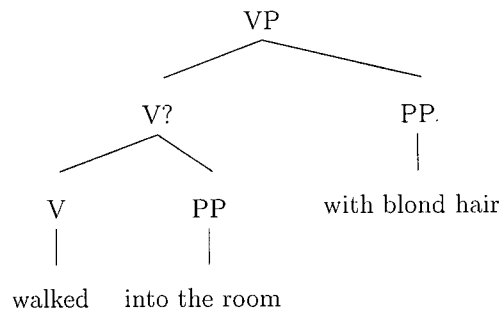
(96) a. A woman [with blond hair] appeared at the door.

²I refer here to extraposition from a noun phrase.

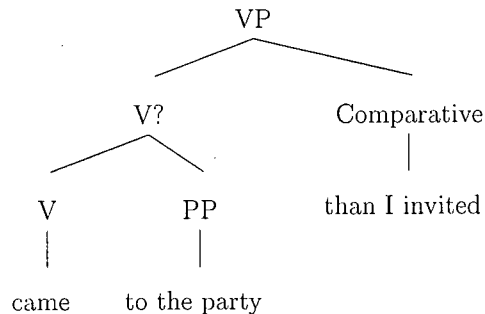
- b. * [With blond hair] a woman appeared at the door.
- c. A woman appeared at the door [with blond hair].

Furthermore, movement must be to a c-commanding position. The possibility of VP ellipsis shows that extraposed constituents can be interpreted with respect to the VP. Under a movement analysis this would involve downward movement, which is disallowed.

- (97) a. A man walked into the room [with blond hair], and then a woman did.



- b. More men came to the party [than I invited], and more women did too.



3.2.3 Heavy NP Shift

Rochemont (1992) argues that Heavy NP Shift (HNPS), shown in (98a), and Presentational *There* Insertion (PTI), shown in (98b), are cases of rightward movement.

- (98) (Rochemont 1992, p. 373)

- a. John bought for his mother a painting that he liked.

- b. There walked into the room a man with long blond hair.

Rochemont notes that HNPS shares with \bar{A} movement the properties

1. HNPS “licenses parasitic gaps” (Rochemont 1992, p. 382)
2. HNPS “shows CED effects, in that there is no possible extraction from it” (Rochemont 1992, p. 382)
3. HNPS’s “postulated trace qualifies as a variable, since it is in a Case-marked A-position” (Rochemont 1992, p. 382)
4. “HNPS in English dative constructions patterns with \bar{A} - and not A-movement with respect to the possibilities for extraction of the direct and indirect objects.” (Rochemont 1992, p. 382-383)

With respect to PTI, Rochemont notes that,

... it is considerably more difficult to determine conclusively the position of the dislocated NP. Parasitic gaps are not licensed by traces in subject position, and although the dislocated subject in PTI exhibits CED effects, such effects are associated also with non-dislocated subjects.

- (30) (a) *Which famous actor did there appear in the newspaper a picture of?
(b) *Which famous actor did a picture of appear in the newspaper?

I will nevertheless hypothesize, following Rochemont (1978), that HNPS and PTI are derivationally parallel: in both cases the dislocated NP is moved rightward by Move α . Thus, although the hypothesis that the dislocated subject in PTI is in \bar{A} -position is consistent with the available evidence it is not confirmed by it.

(Rochemont 1992, p. 383)

Postal (1994) argues that HNPS in fact does not license parasitic gaps. Instead, he claims that rightward extractions license what he terms *pseudoparasitic gaps*. Postal analyses HNPS as a case of (non-coordinate) Right Node Raising. As we saw above, many arguments have been adduced that Right Node Raising should not be analysed as involving movement. If this is the case, then HNPS should not be analysed as involving movement either.

While attempted extraction from a constituent which has undergone HNPS is degraded, I argue presently that it is not clear that this is due to CED effects. Instead, I argue that these cases can be analysed as garden paths. These examples are from Rochemont (1992, p. 382).

- (99) a. John noticed [a picture of his mother] on the wall.
- b. John noticed on the wall [a picture of his mother].
- c. Who did John notice [a picture of *e*] on the wall?
- d. * Who did John notice on the wall [a picture of *e*]?
- (100) a. It was Bill [*O* that John sold Mary a picture of *e*].
- b. * It was Bill [*O* that John sold to Mary a picture of *e*].
- (101) a. Who did John give Mary a picture of *e*?
- b. * Who did John give to Mary a picture of *e*?

Consider first how processing proceeds in (99c). The *filled-gap effect*³ suggests that the parser attempts postulation of a trace in the object position of the verb to resolve the dependency headed by *who*. Since this is incorrect in this example (there is an overt object available), the parser must revise its hypothesis, leading to a measurable effect, the filled-gap effect. This is, in effect, a mild garden

³The filled-gap effect is discussed in more detail in section 2.3.1.

path. Because there is no potential filler available in (99a) or (99b), no such effect is observed in this case.

Consider now how processing proceeds in (99d). Here the parser postulates a trace, just as in (99c). In contrast to (99c), there is no reason to retract this assumption, however, as there is no overt object noun phrase available. The noun phrase [a picture of *e*] is therefore construed as incomplete (lacking an object for *of*) and as superfluous, as it cannot be incorporated coherently into the previously built structure. It should be noted that this case falls under Frazier's (1985) *Impermissible Ambiguity Constraint*, since the parser will systematically misanalyse strings such as this. This constraint assumes that the prohibition is captured in the grammar, perhaps as a reflex of processing difficulties. I make no such claim, however. Instead, I assume that there is a grammatical structure to be associated with the string in (99d), but that the parser is garden-pathed and therefore cannot recover the correct structure.

A similar argument can be constructed for the cases shown in 100 and 101. Where there is a potential filler, the parser attempts to resolve the dependency as soon as possible, proposing a trace in the object position of the verb. The verb has two possible subcategorization frames: $NP_{IO} NP_{DO}$ and $NP_{DO} [PP \text{ to } NP_{IO}]$. In the (a) case the parser encounters an NP after the verb; it is not until the second NP is encountered that the proposed trace analysis must be abandoned. In the (b) case, the trace analysis can be maintained, because the lack of a direct object is indicated by the presence of *to*.

Hawkins (1994) presents another argument against HNPS and PTI being derived by movement. Hawkins notes that the conditions under which HNPS apply are not statable in syntactic terms. HNPS is only licit when the shifted NP is of

sufficient “weight”.

An interesting dilemma arises when the processing difficulty in question involves terminal elements only and there is no appropriate dominating category in term of which a grammatical rule or principle can be formulated. An example is the so-called rule of Heavy NP Shift (cf. Ross 1976) exemplified in (2.2b):

- (2.2) a. I VP[gave NP[the valuable book that was extremely difficult to find] PP[to Mary]]
b. I VP[gave PP[to Mary] NP[the valuable book that was extremely difficult to find]]

This rule cannot be stated in terms of “Move NP,” because not all NPs undergo it, for example single-word NPs such as proper names and pronouns. Not can we propose some subcategory of “Heavy NP,” because such NPs cover a variety of structural types, from complex NPs at the one end that are structurally definable ($NP[NP \bar{S}]$) to those that are simply heavy in terminal elements on the other. But rules do not look at terminal strings – they apply to categories. More generally, heaviness is not a grammatical notion. Whether this “rule” applies at all depends as much on the length of the other constituents of the VP as it does on the length of the heavy NP (cf. ch. 4.3.2). The grammar cannot make a coherent response in these cases, therefore, and this is what explains a lot of confusion in the grammatical literature over how to handle alternations such as (2.2).

(Hawkins 1994, p. 20)

Hawkins compares his “heaviness” approach to Heavy NP Shift to the approach taken by (Rochemont and Culicover 1990), in which Heavy NP Shift is taken to be motivated by considerations of focus.

Notice that whatever focusing properties [Heavy NP Shift] has are not unique to these arrangements. Rochemont and Culicover (1990: 24) point out, for example, that the shifted structure (4.17) can be an appropriate response to the WH-question in (4.18a) but not to (4.18b):

(4.17) John gave to Mary a very valuable book.

(4.18) a. What did John give to Mary?

b. To whom did John give a very valuable book?

This is correct: but the unshifted (4.19) is an equally appropriate response to (4.18a),

(4.19) John gave a very valuable book to Mary.

and (4.19) is also an appropriate response to (4.18b). Therefore (4.17) is actually more limited in its focus options than the basic structure from which it is derived. It is also more limited in other ways: you cannot extract the center-embedded NP in the basic order:

(4.20) a. Who_i did John give a very valuable book to \emptyset_i ?

b. Who_i did John give to \emptyset_i a very valuable book?

c. What_i did John give \emptyset_i to Mary?

There are syntactic and semantic restrictions that need to be accounted for [...]. But they provide no motive for converting (4.19) into (4.17) in performance – on the contrary, they will block the rearrangements in many cases. Based on these considerations, Heavy NP Shift could just as well be called a “restricted focus construction” or a “restricted extraction construction”.

	Constituency	Island effects	Proper Binding Condition	Empty Category Principle	Long range	Parasitic gap
Right Node Raising	X	X				
Extraposition			X	X	X	
Heavy NP Shift/PTI					X	X

Table 3.1: Failed diagnostics for rightward movements

Do these considerations force an obligatory focus interpretation on the NP, as Rochemont and Culicover claim? I see no evidence that they do. The matter is hard to decide, given the slipperiness of the fundamental concept, focus, which is never adequately defined.

(Hawkins 1994, p. 185–6)

3.3 Stating thesis

I have presented arguments in support of the view that overt rightward syntactic movement, as opposed to rightward displacement, does not exist, summarized in table (3.3).

My thesis is that the leftwardness of overt movement derives from the left-to-right nature of on-line processing. I show in the next chapter that any purely syntactic (declarative) approach is in essence stipulatory, and offers no explanation of *why* overt movement is leftward.

Chapter 4

Possible solutions

In this chapter I present several accounts for the linear asymmetry observed in overt syntactic movement. The accounts considered fall into one of two classes. One class consists of syntactic accounts, while the other consists of a processing account.

Within a theory of syntax, the linear asymmetry of movement can be expressed in a number of ways. A constraint on the directionality of movement can be incorporated directly into the definition of movement. Another possibility is to leave the operation of movement oblivious to linear order but ensure that the target position of movement is always to the left of the source position. Yet another approach is to define a function which maps a strictly hierarchical phrase structure representation into a linear ordering of the terminals in the phrase structure tree in such a way that the target position of movement precedes the source position.

In processing terms the linear asymmetry of movement is naturally attributed to the processing of filler-gap dependencies. I discuss two processing models. While neither model sets out to account for the leftwardness of movement, they both strongly suggest a solution. I discuss a particular strength and weakness of each model, which sets the stage for a discussion of my algorithm in chapter 5. My

algorithm builds on both of these processing models.

4.1 Constraining the movement operation

One can account for the leftwardness of movement by stipulating that the movement operation must place the moved constituent to the left of its pre-movement position. Since the movement operation is part of Universal Grammar,¹ a linearity constraint built into the movement operation applies cross-linguistically. This is a desirable consequence. However, this benefit must be weighed against the fact that there is no independent justification to make the movement operation sensitive to the linear order of constituents.

Incorporating a linearity constraint into the movement operation presupposes both that syntactic structure encodes linear order and that the movement operation can make reference to linear order. The assumption that syntactic structure encodes the linear order of elements is not as uncontroversial as it might at first appear. Recent developments of Principles and Parameters type theories, such as Chomsky's (1995) *Minimalist Program*, employ phrase-structure representations which are set-based. As such they encode only containment relations, and no other relations. These types of theories are discussed in section 4.3. Hence it is by no means universally accepted that phrase structure encodes linear order.

The assumption that the movement operation makes reference to linear order lacks motivation. Beyond enforcing the leftwardness of movement there is no reason to assume that the movement operation is sensitive to linear order. The diagnostic

¹Universal Grammar is an encoding of the syntactic competence which humans come "pre-wired" with. It can be thought of as a template for possible human grammars.

properties of movement² make no reference to linear order.

Summarizing, there does not appear to be any independent motivation for the assumption that the movement operation is sensitive to linear order. This is simply a stipulation which offers a description of the facts, but which offers no explanation of them. For these reasons I set aside this approach.

4.2 Constraining the target of movement

A somewhat more plausible approach is to claim that movement is leftward because the target position of movement is always situated to the left of its source position. In this case the movement operation need not make any reference to linear order. In this section I explore how such a constraint could be expressed. I discuss two theories of phrase structure, both of which assume that phrase structure representations encode linear order relations in addition to hierarchical relations between nodes. The two theories I discuss are a traditional \bar{X} theory and the Immediate Dominance/Linear Precedence (ID/LP) theory of phrase structure.

Both the \bar{X} and ID/LP theories are theories of phrase structure rules: they restrict the form of allowable context-free grammars. Whereas a context-free grammar can only express linear order constraints within the right hand side of a rule, \bar{X} and ID/LP theories allow constraints on phrase structure to be applied to more than one rule at a time. In preparation for a discussion of \bar{X} and ID/LP theories I consider how ordinary context-free grammars fall short in their ability to express word order generalizations in natural language.

²These are properties (2a–f) of chapter 1, and are discussed in section 2.2.10.

4.2.1 Context free grammars

Formally, a context free grammar is a 4-tuple (T, N, R, S) where T is a set of terminal symbols, N is a set of nonterminal symbols, R is a set of rules (also called productions), and $S \in N$ is the start symbol. Rules in R have the form $\alpha \rightarrow \beta$, where $\alpha \in N$, and $\beta \in (T \cup N)^*$, $|\beta| \geq 1$.

Consider the set of context free grammar rules given in (102).

(102) $S \rightarrow NP VP$

$S \rightarrow NP_{wh} S_{missingNP}$

$S_{missingNP} \rightarrow Aux VP$

$S_{missingNP} \rightarrow Aux NP VP_{missingNP}$

$NP \rightarrow Det N$

$NP_{wh} \rightarrow who$

$NP_{wh} \rightarrow what$

$VP_{missingNP} \rightarrow VP_{missingNP} Adv$

$VP_{missingNP} \rightarrow V_{transitive}$

$VP \rightarrow VP Adv$

$VP \rightarrow V_{transitive} NP$

$VP \rightarrow V_{intransitive}$

$N \rightarrow emu$

$N \rightarrow penguin$

$V \rightarrow see$

$Det \rightarrow the$

$Aux \rightarrow did$

$Adv \rightarrow yesterday$

This grammar accepts sentences such as *Who did the emu see yesterday*, but not sentences like *did emu see yesterday who*. The rules of the grammar are written in such a way that NP_{wh} is generated to the left of the $S_{missingNP}$.

These rules offer no explanation of why a grammar should choose this order over another, however. It is just as easy to formulate the rule in (103) as the rule in (104). The grammar itself cannot express a constraint which governs the order of constituents on the right hand side of these rules.

$$(103) \quad S \rightarrow S_{missingNP} \ NP_{wh}$$

$$(104) \quad S \rightarrow NP_{wh} \ S_{missingNP}$$

Using a context-free grammar it is also not possible to express a relationship in constituent orders in different rules. For example, a context-free grammar cannot express an hypothesis of the form that if a language has a rule of the form (104), then it will have a rule of the form (105) rather than (106).

$$(105) \quad T \rightarrow NP \ S_{missingNP}$$

$$(106) \quad T \rightarrow S_{missingNP} \ NP$$

This failure of ordinary context free phrase structure rules to adequately capture regularities of natural language word order stems from the fact that each rule exists in isolation. A context-free grammar has no mechanism to express a relationship between rules in a grammar. A context-free grammar is clearly also unable to express a relationship between rules in different grammars. Ordinary context free grammars therefore cannot express the fact that in *any* language overt movement can only be to the left. \bar{X} and ID/LP theories represent different attempts to address these shortcomings of context-free grammars.

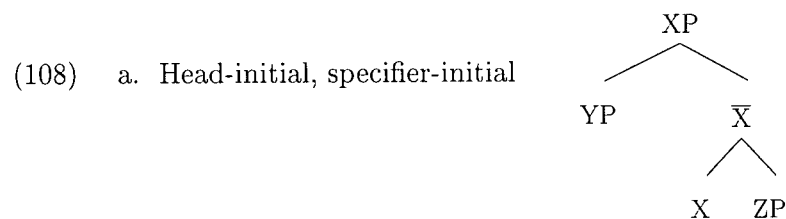
4.2.2 \bar{X} theory

\bar{X} theory is a set of constraints on the expressive power of context free rules.³ \bar{X} theoretic constraints are typically expressed in terms of underspecified phrase structure templates, such as those shown in (107).

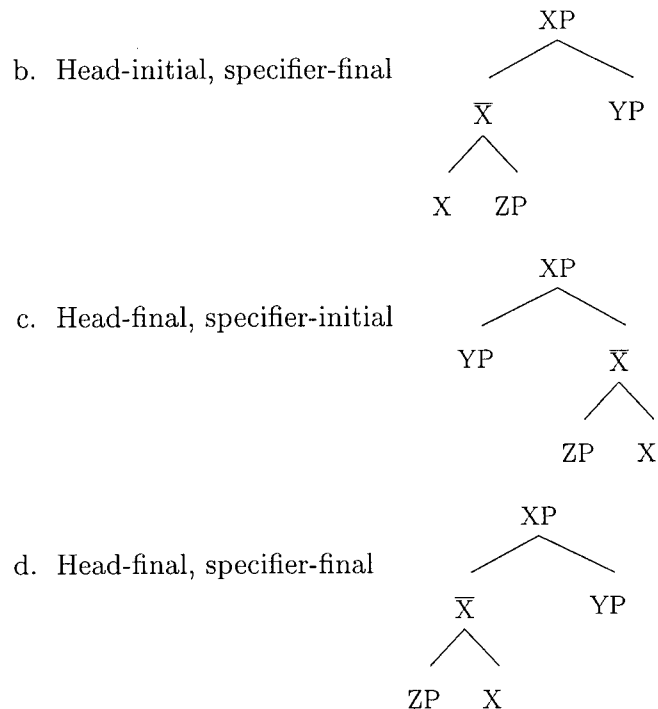
$$\begin{aligned} (107) \quad & XP \rightarrow \bar{X} \ YP \\ & \bar{X} \rightarrow X \ ZP \\ & XP \rightarrow XP \ WP \end{aligned}$$

X is called the *head* of the projection, YP the *specifier*, and ZP the complement. WP is called an *adjunct*.

\bar{X} theory imposes a uniform hierarchical structure on all phrases. \bar{X} theory permits variation in the linear order of heads, specifiers and complements across categories (i.e. a noun phrase may have a different structure than a verb phrase), but requires uniformity within a given category (i.e. all nominal projections have the same structure). The permissible variation is specified using parameters. One parameter specifies whether heads precede or follow their complements. A second parameter indicates whether or not the specifier precedes or follows its sister. The four possible (non-adjunction) phrasal structures are shown in (108).



³Recall the discussion of \bar{X} theory in chapter 2.



In order to account for the leftwardness of overt movement it can be required that the target of overt movement comes before its source, the position from which movement takes place.

What types of position are targets of movement? Movement can either be to a specifier position (109a) or to an adjoined position (109b).

- (109) a. Who did the penguin chase?
 b. The emu, the penguin chased.

If all specifiers are generated to the left of their sisters then overt movement to specifier position is movement to the left. This can be specified with existing \bar{X} parameters, but is descriptive rather than explanatory.

Adjuncts present a problem since there is no parameter in the theory which regulates the position of adjuncts. Adding a parameter which specifies the directionality of adjuncts is not enough, since the branching direction of adjuncts in

a language is not typically uniform. What is required is a mechanism to fix the position of those adjunct positions which are targets of overt movement.

This approach leaves unanswered the question of why these positions must appear to the left of their sisters. Furthermore, in order for this approach to make cross-linguistic predictions it must be that the linear order of targets of overt movement with respect to their sisters is specified for all languages in Universal Grammar. In other words, this is simply part of our innate knowledge of language. For the same reason that it is undesirable to simply stipulate that movement is leftward, it is undesirable and ad-hoc to simply require that targets of movement precede their associated source positions, in the absence of any independent motivation.

4.2.3 ID/LP rules

Generalized Phrase Structure Grammar (GPSG) (Gazdar, Klein, Pullum and Sag 1985) divorces immediate dominance (ID) statements from linear precedence (LP) statements. Generalizations about the linear order of constituents can therefore be expressed for a grammar as a whole, rather than for a single rule (as in a CFG), or for a category and type of position (as in \bar{X} theory). As an example, consider the (context free) grammar in (110), from Gazdar et al. (1985, pg. 47).⁴

- (110) $S \rightarrow NP VP$
 $S \rightarrow AUX NP VP$
 $VP \rightarrow AUX VP$
 $VP \rightarrow V VP$
 $VP \rightarrow V NP$
 $VP \rightarrow V NP VP$
-

⁴In order to simplify the discussion this grammar lacks terminal symbols.

This grammar has several interesting properties. It is true of every rule in (110) that AUX precedes NP, V precedes NP, and NP precedes VP. These are in no way necessary properties of this grammar: the rules might have been otherwise. If this set of rules represents a fragment of a larger grammar, it is impossible to know without inspecting the whole grammar whether these properties are true of the grammar as a whole or only of this small fragment.

A generalized phrase structure grammar permits generalizations such as these to be expressed for a grammar as a whole. Formally, a generalized phrase structure grammar is a 5-tuple (T, N, D, P, S) , where T , N and S are as for a context free grammar.

D is a set of *immediate dominance* (or ID) rules. Whereas the order of items on the right hand side of a context free rewrite rule is fixed, the order in an ID rule is not fixed. The ID rule⁵

$$(111) \quad S \rightarrow \{NP, VP\}_m$$

carries the same information as the following set of context free rules

$$(112) \quad S \rightarrow NP \ VP$$

$$S \rightarrow VP \ NP$$

P is a set of *linear precedence* (or LP) constraints. An LP constraint is a statement of linear order which applies to all rules in the grammar. LP constraints are stated in terms of the relation \prec . A constraint of the form

$$(113) \quad NP \prec VP$$

⁵The right hand side of the rule consists of a *multiset*. A multiset shares with a set that the order of elements is irrelevant. In contrast to a set a multiset may contain duplicate elements. In other words, while $\{A, B\} = \{A, B, B\}$, $\{A, B\}_m \neq \{A, B, B\}_m$.

states that NP does not occur after VP in any ID rule. This LP constraint together with the ID rule

$$(114) \quad S \rightarrow \{NP, VP\}_m$$

carries the same information as the following context free rule

$$(115) \quad S \rightarrow NP \ VP$$

Given a set of LP constraints, each LP constraint must all be true of every ID rule in the grammar. A set of LP constraints is interpreted as being closed under transitivity.

An ID/LP grammar which is extensionally equivalent to the grammar of (110) is given in (116). It is from Gazdar et al. (1985, pg. 48).

$$(116) \quad \begin{array}{ll} \text{a. } S \rightarrow \{NP, VP\}_m \\ \quad S \rightarrow \{AUX, NP, VP\}_m \\ \quad VP \rightarrow \{AUX, VP\}_m \\ \quad VP \rightarrow \{V, VP\}_m \\ \quad VP \rightarrow \{V, NP\}_m \\ \quad VP \rightarrow \{V, NP, VP\}_m \\ \text{b. } AUX \prec NP \\ \quad V \prec NP \\ \quad NP \prec VP \end{array}$$

The LP constraints of this grammar encode the generalizations noted above. The LP constraints apply to the whole grammar (thus to all of the ID rules). Since the LP constraints are interpreted as transitively closed they implicitly impose the following constraints,

(117) $AUX \prec VP$

$V \prec VP$

ID/LP rules express word order constraints for the grammar as a whole, rather than on a rule-by-rule basis.

... ID/LP format grammars capture generalizations by stating constituent order for the grammar as a whole, rather [than] on a one-rule-at-a-time basis as in [context-free phrase structure grammars].

(Gazdar et al. 1985, pg. 46)

ID/LP grammars cannot express contradictory orderings in different rules. For example, the follow set of phrase structure rules cannot be expressed directly⁶ in the ID/LP format because first rule requires the order $NP \prec VP$ while the last requires $VP \prec NP$, from Gazdar et al. (1985, pg. 48),

(118) $S \rightarrow NP VP$

$S \rightarrow AUX NP VP$

$VP \rightarrow AUX VP$

$VP \rightarrow V VP$

$VP \rightarrow V NP$

$VP \rightarrow V VP NP$

ID/LP grammars are able to express many generalizations which ordinary phrase structure rules cannot. Even so, it is not clear why one ordering should be chosen over another. There is no explanation of why a language should not choose rightward movement over leftward movement.

⁶By directly I mean without introducing additional nonterminal categories into the grammar.

Cross-linguistic generalizations are not expressible, since the ID/LP rule schemata form part of a single grammar. A single set of LP constraints cannot simply be made to apply to all grammars, since word order in general differs cross-linguistically.

4.2.4 Summary

Neither \bar{X} theory nor ID/LP theory seems capable of offering an explanation for the leftwardness of movement across languages.

4.3 Constraining the structure to order mapping

Kayne (1994) advances an interesting proposal to account for word order. Kayne proposes that the mapping from hierarchical structure to linear order must satisfy the Linear Correspondence Axiom, given in the following quote. In this quote, the function d maps a nonterminal of a phrase structure tree to the set of terminals which it dominates. Kayne further defines

$$d\langle X, Y \rangle = d(X) \times d(Y)$$

and where

$$S = \{\langle X, Y \rangle \mid X, Y \text{ are nonterminals in the same phrase structure tree}\}$$

we have

$$d(S) = \bigcup_{\langle X, Y \rangle \in S} d\langle X, Y \rangle$$

Kayne's proposal is as follows:

To express the intuition that asymmetric c-command is closely matched to the linear order of terminals, let us, for a given phrase marker, consider the set A of

ordered pairs $\langle X_j, Y_j \rangle$ such that for each j , X_j asymmetrically c-commands Y_j . Let us further take A to be the maximal such set; that is, A contains all pairs of nonterminals such that the first asymmetrically c-commands the second. Then the central proposal I would like to make is the following (for a given phrase marker P , with T the set of terminals and A as just given):

(3) *Linear Correspondence Axiom*

$d(A)$ is a linear ordering of T .

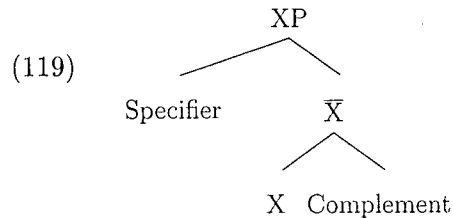
(Kayne 1994, pp. 4–5)

The LCA, together with the assumption that asymmetric c-command maps into linear precedence via the function d , requires that the c-command relation map into linear precedence. Since movement is constrained to be upward (the target must c-command the extraction site) a consequence of Kayne’s theory is that all movement is leftward.

A detailed analysis and critique of Kayne’s proposal is beyond the scope of this dissertation. In this section I raise some issues with respect Kayne’s approach. In chapter 8, after having presented and discussed my parsing algorithm, I briefly revisit Kayne’s proposal.

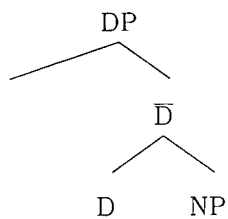
4.3.1 Representing word order differences

In Kayne’s theory all phrases have the structure shown in (119).

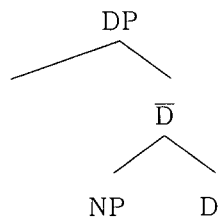


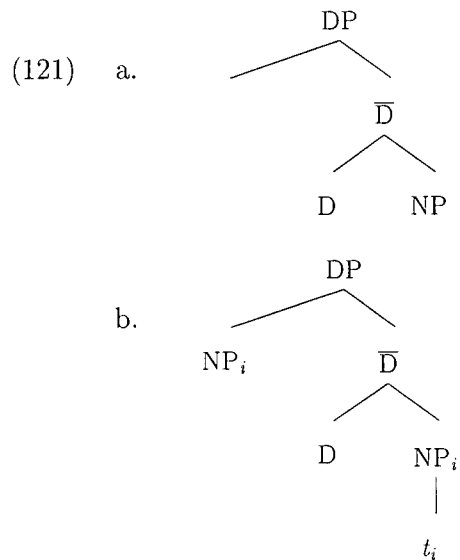
In Kayne's theory, word orders which deviate from this prescribed basic order are derived via movement. The standard (non-Kaynean) phrase structure approach differentiates between basic word order variation and displacement. Basic word order variation is expressed by different parameter settings. Projections with parametrically different word orders share the same hierarchical structure. An example of this is shown in (120a-b). In Kayne's system not only must the fact that there is a difference in word order be specified somehow, there is a different structure associated with differences in word order. The structures corresponding to (120a-b) in Kayne's system are shown in (121a-b). (121b) is derived from (121a) by movement of the NP from the complement of D position to the specifier of D position.

(120) a. Head-Complement order



b. Complement-Head order





Whether or not there is a movement involved in the derivation is potentially testable. Psycholinguistic experiments can be carried out to test for the presence of an antecedent-trace relationship. A positive result for a structure such as (121b) as opposed to (121a) would lend support to Kayne's approach, while a negative result would cast doubt on the psychological reality of the dependencies which Kayne's approach relies on.

4.3.2 Deriving different word orders

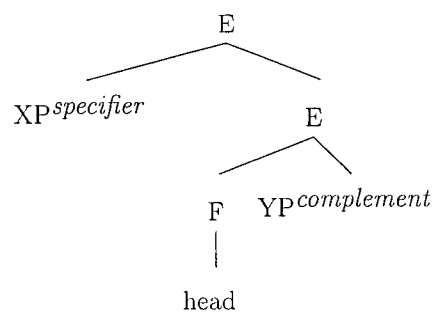
The LCA by itself does not rule out any word order possibilities. The LCA simply determines what the word order associated with a given hierarchical structure must be. In (123)–(128) I show some ways in which all possible orderings of a specifier, head and complement can be derived.

In particular, the LCA alone cannot rule out what would appear to be rightward movement, as in (122a–b). What is required is a theory of possible landing sites of movement and a theory of what motivates movement.

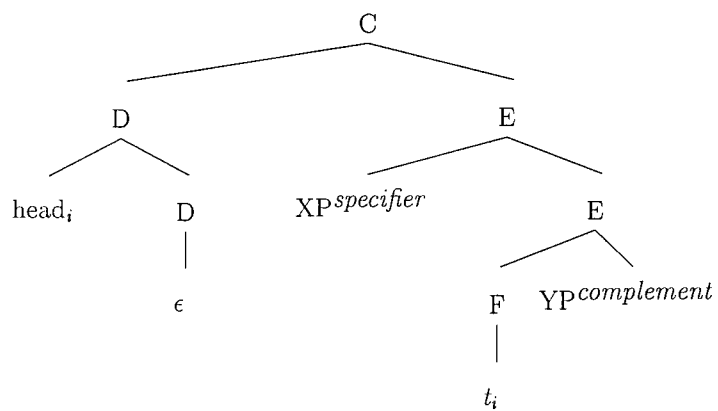
(122) a. Bill saw yesterday who.

b. Saw Bill yesterday who.

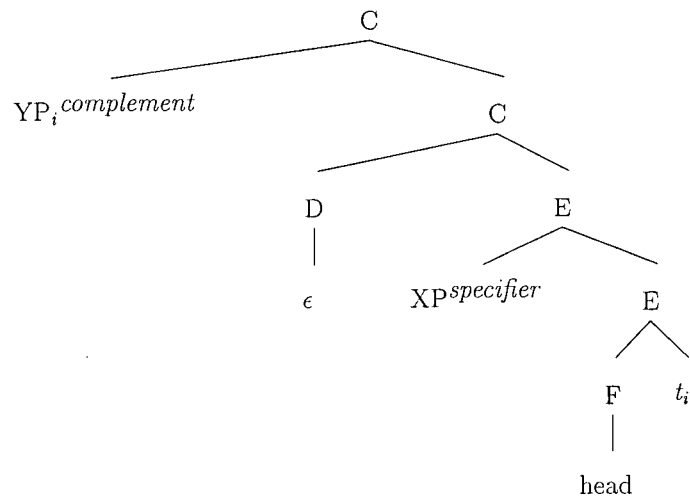
(123) $XP^{specifier}$ head $YP^{complement}$



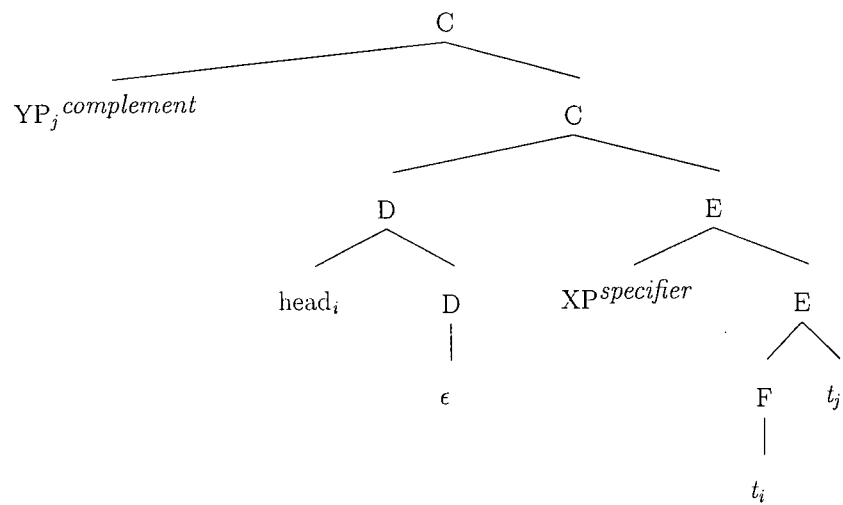
(124) head $XP^{specifier}$ $YP^{complement}$



(125) $YP_{i complement}$ $XP_{specifier}$ head

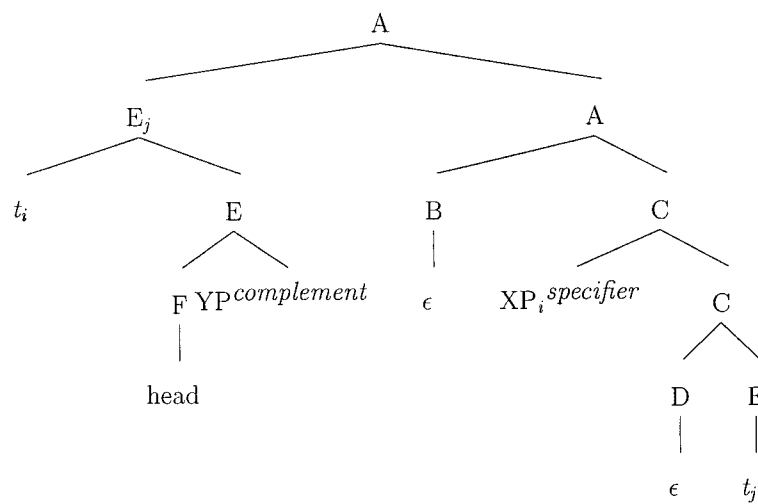


(126) $YP_{j complement}$ head $XP_{specifier}$

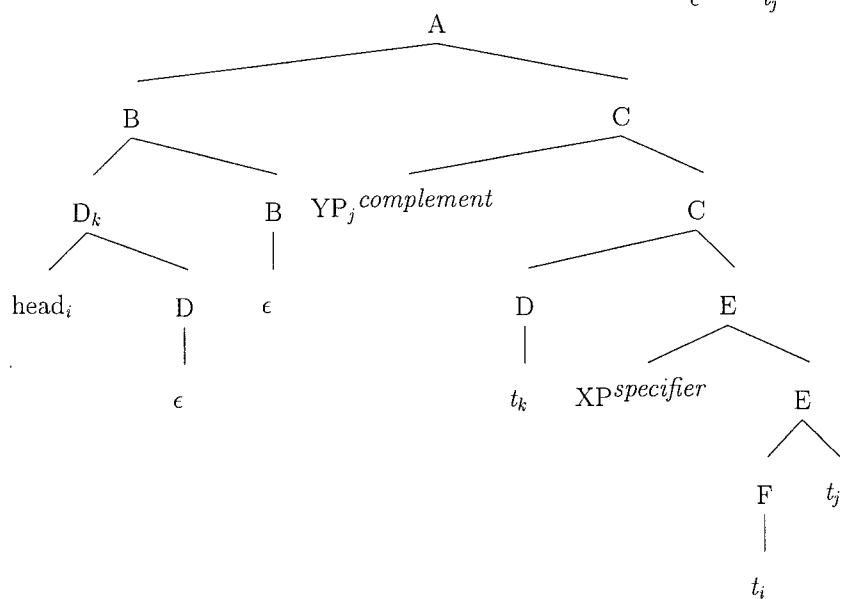


(127) head $YP^{complement}$ $XP^{specifier}$

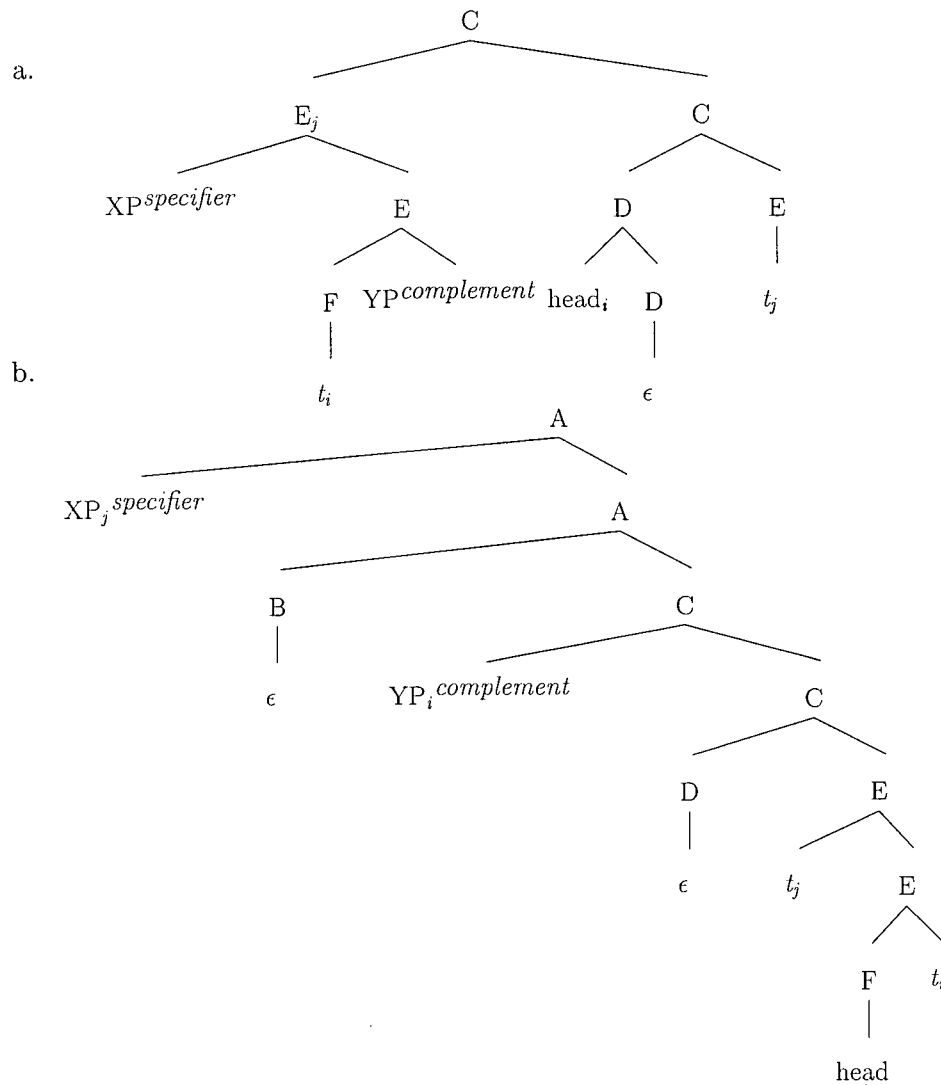
a.



b.



(128) $XP^{specifier} \ YP^{complement} \ head$



4.4 Constraining the processing of movement

From a processing perspective a natural approach is to attempt to derive the leftwardness constraint on movement from the manner in which filler-gap dependencies are processed. I present two models of filler gap processing, and suggest how models such as these can be used to derive the linearity constraint on overt movement.

I first consider Fodor's (1978) gap-driven model of filler-gap processing. Frazier (1987b) and Frazier and Flores D'Arcais (1989) reject this gap-driven model in favour of a filler-driven one, the Active Filler Strategy (AFS), because the gap-driven model makes incorrect predictions with respect to the processing of English. I show that the AFS also does not make correct predictions regarding the processing of English, and argue that the difficulty with Fodor's model is not that it is gap-driven, but rather than it makes use of lookahead (it is a so-called gap-as-second-resort model). This sets the stage for the discussion in chapter 5.

4.4.1 A gap-driven model

In chapter 2 I discuss two dimensions along which filler-gap processing strategies are distinguished. A filler-gap processing strategy can be classified as being either *filler-driven* or *gap-driven*. It can also be classified as being either *first resort* or *last resort*.

Fodor (1978) argues against first resort models of filler-gap processing on the basis that they are bound to make numerous mistakes during processing.

As Wanner and Shiner (ms.) have observed [...] a first resort parser might also waste a great deal of effort by hypothesizing a gap without first looking to see what appears next in the sentence.

(Fodor 1978, p. 435)

To illustrate this point, consider the sentence in (129) below (Fodor 1978, p. 436) as an example.

- (129) *Who_i* did Tom ask Meg to persuade Jill to inform Ted that Bob had spoken to Δ_i ?

There are five noun phrase positions between the filler *who* and its actual gap position (indicated by Δ). Each of these five noun phrase positions is filled by an overt noun phrase (*Tom, Meg, Jill, Ted* and *Bob*).

A parser that hypothesized a noun phrase gap in each of these positions before checking to see whether a noun phrase already appears there would lose time on five hypotheses that it could immediately have determined to be false just by looking at the next word.

(Fodor 1978, p. 436)

Fodor therefore suggests that a parser should make use of a limited amount of lookahead to ensure that a gap is not hypothesized in those cases where this expectation is immediately disconfirmed.

Let us therefore consider a model of the parser according to which it will hypothesize a gap for a filler of category X only in positions whose left context is compatible with the occurrence of a phrase of category X, and only in positions that are not immediately followed by a phrase of category X.

(Fodor 1978, p. 436)

Fodor refers to this as a (gap-as-)second resort strategy. According to the typology of chapter 2, this is a gap-driven first resort with lookahead strategy. It is the use of lookahead which makes the parser consider attaching an available constituent before postulating a gap.

While this is a reasonable strategy, psycholinguistic evidence discovered by Crain and Fodor (1985) and duplicated by Stowe (1986), the *filled-gap effect*, shows that people in fact do entertain gap hypotheses which are disconfirmed almost immediately.

4.4.2 The Active Filler Strategy

Frazier (1987b) and Frazier and Flores D'Arcais (1989) propose that filler-gap dependencies are parsed according to the Active Filler Strategy (AFS), stated in (130).

- (130) *Active Filler Strategy.* Assign an identified filler as soon as possible; i.e. rank the option of a gap above the option of a lexical noun phrase within the domain of an identified filler.

The AFS accounts for the presence of a filled-gap effect in object positions in English. Once a filler has been identified as such, the parser tries to resolve the filler-gap dependency as soon as possible. However, the AFS also predicts a filled-gap effect in subject position. According to Frazier and Flores D'Arcais (1989), the AFS requires that,⁷

... fillers are assigned to the first possible position in a syntactic phrase marker.
[...] This predicts that the filler should be assigned to the leftmost position from which it might have been extracted. Hence, on this "filler driven" account of gap filling, the perceiver will initially try to assign "who" to the subject position in [Who did John see ____?] This hypothesis will need to be abandoned immediately upon receiving the phrase "John".

(Frazier and Flores D'Arcais 1989, p. 332)

⁷Frazier and Flores D'Arcais (1989) claim that the AFS predicts the presence of a filled-gap effect in subject position. The example they use to demonstrate this point is problematic because it contains clear syntactic cues which indicate that extraction from subject position could not have taken place (e.g subject-auxiliary inversion). A better set of examples is,

- (i) I know who you like ____.
- (ii) I know who John saw ____.

In these examples it is conceivable that a parser following the AFS would mistakenly postulate a gap in the subject position following the filler *who*.

This misanalysis does in fact happen with object position (Crain and Fodor 1985), but does not occur with subjects (Stowe 1986). The AFS thus makes an incorrect prediction for the processing of English.

The AFS faces further challenges. It is unclear what happens if a constituent is mis-identified as a filler when it is not, as might happen in the case of an initial NP-NP sequence (131), which can continue in any of the ways in (132).⁸

(131) The boy Joanne

- (132) a. The boy Joanne admires won the lottery.
b. The boy Joanne admires but the girl she is contemptuous of.
c. The boy Joanne admires him a great deal.

Only in (132b) is *the boy* a filler. It seems inevitable that *the boy* will be incorrectly identified as a filler when it is not, or will not be identified as a filler when it is. How the AFS handles situations such as this is unclear. The Minimal Chain Principle, discussed in more detail in section 7.1, suffers the same weakness.

4.4.3 Summary

The model advanced by Fodor predicts that there should be no filled-gap effects in English, while the model advanced by Frazier and Flores D'Arcais predicts that there should be filled-gap effects with both subjects and objects in English. Both of these predictions are incorrect.

The AFS is a filler-driven strategy for recovering filler-gap dependencies.

⁸Intonational cues will not suffice to disambiguate these examples three ways. The topicalization example (132b) cannot be differentiated from both (132a) and (132c) simultaneously. Only in (132b) is *the boy* a filler.

Following the AFS a parser will be able postulate a gap only once a filler has been identified. It therefore follows that a filler must precede its gap in order for the dependency to be recovered. Attributing the observed linearity constraint on overt movements to the AFS seems to be a very promising.

The AFS does not offer a complete answer, however. The AFS does not address those cases in which a constituent is mis-identified as a filler when it is not, or in which a constituent is not identified as a filler when it should be. The AFS also does not address the larger issue of the processing of a full range of empty categories.

The AFS addresses the shortcomings of Fodor's model by adopting a a filler-driven rather than a gap-driven processing strategy. While Fodor's gap-as second resort model does not offer a clear reason for preferring leftward to rightward movements, in chapter 5 I present a gap-driven processing model which does. A key difference between the processing model I present and Fodor's is that my model does not employ lookahead.

Chapter 5

A parsing solution

In this chapter I develop a typology of nominals which forms the foundation of my solution to the problem outlined in chapter 3. I also develop a parsing algorithm which exploits the structure of the typology in processing empty categories. A difficulty in parsing both overt and empty nominals in a uniform manner is that not all relevant properties of the nominal can be determined from the position in which it is initially encountered by the parser. Since a phrase can be licensed in several different positions, the parser must identify all of the licensing relations which a phrase participates in before it can fully identify the grammatical role of the phrase. The key of my algorithm is that a constituent can be underspecified with respect to its feature content during parsing. This allows the parsing algorithm to divorce the question of the presence or absence of an empty category (licensing) from the question of the content of an empty category (identification) (Rizzi 1986). The basic idea is that the problem of licensing and identification of empty categories during on-line processing should be thought of as a case of ambiguity resolution. I employ an underspecification approach, as espoused by Marcus et al. (1983) and Gorrell (1995), to permit the parser to resolve empty category ambiguities incrementally as

parsing proceeds while maintaining deterministic operation (Marcus 1980).

5.1 Nominal subcategories

In this section I examine the standard typology of *empty categories* (ECs), focusing on the two non-trace empty categories, *PRO* and *pro*, which it admits.¹ I propose an alternate, more fine-grained typology of non-trace empty categories based on how nominals are licensed. The structure of this licensing typology plays an important role in my parsing algorithm. It leads to an answer to two central questions. The first asks how the parser determines whether a given position belongs to an empty category. The second asks which empty category the given position belongs to.

The binding theory makes a four-way distinction amongst nominal elements. Nominals are distinguished along the *pronominal* and *anaphoric* dimensions. The standard binding theory typology is shown in (133).

(133)

anaphoric	pronominal	overt	covert
+	+		PRO
+	–	reflexives and reciprocals	NP trace
–	+	pronouns	pro
–	–	R-expressions	<i>wh</i> trace

The features which define the categories make no reference to phonological content (or lack thereof).

The most plausible general assumption is that the typology of ECs simply

¹The term *empty category* is in fact a misnomer. It refers to a category whose elements are phonetically empty, and not to a category which is empty. The term has become standard, however, and I will not depart from the standard usage here.

mirrors that of overt categories, that is, that no new principles are invoked to determine the types of EC.

(Chomsky 1982, pg. 78)

There may be other distinctions, beyond those made by the binding theory, which are relevant.

We would therefore expect to find ECs of each of the four types [in the typology given in (133)] (unless some are barred by independent principles) and no other types. Furthermore, if some category of [133] is subdivided for overt elements (for instance, pure pronominals can be referential or pleonastic), we would expect to find the same subdivisions for ECs.

(Chomsky 1982, pg. 79)

One strategy for identifying classes of empty nominals is simply to identify classes of overt nominals. Any syntactically relevant subclass of overt nominals should have a covert counterpart. Taking this approach to the extreme one might suppose that every overt nominal element has a covert counterpart.² What is important in the present context, however, is to find a level of abstraction which makes distinctions useful for parsing. A typology so fine-grained as to make distinctions at the level of individual lexical items is not likely to be relevant during syntactic parsing. On the other hand the binding theory typology, which makes available only two non-trace empty categories, is not fine-grained enough. My first task is therefore to develop a typology of nominals which is motivated both from the perspective of the parser and syntactic theory.

²The notion that every overt noun phrase has a covert counterpart is not as outlandish as it might seem. This is in effect what is proposed in the copy theory of movement. A trace is viewed as a phonetically empty copy of the moved element.

The parser cannot determine the binding theory status of a phonetically null element by examining the element's intrinsic feature content. However, the parser can deduce which empty category the element belongs to by taking into consideration the syntactic context in which it appears. In this subsection I show how the θ , Case and operator features of a position constitute a sufficient context for empty category identification.

For the rest of this section when I refer to an empty category I am excluding traces from consideration, unless explicitly included.

5.1.1 θ marking

The distinction between *expletive*³ and *non-expletive* noun phrases is directly relevant to the binding theory classification. Following Chomsky (1981) I assume that expletives are non-referential, and therefore must not be bound.⁴ Expletives must hence be treated the same as R-expressions and not as either pronouns or anaphors. Expletives are therefore subject to condition C of the binding theory.⁵ In (134a) *there* is an expletive element. *Some girls* functions as the subject, and alongside (134a) we find (134b) in which there is no expletive, and the subject noun phrase is in its normal pre-verbal position.

(134) a. There arrived some girls.

b. Some girls arrived.

³I assume that the terms *pleonastic* and *expletive* can be used interchangeably.

⁴For relevant discussion, see pages 35 and 101 of (Chomsky 1981).

⁵Condition C applies to those elements which must not be bound. Neither fully referential expressions nor non-referential expressions may be bound. From the perspective of the binding theory they fall into an equivalence class, and are both subject to condition C.

Hence, we might expect that there is a covert counterpart to the English expletive *there*. Indeed, many languages have a covert expletive. Examples (135a–b) show the Italian null-expletive equivalents of (134a–b).

- (135) a. Sono venute delle ragazze.
 pro are come some girls
 There arrived some girls
- b. Delle ragazze sono venute.
 some girls are come
 Some girls arrived

The traditional analysis of expletives holds that they are Case marked but not θ marked.⁶ Ordinary argument noun phrases require both Case marking and θ marking. Alongside the English example (136) there is the Italian example (137).

(136) You speak.

(137) Parli.
 pro speak
 “You speak”

The θ requirement of a noun phrase therefore makes a relevant syntactic distinction in the case of overt noun phrases. This distinction is mirrored in the realm of empty noun phrases, as shown in (138).

(138) Argument and expletive noun phrases

	overt	covert	BT condition
$+\theta$	argument NP	argument <i>pro</i>	condition B
$-\theta$	expletive NP	expletive <i>pro</i>	condition C

⁶The analysis of expletives in the minimalist framework differs from the analysis presented here. For discussion, see Chomsky (1995), Lasnik (1995), den Dikken (1995) and Groat (1995).

5.1.2 Case marking

In the previous section I showed that the θ marking requirements of nominals distinguish argument NPs from expletive NPs. In this section I consider the distinctions made by the Case marking requirements. Case properties have no import on the classification of overt nominals, since all overt nominals must bear Case. Because of the interaction of binding theory and Case theory, covert empty categories can be distinguished by their Case properties, however. Members of the category [+anaphor,+pronominal] must not be Case marked, while members of [−anaphor,+pronominal] must be Case marked.

(139) Case properties of empty noun phrases

	BT features	covert
+Case	[+pronominal, −anaphoric]	<i>pro</i>
−Case	[+pronominal, +anaphoric]	<i>PRO</i>

Case and θ properties, when taken together, identify three relevant categories from the perspective of the parser. These are summarized in (140). Recall that for overt noun phrases [−Case] is ruled out by the Case filter. A negative specification for both Case and θ marking indicates nonexistence of a position.

(140)

Case	θ	Overt	Covert
+	+	argument NP	argument <i>pro</i>
+	−	expletive NP	expletive <i>pro</i>
−	+		<i>PRO</i>
−	−		

Notice that during parsing it is unclear whether a *pro* in the specifier of I position (the VP-external subject position) will be interpreted as an argument or

an expletive in the final analysis (compare (135a) with 137). Both occupy the same structural position. The argument/expletive status is determined according to whether the empty category can be interpreted with respect to a θ position.⁷ Because arguments and expletives are subject to different binding conditions, it is inappropriate to group them under one and the same banner (*pro*).

By using the Case and θ marking as contextual cues for the licensing and identification of empty categories a three-way distinction amongst pronominals has been identified.

5.1.3 Operator status

θ and Case marking serve to distinguish between different types of nominals in A positions. In this section I turn to \bar{A} positions. In \bar{A} positions we are interested in the operator status of nominals. (141)–(143) define *operator* in syntactic terms.

(141) **Definition** A constituent is an *operator* if and only if (i) it appears in an \bar{A} position at LF, and (ii) it binds a variable.

(142) **Definition** A constituent is a *variable* if and only if it is in an A position and it is \bar{A} bound.

(143) **Definition** α *binds* β if and only if (i) α c-commands β , and (ii) α is co-indexed with β , and (iii) there is no γ such that α binds γ and γ binds β .

I discuss work by Rizzi (1990) and Tsai (1994). Rizzi presents an account of the distributional differences of complementizers in interrogative and relative

⁷This foreshadows a change in perspective whereby licensing requirements are taken to be features of chains, not of individual constituents of a phrase structure tree.

clauses, as well as the type (interrogative, declarative, relative) of clauses.⁸ Tsai addresses the distinction between *wh*-movement languages such as English and non-*wh* movement languages such as Chinese.

In this section I discuss specifically *wh*-operators: interrogative and relative pronouns. It is beyond the scope of this thesis to give an in-depth analysis of the diversity of *wh*-operators found cross-linguistically. I put aside noun phrases which act as operators by virtue of having been topicalized. These noun phrases are not marked with a particular feature which requires them to be topics. In English at least, topicalized noun phrases are not morphologically distinct from other noun phrases.⁹ I discuss how these noun phrases are processed in section 5.2.2. Since there is no morphological distinction made in the overt nominals between topics and non-topics, I do not expect that there should be one in the covert nominals either.

The definitions given in (141)–(143) depart somewhat from the standard definitions. The departure lies in the definition of variable. The standard notion is that a variable is a Case marked trace (i.e. a trace of \bar{A} movement). This seems to be too restrictive, however. It is not clear that every variable is Case marked. Consider subject infinitival relative clauses in English, as in (144a). Browning (1987), following Williams (1980), analyses subject infinitival relatives as shown in (144b). *PRO* is taken to be the predicate variable of the relative clause. *PRO* is not a trace and does not bear Case, but *PRO* is θ marked. Hence, it is not correct to say that

⁸Cheng (1991) also discusses clausal typing.

⁹There are intonational cues which signal that a noun phrase should be interpreted as a topic, but these are not inherent properties of a noun phrase, unlike the unique morphological form of *wh*-phrases.

variables must bear Case.^{10,11}

¹⁰It has been proposed that *PRO* does bear a form a null Case (Chomsky 1995). However, a Case marked *PRO* is not a variable according to the definition given, since it is not a trace.

¹¹A natural question to ask is whether the trace of an expletive can be interpreted as a variable. An expletive is Case marked (but not θ marked). If an expletive were to move to an \bar{A} position, its trace should qualify as a variable. (The specifier of I position qualifies as an A position; see section 2.2.7). Can expletives undergo \bar{A} movement? They can surely undergo A movement, as in (i).

- (i) There_i seemed t_i to appear three men.

Let us suppose that that there were such a thing as an “expletive operator”. An expletive operator would be an operator constructed from an expletive. For example, a *wh* phrase derived from the expletive *there* would presumably have the form *where*. Thus, the question is whether something like (ii) is possible or not.

- (ii) * Where_i did t_i arrive three men?

As far as I am aware, natural language does not entertain structures such as this.

A possible explanation for this is that expletive operators are ruled out on semantic grounds. It seems to be a property of natural language that operators must have some semantic content. Expletives, by definition, are devoid of semantic content. It is not unexpected, following this reasoning, that expletive operators are not permitted by natural language.

More that this needs to be said, however. It is to some degree surprising that natural language tolerates expletive elements at all. Indeed, in recent theoretical work in the Principles and Parameters tradition, semantically vacuous elements such as expletives are not permitted to appear in a Logical Form (LF) representation. Any expletive which appears in a structure must be made legitimate somehow. Many different analyses have been proposed (Chomsky 1995, Lasnik 1995, den Dikken 1995, Groat 1995). Under an expletive replacement by associate raising analysis the expletive is made legitimate through the adjunction

- (144) a. Brian is the man to watch Karlheinz.
 b. Brian is [_{DP} the man [_{IP} *PRO* to watch Karlheinz]]

Rizzi's operator typology

Rizzi (1990) develops a typology of clausal complementizers to account for the licensing of an appropriate operator in relative clauses. Rizzi proposes that complementizers be (partially) specified in terms of the features [$\pm wh$] and [$\pm predicative$].

A $+wh$ C^0 must co-occur with a $+wh$ operator in its spec at S-structure and LF; a $-wh$ C^0 cannot co-occur with a $+wh$ specifier (in other words, spec-head agreement with respect to this feature is compulsory in the domain of Comp). A $+pred$ C^0 must head a CP which is predicated of a "subject of predication"; a $-pred$ C^0 heads a clause which cannot be predicated. The $+wh$ specification is compulsory in questions; the $+pred$ specification is the distinctive property of relatives; a double negative specification characterizes declaratives. This feature system gives rise to four cases, which are illustrated in (108).

- (108) a. $+wh$ $-pred$: (I wonder) what 0 [you saw *t*]
 b. $+wh$ $+pred$: The thing which 0 [you saw *t*]
 c. $-wh$ $+pred$: The thing Op that [you saw *t*]
 d. $-wh$ $-pred$: (I know) that [you saw it]

of the associate to the expletive at LF. Raising of the associate to the expletive is motivated by the need of the associate to check Agreement features.

If an expletive operator were to exist, it would need to raise (overtly) to the specifier of C position. In this position the associate would be unable to adjoin to the expletive: the associate needs only raise to check Agreement features. Agreement features are not checked in the specifier of C position. Hence, any structure in which an expletive operator appears is ruled out because the expletive operator has not been rendered legitimate through adjunction of its associate.

In Modern English, *that* is the spellout of $-wh$ in tensed environments; thus, cases traditionally treated as manifesting the Doubly Filled Comp effect can simply be ruled out as involving an inconsistent feature specification of Spec and head:

- (109) a *What that happened?
 b *The thing which that happened

(Rizzi 1990, pp. 67–68)

Rizzi assumes that a phrase is marked $+wh$ if and only if it is overt. Rizzi's two feature typology yields the following possibilities.^{12, 13}

¹²McDaniel, Chiu and Maxfield (1995) adopts Rizzi's typology to account for partial *wh* movement constructions. McDaniel, Chiu and Maxfield alter Rizzi's typology, "so that relative clause Cs are always [-wh]" (McDaniel et al. 1995, p. 734). I do not make this alteration to the typology.

¹³Rizzi assumes that null operators are "not [...]" specified with respect to the feature $\pm wh$ (see Dobrovie-Sorin 1988 for an interesting argument that null operators are not specified $+wh$)" (Rizzi 1990, p. 68)

Looking at Romanian, Dobrovie-Sorin (1990) suggests that there are two ways in which a noun phrase can come to be interpreted as an operator. The first is lexical: an NP is interpreted as an operator if it carries a "quantifier" feature. Dobrovie-Sorin refers to this as the *qu* feature. The second is structural: an NP is interpreted as an operator if it appears in an "operator position". An overt *wh* phrase is lexically specified as an operator. In contrast, a null operator is obliged to make use of the structural mechanism for interpretation as an operator.

Rather than introduce another feature, I will assume (as does Rizzi) that the difference between overt and null operators can be attributed to a difference in specification of the *wh* feature. I assume that a DP bears a $+wh$ specification only if the DP is overt.

(145)

<i>wh</i>	predicative	Overt	Covert
+	+	relative pronoun	
+	–	interrogative pronoun	
–	+		empty operator

Reconsider now the analysis of infinitival relative clauses. If Rizzi is correct, a clause is interpreted as a relative clause only if it has a [+predicative] complementizer. This would appear to be incompatible with Browning's analysis of subject infinitival relatives, repeated here as (146). Subject infinitival relatives are analysed as bare IPs, and not as CPs.

- (146) a. Brian is the man to watch Karlheinz.
 b. Brian is [DP the man [IP *PRO* to watch Karlheinz]]

Browning's analysis of *object* infinitival relatives poses no problem, as in this case there is an empty operator moving into the specifier of C as in (147b), in parallel with tensed relative clauses, as in (148b).

- (147) a. Brian is the man to watch.
 b. Brian is [DP the man [CP *O_i* [IP to watch *t_i*]]]
- (148) a. Brian is the man who we should watch.
 b. Brian is [DP the man [CP who [IP we should watch *t_i*]]]

Browning argues that the empty operator in (147) should be classified as an empty nominal of the subcategory [+pronominal, –anaphoric]. This amounts to saying that there are two significant subcategories of *pro* that should be recognized: *personal pro* and *relative pro*. This parallels the classification of overt pronouns. This difference

can be captured in Rizzi's feature system by assuming that relative *pro* bears the feature [+predicative], while personal *pro* bears the specification [−predicative]. I propose that the empty category *PRO* can be subcategorized in the same way. Thus, in (146) it is not *personal PRO* that appears, but *relative PRO*. Relative *PRO* is marked with a [+predicative] licensing requirement. Relative *PRO* must therefore appear in the specifier of a [+predicative] C, as in (149).¹⁴

(149) Brian is [_{DP} the man [_{CP} *PRO*_{*i*} [_{\bar{I}} to *t_i* watch Karlheinz]]

Note that while Browning's analysis of infinitival relatives can be reconciled with the spirit of Rizzi's proposal, as in (150), Rizzi's typology cannot distinguish between relative *pro* and relative *PRO*. Both must bear the features [−*wh*,+predicative]. I return to this problem below.

(150)

<i>wh</i>	predicative	Overt	Covert
+	+	relative pronoun	
+	−	interrogative pronoun	
−	+		relative <i>pro</i> relative <i>PRO</i>

Types of operator-variable chains

It has long been known that question formation behaves differently in different languages. In English a morphologically distinct question word is moved overtly to the specifier of C. In a language like Chinese there is no overt movement. There is no

¹⁴Relative *PRO* is identified via predication, and cannot be identified through either control or arbitrary interpretation.

difference in word order in (151a), a declarative clause, and (151b), an interrogative clause.¹⁵

- (151) a. ni xihuan Mary.
 you like Mary
 ‘You like Mary’
 b. ni xihuan shei?
 you like who
 ‘Who do you like?’

Nonetheless, Chinese has verbs which display the same selectional properties as do the English verbs *ask*, *believe* and *know*.

- (152) a. * Wolfgang asked me Ardelia bought books.
 b. Wolfgang asked me who bought books.
 c. * Who did Wolfgang ask me bought books.
- (153) a. Wolfgang believes Ardelia bought books.
 b. * Wolfgang believes who bought books.
 c. Who does Wolfgang believe bought books.
- (154) a. Wolfgang knows Ardelia bought books.
 b. Wolfgang knows who bought books.
 c. Who does Wolfgang know bought books.

Verbs belonging to the same class as *ask* select for an interrogative complement, while those of the *believe* class select for a non-interrogative complement. Verbs of the *know* class can select for either.

¹⁵Unless otherwise stated, the Chinese examples are taken from (Huang 1982). Example (151a) is of my own creation.

A verb which selects an interrogative complement requires a $[+wh, -\text{predicative}]$ complement clause. Such a clause must be a projection of a $[+wh, -\text{predicative}]$ C. A $[+wh, -\text{predicative}]$ C requires a $[+wh, -\text{predicative}]$ operator in its specifier. This explains why (152a,c) are ill formed and (152b) and (154b) are well formed.

A verb which selects a non-interrogative complement requires a $[-wh, -\text{predicative}]$ complement clause. This explains why (153b) is ill formed and (153a,c) and (154a,c) are well formed.

Even though there is no overt movement of *wh* phrases into the specifier of C position in Chinese, the same selectional constraints are in effect. This is demonstrated by the data in (155)–(157). The question raised by this data is how the selectional restrictions can be enforced when there is no movement of the *wh* phrase into the specifier of C position.

- (155) Zhangsan wen wo shei mai-le shu
 Zhangsan ask me who bought books
 a. 'Zhangsan asked me who bought books'
 b. * 'Who did Zhangsan ask me bought books'
- (156) Zhangsan xiangxin shei mai-le shu
 Zhangsan believe who bought books
 a. * 'Zhangsan believes who bought books'
 b. 'Who does Zhangsan believe bought books'
- (157) Zhangsan zhidao shei mai-le shu
 Zhangsan know who bought books
 a. 'Zhangsan knows who bought books'
 b. 'Who does Zhangsan know bought books'

Huang proposed that question words in languages like Chinese undergo covert movement at an abstract level of syntactic representation called Logical Form (LF).

Further investigation has shown that Chinese question words are not on-par in their behaviour with English question words.

Tsai (1994) presents a very interesting account of the differences in behaviour between Chinese, Japanese and English question formation.¹⁶ Tsai's proposal builds on an analysis of the sublexical structure of *wh* words.

...let's compare the following paradigms:

(12)	a.	<u>wh-words</u>	b.	<u>pronominals</u>
		wh-o wh-en		th-ey th-en
		wh-om wh-ere		th-em th-ere
		what		th-at

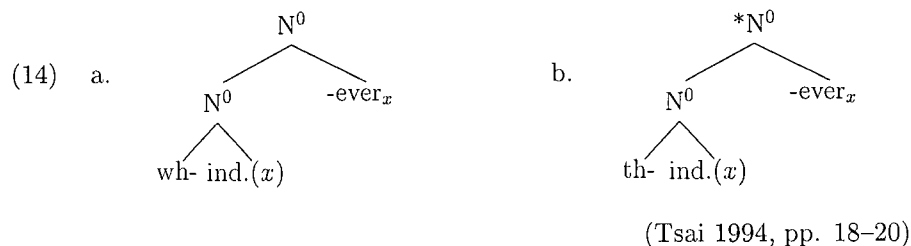
By comparing (12a) with (12b), it is not difficult to see that English *wh*-words and pronominals are more or less built on the same materials except that the prefix for pronominals is *th-* instead of *wh-*. Nevertheless, there is a crucial distinction between these two morphemes: *Th-*, for obvious reason, should be regarded as a reduced form of English definite article *the*, capable of licensing the indefinite morphemes it attaches to (i.e., *ey*, *en*, *em*, *ere*, and *at*). *Wh-*, on the other hand, does not seem to act as a determinant of quantificational force,

¹⁶It is beyond the scope of this thesis to consider the Japanese data in any detail. I do not believe there are any insurmountable difficulties in extending the parsing model proposed here to the Japanese type of question formation.

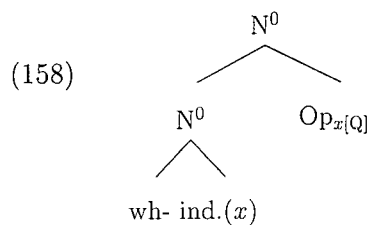
as evidenced by the free relative construals of *wh*-words:

(13)	a.	free relative <i>wh</i> 's	b.	pronominals
		wh-o-ever wh-en-ever		*th-ey-ever *th-en-ever
		wh-om-ever wh-ere-ever		*th-em-ever *th-er(e)-ever
		what-ever		*th-at-ever

(13a) shows that *wh*- does not block binding from the suffix *-ever*, which contributes universal force to the indefinites, as illustrated by (14a). As a result, *whoever* can be paraphrased directly as ‘anyone’, *whatever* as ‘anything’, *when-ever* as ‘anytime’, and so forth. In contrast, pronominals cannot be suffixed by *-ever*, as shown by (13b). This indicates that *th*- blocks the binding construal between *-ever* and the indefinite in (14b), just as its determiner counterpart might do in a full DP:



According to Tsai’s analysis, a *wh* word in English contains an interrogative operator, and has the structure shown in (158).



I noted above that Chinese question formation does not involve overt movement. It displays neither *wh*-island nor complex NP island effects. Question formation in English is sensitive to both types of islands. Tsai takes the difference to stem from a difference in how the operator-variable chain is formed in the two languages. There is independent evidence that there are two ways to form chains. Islands affect movement. Hence, chain formation via movement is sensitive to both *wh*-islands and complex NP islands. Chains formed by unselective binding are not subject to these island constraints since the island-sensitive process of movement is not involved. Under Tsai's analysis question formation in English involves both types of chain formation, while in Chinese only unselective binding is involved. Japanese makes use of both chain formation and unselective binding. This accounts for the fact that Japanese question formation is sensitive to some islands, and yet is not sensitive to all islands. In all three languages the interrogative operator $\text{Op}_x[\text{Q}]$ unselectively binds the variable x which is introduced by the indefinite morpheme "ind.(x)".

In English $\text{Op}_x[\text{Q}]$ forms an integral part of the *wh* word and cannot be moved independently of it. Hence the whole *wh* word must move in order to satisfy the *wh* criterion (Tsai 1994, p. 18):

(159) The *wh* criterion

- a. A *wh*-operator must be in a Spec-head configuration with an $\text{X}^0_{[+wh]}$.
- b. An $\text{X}^0_{[+wh]}$ must be in a Spec-head configuration with a *wh*-operator.

In Chinese $\text{Op}_x[\text{Q}]$ exists independently of the indefinite, and is generated directly in the specifier of C in order to satisfy the *wh* criterion. Only an unselective binding relationship is established.¹⁷

¹⁷Cheng (1991, p. 122–3) notes that question words in Chinese are not the same as *wh* words in English.

...wh-words in Mandarin Chinese can be interpreted in three different ways. They can be interrogative words, existential quantifiers and universal quantifiers. [...] The interpretation of a wh-word varies depending on another element in the sentence. The elements which can determine the reading of a wh-word are: a wh-particle (*ne* or its null counterpart), a yes-no particle (or A-not-A question), a negative marker and *dou* 'the universal marker'

Cheng proposes an analysis of these so-called *wh* words as *wh* indefinites. These indefinite noun phrases must fall in the scope of some form of operator. They are interpreted according to the properties of the operator in whose scope they appear. For example, in the scope of a negative operator they behave like negative polarity items, like anybody.

- (i) * I think anybody stole the book.
- (ii) I do not think anybody stole the book.

The following Chinese examples are from Cheng (1991).

- (iii) *sheme* has only interrogative reading

hufei chi-le sheme (ne)
Hufei eat-ASP what Q_{WH}
'What did Hufei eat?'

- (iv) existential (polarity) reading of *sheme*

qiaofong mai-le sheme ma
Qiaofong buy-ASP what Q_{YN}
'Did Qiaofong buy anything?'

- (v) universal reading of *sheme*

botong sheme dou chi
Botong what all eat
'As for Botong, he eats everything.'

With respect to Japanese, Tsai (1994, pg. 49) observes that,

Although it is still under debate whether Japanese displays genuine *wh*-island effects [...] it is generally agreed that Japanese lacks Complex NP (i.e., strong island) effects.

Under Tsai's analysis Japanese is like Chinese in that $\text{Op}_x[\text{Q}]$ exists independently of the indefinite. Tsai proposes that $\text{Op}_x[\text{Q}]$ is generated in the specifier of D (or P). Thus, not only is an unselective binding relationship involved in question formation, but also a movement relationship.

Consequently, [Japanese] patterns with English *wh*'s [...] in displaying *wh*-island effects [...] On the other hand, since the Q-operator is already in the Spec of DP, any *wh-in-situ* within its c-command domain (and hence within a Complex NP) can be licensed by unselective binding ...

(Tsai 1994, pg. 50)

Summarizing the predictions of the analysis, Tsai states,

The distribution of locality effects in the three types of language may thus be summarized as follows:

(76)

	English	Japanese	Chinese
<i>Wh</i> -island effects	yes	yes	no
Complex NP effects	yes	no	no

This is exactly what we would expect from the distinct positions of Q-operators

in these languages, as illustrated below (word orders irrelevant):

- (77) a. Chinese-type: [CP Op_x[Q] [IP...wh(x)...]]
 b. Japanese-type: [CP Op_x[Q] [IP...[PP/DP t_x [...wh(x)...]]...]]
 c. English-type: [CP [PP/DP wh(x)-Op_x[Q]]_k [IP...t_k...]]

Since Chinese Q-operators are inserted in the CP Spec, no movement is involved. In contrast, since Japanese Q-operators are inserted in the DP Spec, the “half-way” movement to the CP Spec evades Complex NP effects, but still respects the *wh*-island constraint, or whatever principle it might be reduced to. As for English, since the whole *wh*-phrase must move to fulfill feature-checking on the CP Spec, both Complex NP and *wh*-island constraints are to be observed.

(Tsai 1994, pp. 52–53)

Under Tsai’s analysis there is a covert operator possessing the appropriate features in the specifier of C in each of the Chinese examples (155–157). This covert operator has not been moved, but appears in its “base-generated” position.

There are two things I wish to take from Tsai’s proposal. First is the proposal that there are two different mechanisms involved in forming the operator-variable chain: movement (in English) and unselective binding (in Chinese). Second is the idea that the sublexical structure of *wh* words varies across languages, and that the “operator part” of *wh* words can exist independently in a specifier of C position.

5.1.4 A typology of nominal chains

In this section I tackle the question of how Tsai’s analysis can be made compatible with Rizzi’s typology. I noted above a difficulty with Rizzi’s typology, that it does

not make a distinction between a relative *PRO* and a relative *pro*. Tsai's analysis makes finer distinctions between different types of interrogative operators than can be accommodated in Rizzi's feature system. It is, of course, possible to add more features to Rizzi's typology in order to make the requisite distinctions. While this solves the problem at hand, it is not desirable to simply invent new features. Fortunately the Case and θ marking features discussed above seem to make the correct distinctions. Four binary-valued features yield sixteen possible subcategories of nominals. In this section I explore this feature space.

It is important to notice before proceeding further that I am broadening the perspective slightly. In the previous discussion I considered how the parser could identify an element on the basis of the syntactic context in which it appeared. In other words, the focus was on deducing what element could appear in a position from the licensing properties of that position. Case and θ marking occurs in A positions, but $[\pm Wh]$ and $[\pm Pr]$ marking occurs in \bar{A} positions. A single element cannot occupy two or more positions, but a *chain* can. The focus in this section is therefore on the licensing and identification of chains.

With this introduction I can dispense with one of the sixteen possibilities immediately. A completely negative specification for all the features indicates the complete absence of licensing features. In other words, this represents nothing. I now turn to a discussion of the remaining fifteen cases. These fall naturally into five groups of three, each of which I consider in turn.

The Case- θ typology

Recall the Case- θ typology given in (140). These possibilities are naturally mapped into the four-feature typology as shown in (160). The blank entry in this table is

ruled out by the grammar: overt nominals must bear Case.

(160)

<i>wh</i>	predicative	Case	\emptyset	Overt	Covert
–	–	+	+	argument NP	argument <i>pro</i>
–	–	+	–	expletive NP	expletive <i>pro</i>
–	–	–	+		argument <i>PRO</i>

The *wh*-predicative typology

Recall the *wh*-predicative typology given in (145). These possibilities are naturally mapped into the four-feature typology as shown in (161). The empty entries in this table are ruled out under the assumption that operator phrases are marked *+wh* only if they are overt.

(161)

<i>wh</i>	predicative	Case	\emptyset	Overt	Covert
+	+	+	+	relative pronoun	
+	–	+	+	interrogative pronoun	
–	+	+	+		relative <i>pro</i>

Impossible expletives

The cases shown in (162) correspond to different forms of expletive operators. I assume that such operators are ruled out, as discussed in section 5.1.3.

(162)

wh	predicative	Case	θ	Overt	Covert
+	+	+	-		
+	-	+	-		
-	+	+	-		

Relative *PRO*

Operators which are specified $-Case$ and $+\theta$, shown in (163), cannot be overt (because of the Case filter). Accepting Rizzi's assumption, covert operators must be specified $-wh$. The remaining entry is the relative operator in subject infinitival relatives: relative *PRO*.

(163)

wh	predicative	Case	θ	Overt	Covert
+	+	-	+		
+	-	-	+		
-	+	-	+		relative <i>PRO</i>

Tsai's operators

There are three elements of the feature space still to account for. These possibilities are shown in (164).

(164)

<i>wh</i>	predicative	Case	θ	Overt	Covert
+	+	-	-	?	?
+	-	-	-	?	?
-	+	-	-	?	?

Recall that Tsai reduces the difference in question formation in English and Chinese to a difference in whether an interrogative operator is incorporated as part of the *wh* word, as in English (165a) or is a “free agent”, as in Chinese (165b).

- (165) a. [CP [PP/DP *wh*(x)-Op_x[Q]]_k [IP... t_k...]]
 b. [CP Op_x[Q] [IP... *wh*(x)...]]

A free agent operator is not moved from a Case or θ marked position. Instead, it unselectively binds a variable in such a position. Hence the free agent operator does not itself require Case or θ marking. We therefore have the possibilities shown in table 166.¹⁸

¹⁸Indeed, under Tsai’s analysis this free agent operator is (at least in English and Japanese) a sublexical element of the *wh* phrase. It is natural to assume that the phonetic matrix resides with the phrasal element and not with the sublexical operator part. Hence, there is no contradiction in having a +*wh* specification with a covert element in table 166.

(166)

<i>wh</i>	predicative	Case	θ	Overt	Covert
+	+	-	-		
+	-	-	-		interrogative operator
-	+	-	-		relative operator

It is worth noting that *wh* particles in languages such as Chinese and Japanese, like *ne* in Chinese and *ka* in Japanese, are generally assumed to be heads, or X^0 categories (see, for example, (Cheng 1991, pg. 35)). These particles can be taken to be on-par with the English complementizer *that* as the overt spell-out of a feature of C. In other words, one can assume that the question particles *ne* and *ka* are an overt spell-out of the $+wh$ feature.

I assume that the independent features *wh* and *pr* cannot coexist outside of a complete phrase. A complete *wh* operator phrase must bear both Case and θ marking. Hence the completely blank entry for the first row of (166).

5.2 Parsing revisited

In this section I describe the various components of my parsing algorithm. In section 5.4 I present the complete algorithm.

5.2.1 Underspecification and augmentation

Ambiguity is pervasive in natural language. There are several ways of dealing with ambiguity. One obvious way is to maintain one unambiguous structure for each possible reading of an ambiguous utterance. This approach is problematic because it

may require a very large number of structures to be maintained. Furthermore, multiple structures must be maintained until disambiguation occurs, and disambiguation may not occur within the scope of a single utterance.

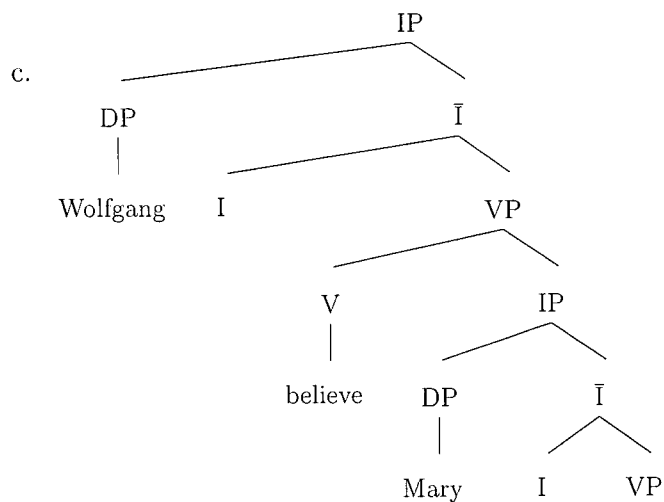
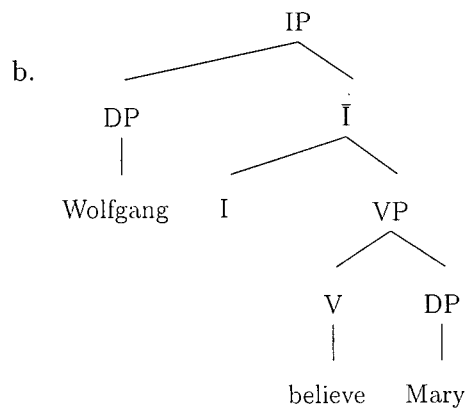
Another possible approach to dealing with ambiguous utterances is to maintain a single underspecified representation. It is not obvious that a single underspecified representation can admit all possible readings of a given ambiguous utterance. While I do not delve into this issue here, as it is beyond the scope of this dissertation, this may in fact be advantageous. A strong argument against the multiple representation approach is the existence of garden path utterances (see section 2.3.1 for discussion). It has been argued that it is precisely these types of ambiguities which cannot be represented using a single underspecified representation (see, for example, discussion by Pritchett (1992) and Gorrell (1995)).

Marcus et al. (1983), Pritchett (1992)) and Gorrell (1995) make use primarily of structural underspecification. This is explained below. My algorithm extends the use of underspecification to the specification of licensing requirements.

The proposal that the parser can leave licensing specifications underspecified extends the idea of “minimal commitment” in structure building to licensing specifications. Marcus et al. (1983) propose that a deterministic parser build underspecified tree representations in which the primary structural relationship is not *daughter* but *descendant*. This underspecification of structural relations allows such a “minimal commitment” parser to incorporate incoming material into the evolving parse tree, without having to specify immediately its exact structural relationship to the rest of the parse tree.

As an example, consider how a minimal commitment parser processes (167a).

(167) a. Wolfgang believes Mary to be lying.



In the final analysis, *Mary* is interpreted as the subject of the embedded clause. Initially, however, *Mary* is interpreted as the object of the matrix verb *believe*, as in (167b). A minimal commitment parser can reinterpret *Mary* as the subject of the embedded clause without recourse to backtracking, because the initial structural commitment, namely that *Mary* is a descendant of the VP headed by *believe*, still holds, as in (167c).

Consider now how underspecification of licensing requirements might work to the parser's advantage. The θ criterion requires that argument DPs be assigned θ roles. It is not the case that every DP is interpreted as an argument. A left

dislocated DP, as in (168), is not an argument. In contrast a topicalized DP, as in (169), is an argument.

(168) Ardelia, I think that Wolfgang really admires her.

(169) Ardelia, I think that Wolfgang really admires.

The parser cannot distinguish between these two cases on the basis of their attachment site (they are both adjoined to IP). Neither example gives rise to a garden path effect. This implies that the parser does not arbitrarily choose to specify DPs as requiring or not requiring θ marking. If it did, then it would be wrong either in the case of left dislocation or topicalization. In chapter 2 I assumed that DPs were projected with both a Case and a θ licensing requirement. On the basis of minimal pairs like left dislocation and topicalization, this assumption cannot be maintained in the general case (it is, in fact, abandoned in section 5.2.2).¹⁹

In order for a phrase structure representation to be well-formed, all constituents must have their licensing requirements fully specified at the end of a parse. However, there is no requirement that this condition hold throughout the parse. A simple solution is thus to assume that constituents can be underspecified with respect to their licensing requirements during a parse. This simple proposal leads to interesting predictions. These predictions are explored in chapter 7.

The licensing mechanisms at the parser's disposal are given in (170).

- (170) a. θ marking
b. Case marking
c. *wh* feature

¹⁹As discussed in section 4.4.2 filler-driven approaches such as the Active Filler Strategy and the Minimal Chain Principle, cannot adequately deal with optional fillers.

- d. predicative feature
- e. functional selection
- f. adjunction (e.g. modification, predication, topicalization)

(170a–d) are bidirectional licensing mechanisms. By this I mean that (i) the licensor has a need to discharge this feature, and (ii) the constituent being licensed has a need to be assigned this feature. (170e–f) are unidirectional licensing mechanisms. Functional selection is a requirement only of the selecting head, not of the selected constituent. A clause can be licensed via predication, but there is no need on the part of the licensing DP to be a subject of predication.²⁰ Similarly, an IP can license a DP as an adjunct, but it need not.

(170a–d) are also the licensing mechanisms which are relevant for identifying subcategories of nominals. Nominals must be fully specified with respect to these four licensing features at the end of a parse. They need not be fully specified with respect to them during the parse.

The licensing features shown in (170a–d) are specified on the constituent being licensed as features which must be assigned to the constituent. These form the constituent's licensing specification. A constituent's licensing specification can be augmented with non-contradictory licensing specifications. Figure 5.1 summarizes the possibilities. In this figure [+op] is an abbreviation for any of the following specifications: [+Wh,+Pr], [+Wh,−Pr] or [−Wh,+Pr]. A nominal with any of these licensing requirements qualifies as an operator. Augmentation can take place up any of the lines in figure 5.1. I discuss *when* the parser attempts to augment a constituent's licensing specification when I revisit attachment and extension in

²⁰From the parser's perspective predication is adjunction to a DP by a CP bearing a [+pr] feature.

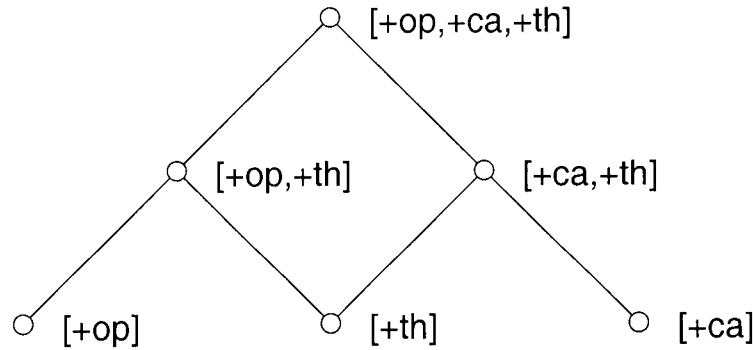


Figure 5.1: The parser's view of chain-heading empty categories

sections 5.2.2 and 5.2.3 respectively.

(171)	Node in figure 5.1	<i>wh</i>	predicative	Case	θ	Overt	Covert
		[+op]	+	-	-	-	interrogative operator
		[+op]	-	+	-	-	relative operator
		[+ θ]	-	-	-	+	argument <i>PRO</i>
		[+ca]	-	-	+	-	expletive NP expletive <i>pro</i>
		[+op,+ θ]	-	+	-	+	relative <i>PRO</i>
		[+ca+ θ]	-	-	+	+	argument NP argument <i>pro</i>
		[+op,+ca,+ θ]	+	+	+	+	relative pronoun
		[+op,+ca,+ θ]	+	-	+	+	interrogative pronoun
		[+op,+ca,+ θ]	-	+	+	+	relative <i>pro</i>

In general, underspecification allows the parser to delay certain decisions. Underspecification is also a mechanism which permits ambiguity resolution to proceed monotonically. For example, Alshawhi and Crouch (1992) “describes how mono-

tonic reference resolution and scoping can be carried out using a revised Quasi Logical Form (QLF) representation. Semantics for QLF are presented in which the denotations of formulas are extended monotonically as QLF expressions are resolved.” (Alshawhi and Crouch 1992, pg. 32). In particular, underspecification allows the parser to divorce the initial licensing of a constituent from its complete identification.

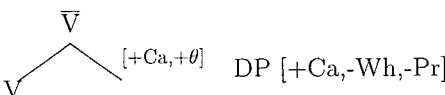
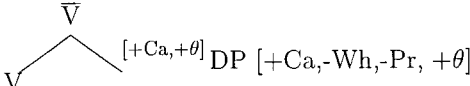
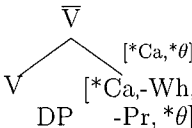
5.2.2 Attachment

The parsing algorithm operates by building a maximally licensed structure at each stage of processing. One stage is called *attachment*. Attachment refers to the combination of existing and new structure into a single, larger structure. In chapter 2 I discussed how attachment works to satisfy licensing requirements. That discussion took place under the assumption that DPs are initially projected with a $[+\theta]$ licensing need. This assumption, which is both unnecessary and problematic for the reasons outlined in section 5.2.1, is now abandoned. Instead, a DP is projected with an underspecified licensing requirement specification.

Consider for example the case of an overt DP which is to be attached into the object position of a typical transitive verb. Such a verb assigns both a θ role and Case marking to the object position. A DP is projected without a specification for θ marking, however, so as it stands the DP cannot discharge all of the licensing features of the position, which is a requirement for attachment to succeed (since only one chain may pass through a position). Example (172a) shows this case, where DP is not specified for a θ role requirement, and V assigns both a θ role and Case to its complement.

If a phrase is to discharge a predicate’s θ role the phrase must be specified

as needing a θ role. A DP's licensing features can be *augmented* to include a $[+\theta]$ specification. This is shown in example (172b). The next step, attaching DP as a complement of V, proceeds as before. The result is shown in (172c).

- (172) a.  DP [+Ca, -Wh, -Pr] DP underspecified wrt θ
- b.  DP [+Ca, -Wh, -Pr, + θ] DP augmented with θ requirement
- c.  DP [*Ca, -Wh, -Pr, * θ] DP attached into \bar{V} structure.

Functional selection is the obligatory selection by a functional head of a complement of a certain category. This is a uni-directional requirement. The functional head requires a complement with certain features, but the selected head has no concomitant requirement that it be selected by a given functional head. Since there is no licensing requirement specified on the selected constituent, augmentation is never involved in functional selection.

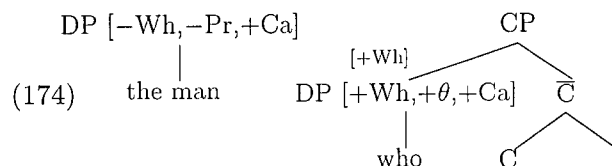
Predication refers (in the context of this dissertation) to the licensing of a clause as a relative clause modifying a DP. One context which triggers the licensing of a clause via predication is (173a), where 'WH' refers to a *wh* phrase. An example of this configuration is shown in (173b).

- (173) a. DP [_{CP} WH ...]
- b. the man who Mary likes

At the point when *who* is being processed, the parser must determine how it is licensed in the current structure. *Who* is specified in the lexicon with the licensing features $[+Wh, +Ca, +\theta]$.²¹ Note that it is underspecified with respect to the feature

²¹A question which naturally arises is how adjunct *wh*-phrases, such as *why*, *how*, *when*

[\pm Pr]. Since *who* is specified as [$+$ Wh], it must appear in the specifier of a [$+$ Wh] C. The parser constructs an appropriate projection of C and attaches the *wh* phrase in its specifier, as in (174).



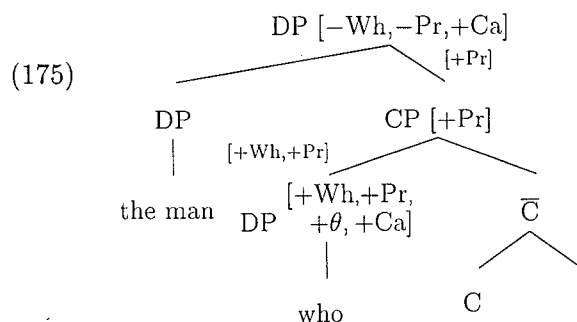
The DP *the man* cannot be licensed with respect to the CP. The CP can be licensed via predication with respect to *the man*, however. A clause licensed via predication is endowed with the feature [$+$ Pr]. This feature is shared with the specifier of C and *where* are processed. Under most analyses, these question words are adverbial in nature, and originate in adjunct position. Since there is no obligatory gap in the case of an adjunct gap, the parser will simply choose the first possible optional gap site.

Cinque (1999) proposes that there exists a fixed and universal hierarchy of clausal functional projections which houses all types of adverbs. The head of each functional projection can host an adverbial expression in its specifier.

What are the implications of such a clausal structure for processing? Since the hierarchy is universal and fixed, each language presents the same hierarchical clause structure. Therefore it may be assumed that the parser projects a complete clause structure (one which includes all functional heads from the universal hierarchy).

In the case where a *wh* phrase has been overtly moved it should be possible for the parser to consider each (compatible) specifier in this universal clause structure as possible source of the movement. What remains unclear is how the parser should select amongst multiple plausible extraction sites. It is also unclear whether the parser should be permitted to postulate empty operators in any of these positions and if so, how to constrain the postulation of empty operators so that spurious operators are not introduced into the structure. I have no answers to these questions at this point.

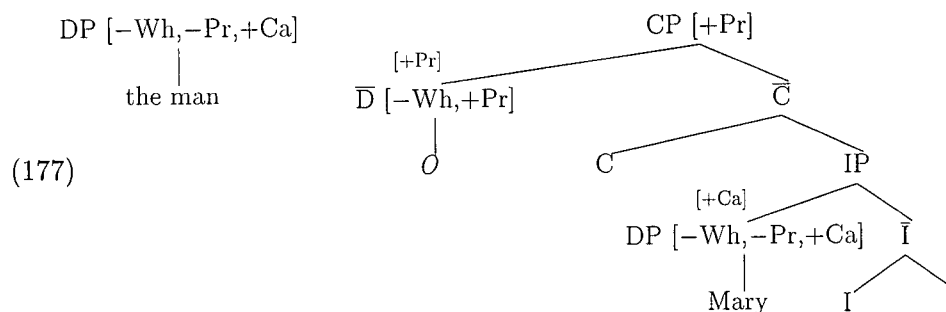
position.²² The *wh* phrase is therefore augmented to include the [+Pr] specification. The result is shown in (175).



Another context which triggers the licensing of a clause via predication is depicted in (176a), with an example in (176b). This case is distinguished from the cases in (168) and (169) by the lack of a heavy pause between DP_1 and DP_2 . A heavy pause, or stress on DP_1 , blocks licensing via predication.

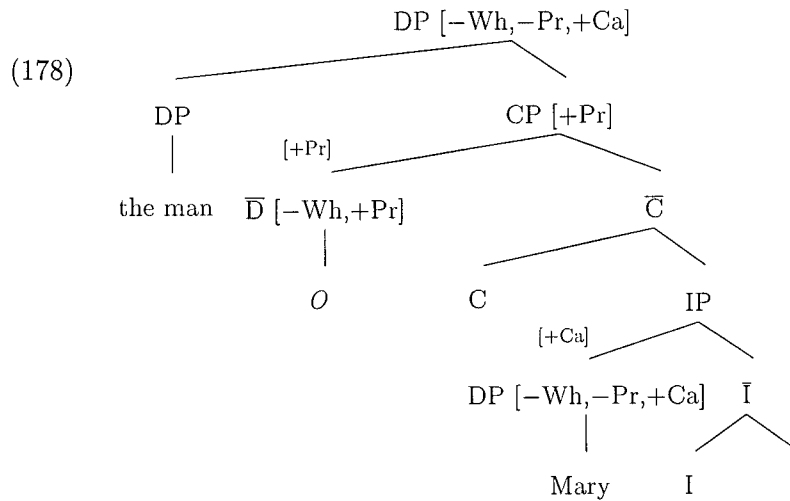
- (176) a. DP_1 [IP DP_2 ...]
 b. The man Mary likes

Processing proceeds somewhat differently here than in (173). In this case there is no projection of C, only an IP available. In order to license this clause via predication a projection of C must first be constructed, as in (177).



²²This can be accomplished by feature percolation from the phrasal level to the head, and specifier-head agreement.

In order for the CP to be properly licensed, it must be licensed via predication:
 tion:



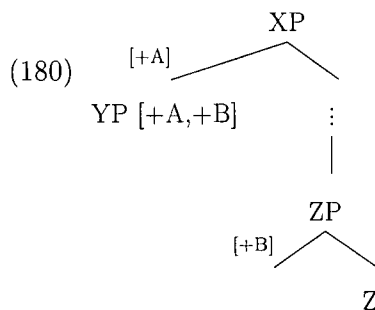
DPs can also be licensed as adjuncts. This option is indicated to the parser by stress or a pause, as in a topicalization or left dislocation structure.

(179) DP_1 [IP DP_2 ...] (with a pause between DP_1 and IP and/or heavy stress on DP_1)

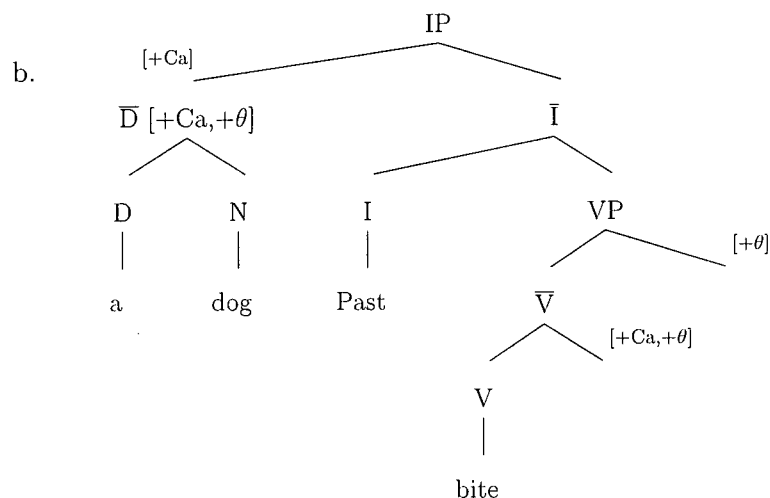
5.2.3 Extension

As with attachment, extension was introduced in chapter 2. Here I review extension, taking into account the possibility of augmenting the licensing specification of the moved constituent.

Recall that I use the term extension to refer to the process by which the parser recovers a movement configuration. A straightforward movement-triggering configuration can be depicted as in (180). An example is shown in (181a–b).



(181) a. A dog bit Billy.



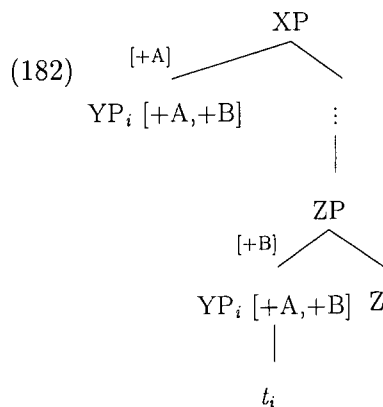
Consider first the general case shown in (180). The empty position to which the licensing feature B is assigned is encountered when Z is projected and ZP is attached into the CPPM.

An example of such a configuration is shown in example (181). In this case the θ role assigned to the VP internal subject position must be discharged, but there is no unattached material which can be attached into this position in order to discharge the θ role. Assuming that the DP *a dog* must receive a θ role to be licensed, its θ requirement is satisfied if it is construed as having moved from the VP-internal subject position.²³

²³Since I just abandoned the assumption that DP's are projected with a θ requirement,

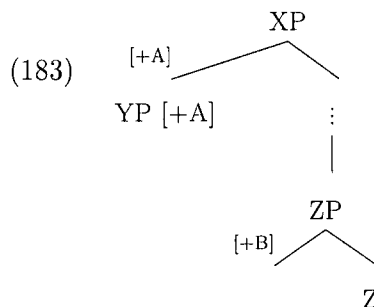
Assume that there is nothing which can attach directly into this position to satisfy its licensing requirements. How can this position's licensing requirements be satisfied? The answer is through movement of YP. Movement of YP to this position assigned the feature $[+B]$ satisfies not only the requirement of the position to discharge the $[+B]$ feature, but also the requirement that YP be assigned the $[+B]$ feature.

The phrase YP has one unsatisfied licensing need, namely $[+B]$. Movement allows YP to behave as though it occupies more than one position. By postulating a trace of YP in the empty position, and creating a chain by coindexing YP and the trace, Z can discharge its licensing requirement to YP. As a side effect, YP has its remaining licensing requirement satisfied.



Consider now the case in which a licensing feature must be discharged, but there is no unattached phrase to fill the position, and there is no constituent in the tree which requires this licensing feature. This movement-triggering configuration can be depicted as follows:

it seems odd that I now seem to assume that this is the case. In fact, I do not assume that the DP is projected with a θ requirement. Instead, I assume that the DP acquires this θ requirement through augmentation. This is discussed below.



This configuration is unlike the previous case in that YP does not have any unsatisfied licensing requirements. Suppose that the licensing specification $[+B]$ is not incompatible with YP. Then if YP were to acquire the licensing feature B, this case could be reduced to the case above. I will assume this to be the case.

The first step is to *augment* the licensing condition structure of YP. The structure we are left with is precisely that shown in (180). Processing proceeds as indicated above, with construction of a movement chain to satisfy the $[+B]$ licensing requirement. The final configuration is as shown in (182).

How is compatibility of licensing features indicated? By lack of a licensing feature specification (i.e. by underspecification). To specify that YP is incompatible with the licensing feature B, YP is given a licensing feature structure explicitly indicating the incompatibility:

(184) YP $[+A, -B]$

5.2.4 Gap-driven processing

In section 5.1 I developed a typology of nominals based on four binary-valued features. In order to be well-formed at the end of a parse a nominal must be fully specified for all four features. During processing, however, it may be unclear what the full feature specification of a constituent should be. This uncertainty comes

about because of the possibility of movement.

The feature content of a nominal, whether overt or covert, must be recoverable from the syntactic context in which it appears. Movement allows that context to be spread over several positions in a tree. In order to divorce licensing from identification, I propose that the feature content of constituents is underspecified. A constituent can be licensed on the basis of any licensing feature. The full content of the constituent cannot be determined until all the licensing mechanisms have been determined.

In chapter 4 I presented arguments against the human sentence processor using a filler-driven approach to filler-gap dependencies. I showed that neither Fodor's (1978) *gap-as-second resort* model nor Frazier's (1987a) *Active Filler Strategy* (AFS) correctly accounts for the known filled-gap effects in English, as described by both (Crain and Fodor 1985) and (Stowe 1986).²⁴

There is more to the story than this, however. Evidence from languages other than English must be considered. Frazier (1987a), Frazier (1987a) and Frazier and Flores D'Arcais (1989) present evidence from Dutch which they claim argues strongly against a gap-driven approach, and in favor of the AFS.²⁵

The syntax of Dutch is similar to that of English, but Dutch exhibits a different word order than English. Dutch has an underlying SOV word order, with

²⁴(Lee 1999) presents preliminary evidence which brings into question whether English truly lacks a filled-gap effect in subject position. However, the author noted [p.c.] that further experiments are planned to clarify the results. Because the results are at this point preliminary, I will not draw any definite conclusions based on them. The results do indicate that there may be reason to question the lack of a filled-gap effect in subject position in English which, if true, would throw into question the accuracy of the model I propose.

²⁵Schriefers, Friederici and Kühn (1995) present similar evidence for German.

a *verb second* (V2) effect in main clauses. In a language with V2, the word order in main clauses is different from that in subordinate clauses. In main clauses the verb occupies the second (constituent) position.

- (185) a. Jan_i zag_j t_i iets t_j
 Jan saw something
 Jan saw something.
- b. Jan_i zag_j iets t_i t_j
 Jan saw something
 Something saw Jan.

Fodor's gap-as-second resort strategy predicts that an initial noun phrase should be preferentially interpreted as the object of the verb. This preference comes about because the gap-as-second resort strategy will choose the unmoved noun phrase to fill the first argument position of the verb (the subject position), leaving the object position the only one available for the fronted noun phrase.

The AFS predicts just the opposite, that an initial noun phrase should be preferentially interpreted as the subject of the verb. This preference comes about because the AFS associates the unmoved noun phrase with the first available position, which in this case is the subject position. The unmoved noun phrase is left to be interpreted with respect to the object position.

The experimental results which are reported support the predictions of the AFS. Notice, however, that the results are not inconsistent with all gap-driven approaches. In fact, their evidence only argues against a gap-driven approach which relies on lookahead. A gap-driven approach without lookahead makes the same predictions as the AFS in these cases.

A gap-driven approach does not change its processing in the face of an identified filler. Indeed, under a gap-driven approach the filler is not identified until a

potential trace site has been located. The parser will not attempt to postulate a trace in a position if there is overt material available which satisfies the licensing requirements of the position. Without using lookahead, overt material takes precedence over a displaced constituent only if the overt material precedes the licenser for the position. This happens in the subject position in English, but not the object position. Hence, there is no filled-gap effect in subject position in English simply because the option of a trace is not considered precisely when there is overt material to fill this position. In the object position, however, because the licenser precedes any overt material which may be present, the parser will attempt to postulate a trace before processing the next element from the input.

In English, the proposed algorithm therefore correctly captures the filled-gap effect with objects, as well as the lack of filled-gap effect with subjects.

Data from Dutch is discussed in the remainder of this section. In section 5.2.4 the data concerning extraction preferences in Dutch main clauses is reviewed. In section 5.2.4 the very interesting data regarding extraction preferences in Dutch relative clauses is studied.

The main point to be made about the proposed algorithm is that, in contrast to both the gap-as-second resort strategy the AFS, it does not make *absolute* predictions regarding the filled-gap effect across languages. Instead, the predictions which are made are a function of the word-order, or more precisely, the order of presentation to the parser of the licensors of positions relative to the order of presentation to the parser of the licensees. This is exemplified by the behaviour of the parser when presented with language with properties different from English, such as Dutch.

Dutch main clauses

Although Dutch is a verb final language, it exhibits verb-second (V2) effects in main clauses. It is therefore not unreasonable to assume that the VP-external subject position is identified as soon as the V2 position is parsed. Since the V2 position contains a tensed verb, the parser could at least project structure from an Inflectional head, and build an IP. If the verb encountered is a main rather than an auxiliary verb, the parser could conceivably also build a VP to host the main verb. In either case the VP-external subject position is available. The only licensing requirement that the fronted noun phrase possesses is a Case requirement, which will be satisfied in the VP external position. The fronted noun phrase could be augmented with a θ role requirement, and be made compatible with the object position. Given the choice between augmenting or not, the parser will choose not to augment. Hence, the parsing algorithm predicts a general preference for interpreting ambiguous extractions as subject extractions, all else being equal.

In Dutch, the proposed algorithm in effect predicts a filled-gap effect in V2 clauses where the object has been extracted. This is exactly what was found in Frazier and Flores D'Arcais (1989). They tested several sentences in which the subject-verb agreement disambiguated the extraction site of the fronted noun phrase. They found that sentences which contained a fronted object took longer to process than those which contained a fronted subject:

The evidence for the superiority of subject-initial sentences is particularly impressive when we remember that the more difficult object-initial sentences are actually disambiguated earlier than the subject-initial sentences, as illustrated in (10).

(10) a. NP-sg. V-sg ... (De chauffeur betaalt...)

b. NP-sg. V-pl ... (De chauffeur betalen...)

The string in (10a) is still consistent with either a subject-initial or an object-initial sentence, whereas the string in (10b) is inconsistent with subject-initial analysis. The fact that sentences like (10a) are processed *faster* despite *later* disambiguation is thus interesting. It argues against a multiple analysis approach to syntactic processing in general, since in this case we would expect earlier disambiguation to necessarily simplify processing, reducing the number of analyses which must be computed and maintained in memory. This fact is also consistent with the idea that the processor is not delaying analysis of a structurally ambiguous input, since in this case we would expect object-initial sentences to be easier than subject-initial sentences, due to the earlier disambiguation point in the object-initial sentences reducing the memory burden imposed by unstructured material.

(Frazier and Flores D'Arcais 1989, pg. 340)

A gap-driven parser does not identify fillers as such. Instead fillers sit where they are until an empty position is identified. If no overt material can be attached into this position, the parser checks for potential fillers in the structure. If any are found, the parser augments if necessary, and extends if possible, in order to satisfy the licensing requirements of the position.²⁶

²⁶Searching through a potentially unbounded structure raises concerns for any claim that this algorithm is a psycholinguistically plausible model of human sentence processing. Human sentence processing is believed to operate in linear time. This is incompatible with searching through a potentially unbounded amount of structure. There are (at least) two ways to address this concern. One is to make use of intermediate traces in the derivation

From the perspective of my parsing algorithm, the reason that in Dutch matrix clauses object extractions take longer to process than subject extractions is the same as the reason that English shows a filled-gap effect in object but not subject extractions. In English a filled-gap effect shows up in cases where extraction appears to be from an object position, but is in fact from a subordinate clause. In Dutch a similar effect shows up when an object extraction is mistaken for a subject extraction.

Dutch relative clauses

Several studies (Frazier 1987a, Schriefers et al. 1995) have shown quite strong evidence that there is a filled-gap effect in subject position in object relative clauses (those exhibiting relativization from object position within the relative).

The relative clause example in (186), from (Schriefers et al. 1995), serves to frame the relevant discussion.

- (186) a. ik schreef aan de man die mijn vriendin bezocht.
 I wrote to the man who my girlfriend visited
-

of long-range movements. This approach raises the question of how intermediate traces are licensed. Another approach is put forward by Frank (1992). Frank recasts a standard Principles and Parameters theory of syntax in a Tree-Adjoining Grammar (TAG) framework. In this theory no intermediate traces are used, so the question of how they are licensed does not arise. In the syntactic theory that Frank develops all movement is localized to the elementary trees of a TAG. The size of elementary trees is bounded. Because Frank's parsing algorithm is deterministic (i.e. it pursues only one parse at a time), and because no unbounded search is required, it operates in linear time. Although I do not explore this issue further here, it is straightforward to incorporate the unadjoining and unattachment operations of Frank's algorithm into the algorithm presented here.

I wrote to the man who my girlfriend visited or

I wrote to the man who visited my girlfriend

- b. ik schreef aan de man [_{CP} die_i mijn vriendin *t_i* bezocht]

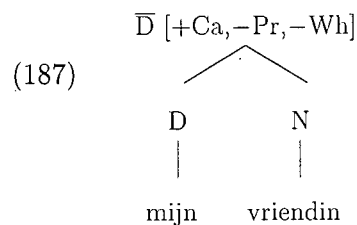
I wrote to the man who my girlfriend visited

- c. ik schreef aan de man [_{CP} die_i *t_i* mijn vriendin bezocht]

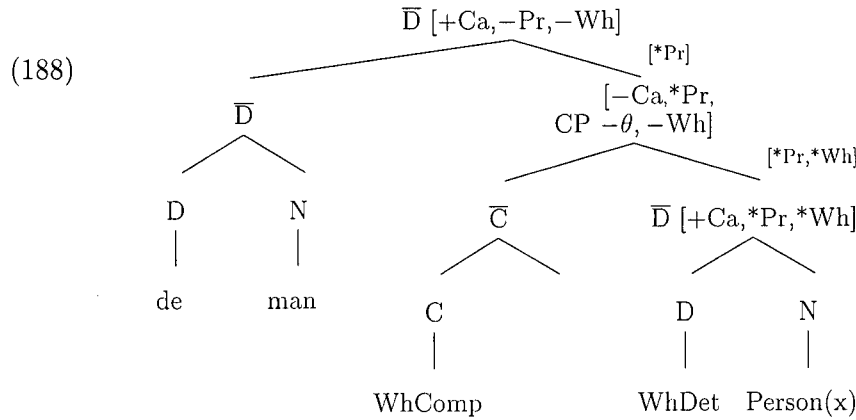
I wrote to the man who visited my girlfriend

The issue is the initial (or preferred) interpretation of the role the head of the relative clause in (186a), *the man*, within the relative clause. There are two possible extraction positions (subject and object position), shown in (186b) and (186c).

Focusing on the processing of the relative clause and its head (*de man die mijn vriendin bezocht*), the proposed structure building mechanism constructs the structure shown below up to the point of having seen the verb of the relative clause.



The following structure is in the parser's stack:



Because the next token read is the verb of the relative clause, the parser attaches *mijn vriendin* as the object of the verb - the only reasonable attachment decision, resulting in a subject-gap preference.

The crucial difference between English and Dutch under this analysis is exactly the difference in word order. Because information is presented to the parser in a different order in the two languages, due to Dutch being head-final in embedded clauses and English not, the parser builds different parse trees, yielding different predictions. Thus, there is no need to postulate that different filler-gap processing mechanisms are active in these two languages to account for the available processing data.

5.3 Processing empty categories

In this section I discuss how the parser postulates empty categories, and how it fully instantiates their feature content as processing proceeds.

5.3.1 Creation of trace empty categories

There are two types of empty categories. Traces form the first category. A trace is licit in a syntactic structure only if the structure also contains the trace's antecedent. It is crucial, therefore, that the parser will postulate a trace only if there is already an antecedent for it. This derives the requirement that overt movement be leftward.

Furthermore, if the parser were willing to postulate a trace without having identified an antecedent, then it would be willing to postulate a dependency for which it has no evidence. It is my contention that the human parser is conservative and does not do this.

5.3.2 Postulation of chain-heading empty categories

Empty categories which are not traces head their own chains. They have some intrinsic content to contribute to a syntactic structure. It is important to note that a given language need not license all of the possible chain-heading empty categories shown in (171). For example, English does not license either argument or expletive *pro*, yet Italian licenses both. Exactly what these additional licensing requirements are is something which I do not attempt to answer. I do assume that the parser will check that all relevant licensing requirements are met before a given empty category is postulated.

The parser will postulate a chain-heading empty category if and only if:

- the parser encounters a position which must be filled;
- the parser has not yet encountered any overt material which can fill the position;
- there is no previously occurring element in the parse tree which could have

moved from this position (i.e. it is not possible to postulate a trace in this position); and

- there is a chain-heading empty category which is licensed in the position in question.

5.3.3 Licensing features and empty categories

How are the licensing feature structures of empty categories derived? The licensing features of traces are derived from the licensing features of their antecedents. Recall that a trace is postulated only when an existing chain²⁷ can be extended to a new position, in order that a licensing requirement of the chain be satisfied. The unsatisfied licensing requirements of the postulated trace will be those of the chain, less those feature(s) of the chain which are satisfied in the trace position.

The initial licensing condition structure of a chain-heading empty category is determined by the licensing features assigned to the position in which it is postulated. In other words, if a chain-heading empty category is to be postulated in a position to which only Case is assigned, the licensing feature structure of the empty category will consist only of Case marking, and the empty category postulated is in this situation the empty expletive.

Chain-heading empty categories can be extended just as overt nominals can be. In this way chain-heading empty categories can acquire more licensing features as the parse proceeds.

²⁷The antecedent chain can be a single-membered chain.

5.4 The parsing algorithm and an implementation

5.4.1 The algorithm

The basic operations of the parser are shown in figure 5.2. Extension is ordered before postulation because the parser prefers to satisfy existing licensing requirements to introducing phonetically empty elements into the structure simply to allow discharge of a licensing feature. Figure 5.3 shows the same algorithm using a flowchart.

5.4.2 An implementation

This section gives a brief overview of the implementation of the proposed parsing algorithm. The purpose of implementing the parsing algorithm is as a proof-of-concept. The implementation produces all the parse trees shown in chapters 5 and 6. This should give some reassurance that there are no hidden assumptions required to make the proposal work as advertised.

The implemented parser is by no means a complete parser. It implements only as much as is required to demonstrate the operation of the proposed parsing algorithm.

The parser is implemented in Prolog, and runs under both Quintus and SICStus Prolog. The parser can produce \LaTeX code for use with the *ec\textit{tree}* tree drawing package to draw the final parse trees shown in this dissertation. The parse trees shown are not exactly as those produced by the parser. They have been edited in the following ways to make them more readable:

- Every node is decorated with its licensing feature structure. I have removed the licensing feature structures which are completely negatively specified.
- In general, unless the negative specification of a licensing feature is relevant

Initialization Tokenize the input stream. The stack is initially empty.

While there are more tokens to process, do the following loop. Otherwise, parsing is finished, and the parse tree is situated at the top of the stack.

Reception Read a token from the input stream.

Projection The parser projects structure on the basis of the item's lexical properties. Let CPPM be the newly projected structure.

If stack is non-empty

Repeat

Let TOP be the top element of the stack.

Attachment Try to combine TOP and CPPM. Attachment is done to satisfy constraints on the assignment of licensing features, such as Case or θ marking. These features cannot be left unassigned. The licensing requirements of the constituent being attached may be augmented if needed.

If TOP and CPPM cannot be combined

then ATTACH = false

else

Combine TOP and CPPM. Let CPPM be the resulting structure.

ATTACH = true

Extension *Extension* associates a constituent with more than one position in the phrase structure tree. Extension is gap-driven: whenever a potential gap position is identified, the CPPM is searched for a suitable filler to relate to this position.

A constituent is extended into a gap position to discharge a feature which must be assigned. The licensing feature structure of the constituent may need to be augmented so that the constituent can accept the feature being assigned. A trace of the extended constituent is inserted into the gap position.

Consider all gaps within the newly attached constituent as possible extension sites. Extend into a gap if there is a filler in the CPPM which can discharge all of the licensing features of the gap position. Rank fillers not needing augmentation above those requiring augmentation.

Postulation Postulate phonetically empty elements to allow discharge of any remaining licensing features.

Consider all gaps within the newly attached constituent as possible postulation sites. Postulate a chain-heading empty category if there is an empty category available in the language which can discharge all of the licensing features of the gap position.

While ATTACH is true AND stack is non-empty.

Push the projected structure CPPM onto stack and repeat the loop.

Repeat from **While**.

Figure 5.2: The final version of the algorithm

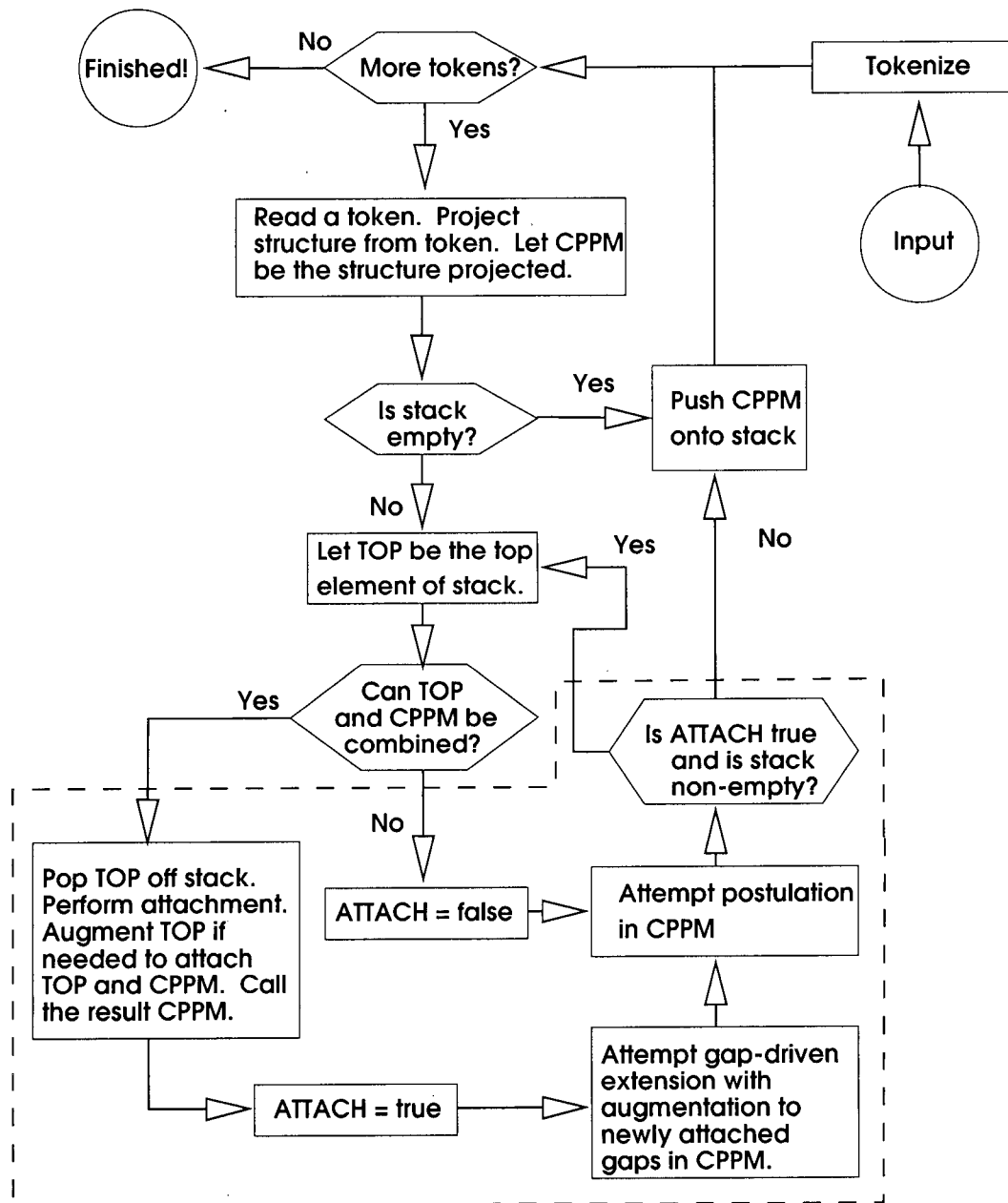
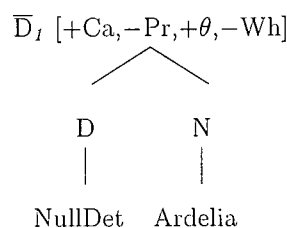


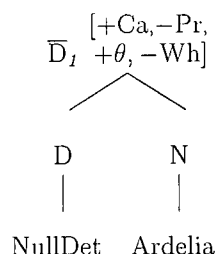
Figure 5.3: The relationship of my contribution (inside dashed line) to a standard licensing parser

to a example, all negative licensing specifications have been removed from the output.

- Because the *ecmtree* package is not very adept at handling long labels associated with nodes in trees, long node labels which the parser produces sometimes run into each other, producing unreadable output. Whenever all four licensing features are shown for a node, I have therefore replaced the standard representation,



with the more compact one shown below:



The main predicates of the parser are common regardless of the language being parsed. The definitions of some predicates are unavoidably dependent on the language being parsed. Most obviously those predicates which are related to the lexicon differ from language to language. Also, those predicates which are related to the tokenization of a language vary according to the language being parsed.

The lexicon encodes the following information for each lexical item:

- syntactic category

- licensing requirements
- licensing features associated with complement position
- licensing features associated with specifier position

The tokenizer performs the following tasks:

- removes any punctuation from the input stream
- separates tense information from the verb
- undoes the effects of head movement
- inserts null determiners as needed:
 - A proper name like *Ardelia* is tokenized as *NullDet Ardelia*.
 - A *wh* phrase like *who* is tokenized as *WhDet Person(x)*.

5.5 An example

To demonstrate the algorithm in action, consider an example where postulation is successful in English. Example (189) is a relative clause lacking both a complementizer and an overt relative pronoun. (189a) is tokenized as in (189b).

- (189) a. The man Ardelia kissed
- b. the man NullDet Ardelia Past kiss

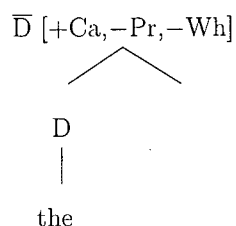
Round 1

Reception Read **the** from the input.

Projection Project $[\bar{D} \text{ the } [N]]$, where **the** f-selects any bar-level projection of N.

Attachment The projected structure becomes the CPPM.

Postulation Nothing to postulate.



Round 2

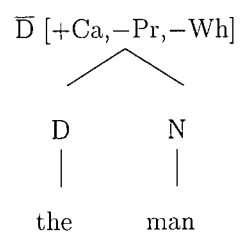
Reception Read **man** from the input.

Projection Project $[NP \text{ man}]$.

Attachment Attachment $[NP \text{ man}]$ to satisfy f-selection requirement of $[DP \text{ the } [NP]]$.

Extension Nothing to extend.

Postulation Nothing to postulate.



Round 3

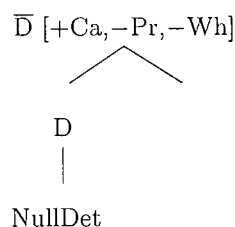
Reception Read **NullDet** from the input.

Projection Project $[\bar{D} \text{ NullDet } [N]]$, where the empty determiner head f -selects any bar-level projection of N .

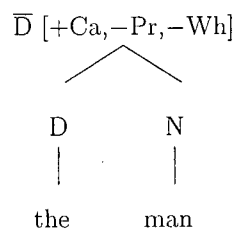
Attachment Because no attachment can take place, the old CPPM is pushed onto the stack, and the newly projected structure becomes the CPPM.

Postulation Nothing to postulate.

stack 1:



stack 2:



Round 4

Reception Read **Ardelia** from the input.

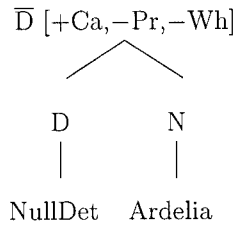
Projection Project $[_{NP} \text{ Ardelia}]$.

Attachment Attachment $[_{NP} \text{ Ardelia}]$ to satisfy f -selection requirement of $[_{DP} \text{ NullDet } [_{NP}]]$.

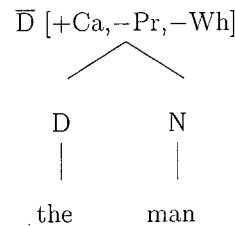
Extension Nothing to extend.

Postulation Nothing to postulate.

stack 1:



stack 2:



Round 5

Reception Read **Past** from the input.

Projection Project $[\text{IP } [+Ca] [\bar{I} [\text{I Past}] [-N, +V] e]]]$.

Attachment Attach $[\bar{D} \text{ NullDet Ardelia}]$ as specifier of IP.

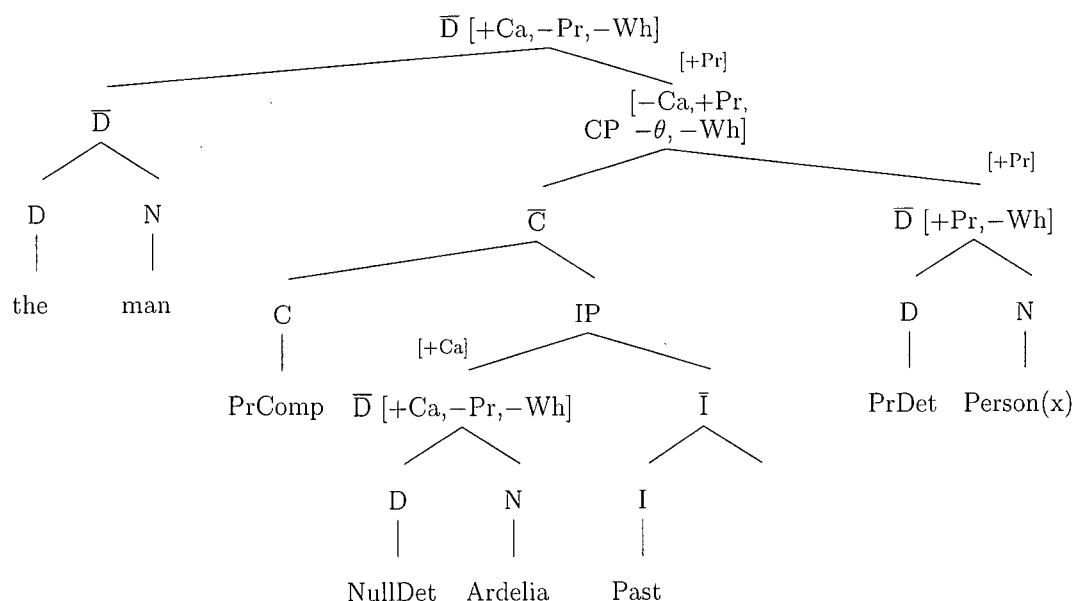
Extension Nothing to extend.

Postulation Nothing to postulate.

Attachment IP cannot be directly licensed with respect to the DP on the stack, but a full CP could be licensed via predication as a relative clause modifying DP. A full CP is therefore projected. This CP is augmented with the feature $[+Pr]$. Via the mechanisms of feature percolation and Specifier-Head agreement, the C head assigns the feature $[+Pr]$ to its specifier.

Extension Nothing to extend.

Postulation Postulate a $[-Wh, +Pr]$ operator in the spec of CP. Notice that in the tree diagrams this null operator is shown on the right periphery of the CP. There is no significance to this ordering. The underlying representation is strictly hierarchical. When printing out a tree structure, the parser attempts to maintain the order of lexical items in the tokenized input. Null elements postulated by the parser, which do not have a “position” specified in the tokenized input stream, are generally drawn to the right periphery of their maximal projections.



Round 6

Reception Read **kiss** from the input.

Projection Project VP.

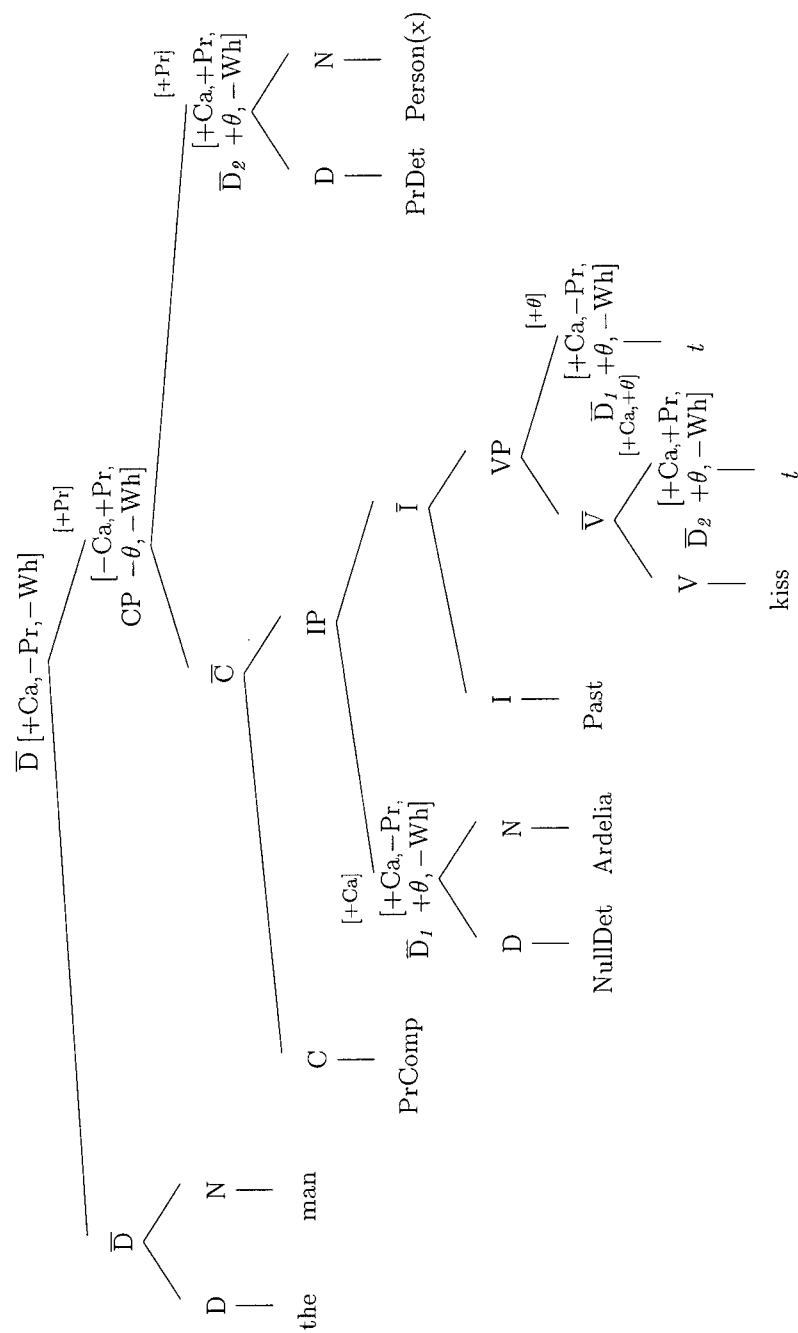
Attachment Attach VP as complement of I.

Extension There are two positions to consider with respect to extension: the VP-internal subject position and the object position. There are two sources for extension: $[\overline{D} \text{ Ardelia}]$ and $[\text{DP } O]$. Because $[\overline{D} \text{ Ardelia}]$ is Case marked, it can only be extended into the VP-internal subject position, thereby satisfying a θ assignment requirement on the verb, and satisfying the θ reception requirement on the \overline{D} . The operator can be extended into the object position, satisfying both the Case and θ discharge requirements of the verb.

Postulation Nothing to postulate.

The resulting tree is shown on the next page.

Finished! Since the input stream is empty and all the licensing requirements of the structure we have built are satisfied, this parse comes to a successful conclusion.



5.6 Processing vs. grammatical exclusion

Strings which are not part of a language can be excluded by either the grammar of the language or the parser, or both. My thesis is that overt rightward movement configurations are disallowed by the parser, but not (necessarily) by the grammar.

A question which is naturally raised of this work²⁸ is whether there is any independent evidence to argue for a causality in this explanation, and if so, what the direction of the causality is. The issue is whether there is evidence to suggest that the operation of the parser has been shaped by the grammar, or the other way around.

There is evidence to suggest in general that the grammar and parser influence each other (Hawkins 1994). I have argued that there is no easy (non-stipulatory) way to express in grammatical terms that overt rightward movement is to be ruled out. I claim that it is the combination of no lookahead, the gap-driven mechanism for filler-gap processing and the structure building mechanism which explains why overt rightward movement is unparsable as such. These properties of the parser are motivated by psycholinguistic evidence:

- The property of no-lookahead is motivated by psycholinguistic evidence rooted in the work on garden paths.
- The gap-driven nature of filler-gap processing is motivated by psycholinguistic evidence related to the filled-gap effect.
- The most important structure-building property is determinism. This is also motivated by work on the processing of garden paths.

²⁸Lyn Frazier, p.c.

None of these properties of the parser are universally accepted, but neither are their alternatives. I have argued that that while there is no satisfying (non-stipulatory) way of expressing the constraint on overt rightward movement in terms of grammar, there is a psycholinguistically plausible model for parsing which does offer an explanantion.

Chapter 6

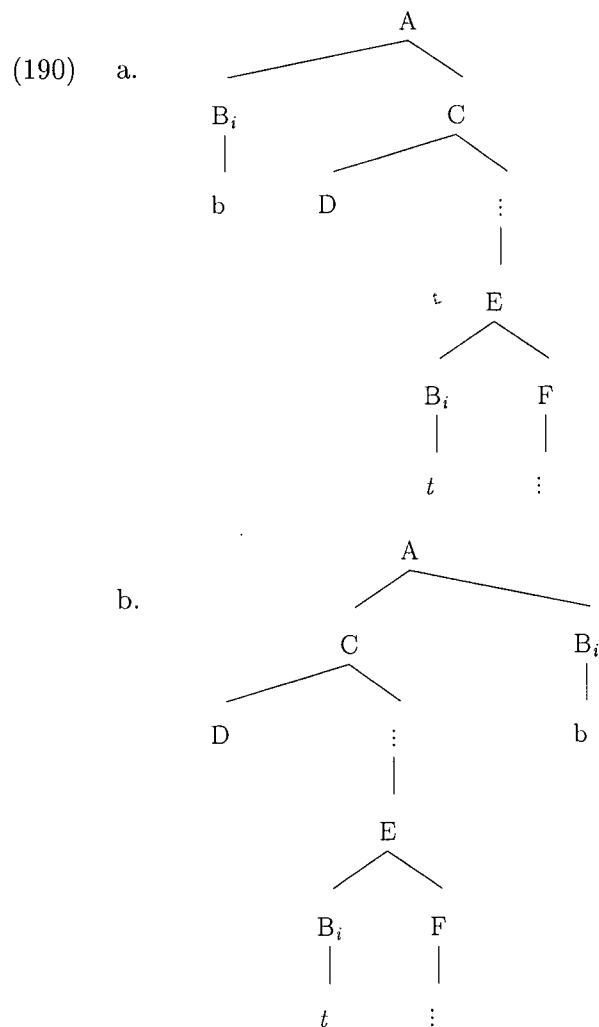
Testing the thesis

In chapter 5 I developed a parsing algorithm based on processing models which view parsing as an incremental on-line process. At each stage of computation the structure built is the smallest superset of the structure at the previous stage which is maximally well-formed. My contribution to this model is a monotonic, incremental mechanism for interpreting empty categories.

My thesis is that the leftwardness constraint on overt movement should be given a processing explanation rather than a syntactic explanation. I claim that the manner in which the parser handles empty categories derives the requirement that overt movement must be leftward.

In this chapter I test this claim by showing that a representative set of leftward movements can be parsed without difficulty. I also demonstrate why a rightward movement configuration cannot be parsed. More precisely, my claim is that a syntactic structure which involves a movement relation can only be linearized in such a way that the filler for the gap (the head of the movement chain) is identified before the parser is forced to make a final decision concerning the status of the gap position (the foot of the movement chain). To illustrate my claim, consider the two

structures shown in (190a-b).



The two trees shown share exactly the same hierarchical structure. Only the linearization of the constituent $[B\ b]$ is different in the two cases. In (190a) $[B\ b]$ appears to the left of the rest of the tree, while in (190b) $[B\ b]$ appears to the right. My claim is that only with the linearization shown in (190a) will the parser be able to recover the given hierarchical structure. Presented with the linearization shown in (190b) the parser either cannot recover any coherent structure, or else it recovers a different hierarchical structure (with a different interpretation). Which of these

two cases is realized depends on particular details of the construction and language under consideration.

6.1 Overt movements

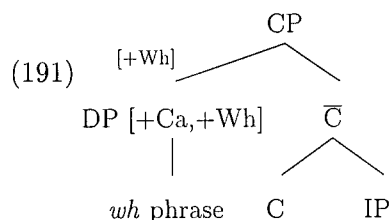
6.1.1 Overt *wh* movement

An overt *wh* phrase can be licensed in a number of ways. It can be licensed as an interrogative pronoun in the specifier of a non-relative C or as a relative pronoun in the specifier of a relative C. In the former case, it is endowed with the features $[+Wh, -Pr]$, whereas in the latter case it bears the features $[+Wh, +Pr]$. A *wh* phrase can also be licensed in-situ, where it can be given a multiple question interpretation, or an echo question interpretation. Since a discussion of the in-situ possibilities is outside the scope of this dissertation, the main issue to consider is whether a *wh* phrase is interpreted as an interrogative or relative operator.

Overt *wh* movement is perhaps the easiest movement for the parser to deal with. The moved element is easily identifiable: the $[+wh]$ feature which it bears is marked in its phonological form. The $[+wh]$ feature identifies the *wh* phrase as an operator, and indicates to the parser that the *wh* phrase must be attached into an operator position.

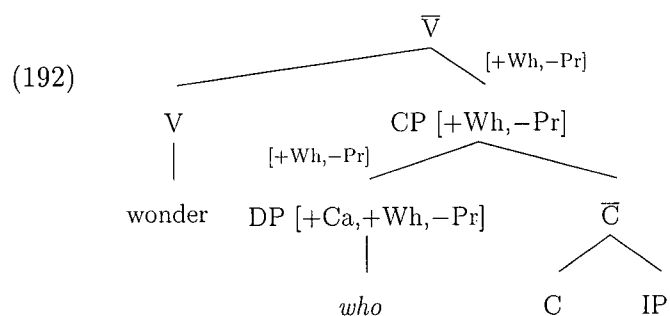
Since *wh* phrases can be interpreted with either an interrogative or relative function depending on the context in which they appear, *wh* phrases are underspecified in the lexicon with respect to the feature $[\pm Pr]$. Moreover, *wh* phrases are overt and must therefore bear Case. Hence, the lexically-specified licensing requirements of *wh* phrases is $[+Wh, +Ca]$.

A typical attachment scenario for a *wh* phrase is shown in (191).

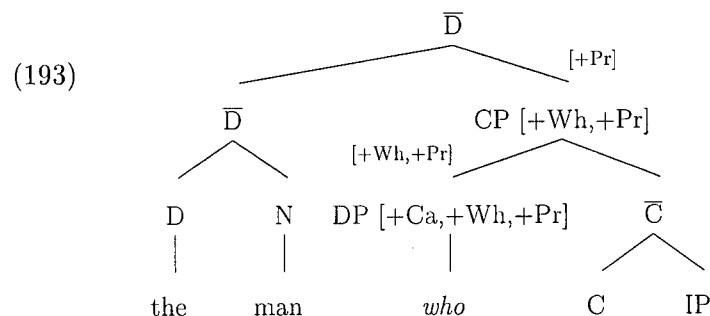


A *wh* phrase must be fully specified with respect to the [Pr] feature. It can be augmented with a [\pm Pr] specification in the environment of a [\pm Pr] C. C is [+Pr] only if its projection is licensed via predication. This case is considered in the next subsection. If a projection of C is not licensed via predication, C is [-Pr].

(192) exhibits part of the structure of a sentence such as *John wonders who Mary ...*, and shows a *wh* phrase attached as an interrogative operator.



(193) exhibits part of the structure of a sentence such as *The man who Mary likes ...*, and is an example of a context in which a relative reading results for the *wh* phrase.



I return to the relative reading in section 6.1.2. In the next two subsections I show how the implemented parser handles the examples (194a–b). These examples involve question formation from object and subject position respectively. I work through both examples in some detail because the cues for the parser to build a projection of C are different in the two cases.

- (194) a. What did Wolfgang buy?
 b. Who bought a lollipop?

Question from object

Recall that I do not handle head movement, so the tokenization stage separates the tense information from *did*, and places it in the correct position in the input stream (as if head movement had not occurred). (195a) is tokenized as in (195b).

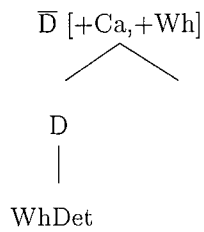
- (195) a. What did Wolfgang buy?
 b. WhDet Thing(*x*) do NullDet Wolfgang Past buy

Round 1

Reception Read **WhDet** from the input.

Projection Project $[\overline{\text{D}} \text{ WhDet}]$ with licensing requirement $[+\text{Ca}, +\text{Wh}]$. The head of this phrase functionally selects a projection of N.

Attachment There is nothing to attach this structure to, so it simply becomes the CPPM.



Round 2

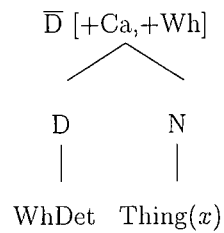
Reception Read **Thing**(x) from the input.

Projection Project $[_N \text{ Thing}(x)]$.

Attachment Projected structure becomes the CPPM. Attach CPPM into the structure at the top of the stack to satisfy the functional selection requirement of D.

Extension Nothing to extend.

Postulation Nothing to postulate.



Round 3

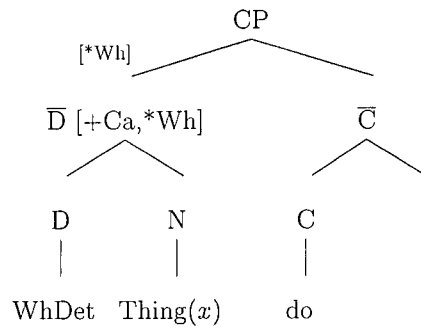
Reception Read **do** from the input.

Projection Project $[_{CP} \text{ do}]$ with licensing features $[+Wh, -Pr]$ in specifier.

Attachment Augment DP with $[-Pr]$, then attach into specifier of CP.

Extension Nothing to extend.

Postulation Nothing to postulate.



Round 4

Reception Read **NullDet** from the input.

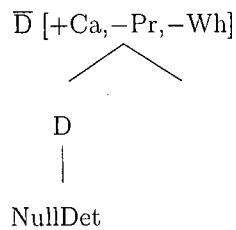
Projection Project [\overline{D} NullDet] with licensing requirement [+Ca, -Pr, -Wh].

The head of this phrase functionally selects a projection of N.

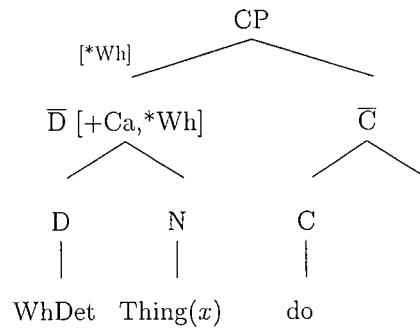
Attachment [\overline{D} NullDet] cannot be attached into the current structure. The projected structure becomes the CPPM. Push the CPPM onto the stack.

The stack now has two structures.

stack 1:



stack 2:



Round 5

Reception Read **Wolfgang** from the input.

Projection Project $[_N \text{ Wolfgang}]$ with a negative licensing specification.

Attachment Projected structure becomes the CPPM. Attach CPPM into the structure at the top of the stack to satisfy the functional selection requirement of D.

Extension Nothing to extend.

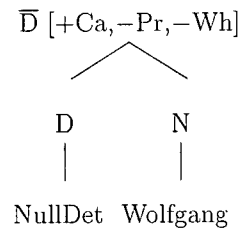
Postulation Nothing to postulate.

Because stack is not empty, attempt attachment again.

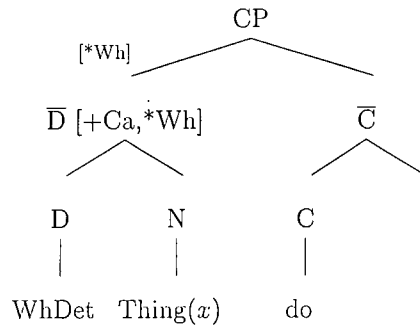
Attachment The two structures in the stack still cannot be combined.

Postulation Nothing to postulate.

stack 1:



stack 2:



Round 6

Reception Read **Past** from the input.

Projection Project $[_{IP} \text{ Past}]$, assigning $[+Ca]$ to its specifier, and functionally selecting a projection of V as complement.

Attachment Projected structure becomes the CPPM. Attach the structure at the top of the stack into the CPPM to satisfy the common $[+Ca]$ requirement. The resulting structure becomes the CPPM.

Extension Nothing to extend.

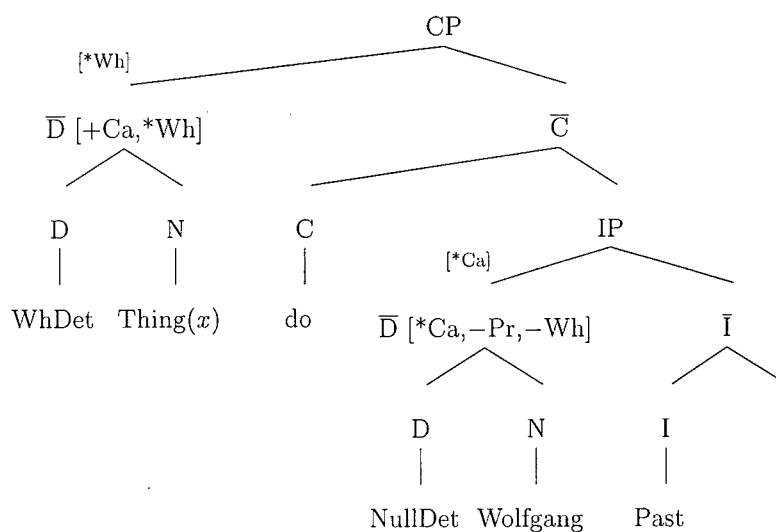
Postulation Nothing to postulate.

Because stack is not empty, attempt attachment again.

Attachment Attach the CPPM into the structure at the top of the stack to satisfy the functional selection requirement of C.

Extension Nothing to extend.

Postulation Nothing to postulate.



Round 7

Reception Read **buy** from the input.

Projection Project $[\text{VP buy}]$ with licensing features $[+\theta]$ in specifier and $[+\theta, +\text{Ca}]$ in complement.

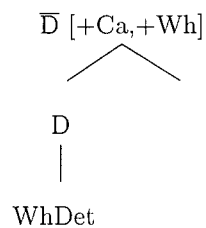
Attachment Attach VP as complement of I.

Extension Augment $[\text{D̄ NullDet Wolfgang}]$ with $[+\theta]$ and extend into specifier of VP. Extend $[\text{D̄ WhDet Thing(x)}]$ into complement of VP.

Postulation Nothing to postulate.

Projection Project $[\overline{D} \text{ WhDet}]$ with licensing requirement $[+Ca, +Wh]$. The head of this phrase functionally selects a projection of N.

Attachment Projected structure becomes the CPPM.



Round 2

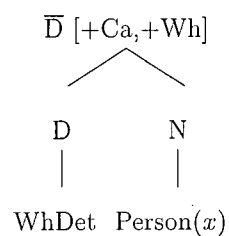
Reception Read **Person**(x) from the input.

Projection Project $[_N \text{ Person}(x)]$ with a negative licensing specification.

Attachment Projected structure becomes the CPPM. Attach CPPM into the structure at the top of the stack to satisfy the functional selection requirement of D.

Extension Nothing to extend.

Postulation Nothing to postulate.



Round 3

Reception The next token in the input stream is **Past**, or category Inflection.

The parser inserts a **WhComp** token into the input stream because,

in the context of Inflection, the preceding *wh* phrase is not in-situ and must therefore occupy a specifier of C position. Since no C has been encountered, the parser inserts an empty C.

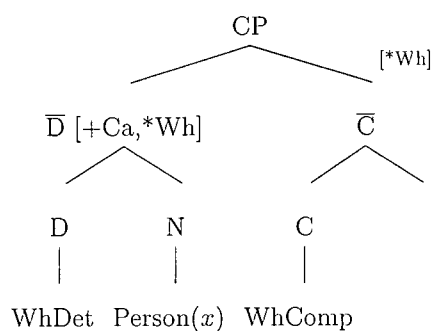
Projection Project [_{CP} WhComp] with licensing features [+Wh] in specifier.

The C functionally selects a projection of I.

Attachment Attach \overline{D} into specifier of CP.

Extension Nothing to extend.

Postulation Nothing to postulate.



Round 4

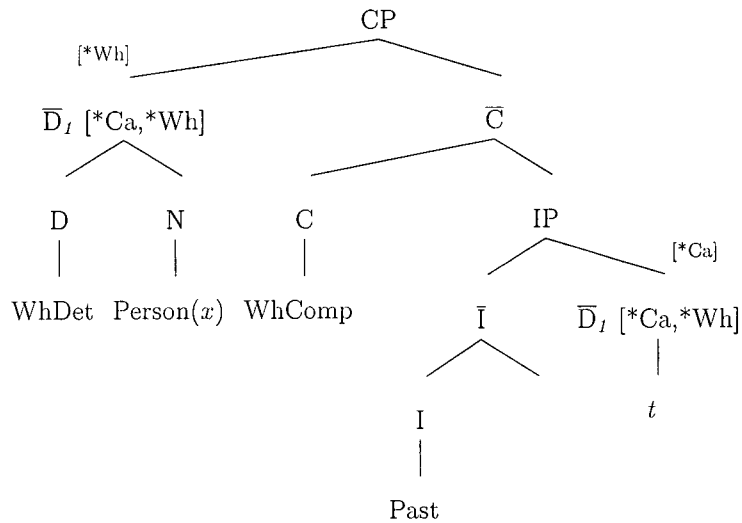
Reception Read **Past** from the input.

Projection Project [_{IP} Past], assigning [+Ca] to its specifier, and functionally selecting a projection of V as complement.

Attachment Projected structure becomes the CPPM. Attach the CPPM into the structure at the top of the stack to satisfy the functional selection requirement of C.

Extension Extend the (single-membered) *wh* chain into the specifier of I position in order to discharge the [+Ca] feature.

Postulation Nothing to postulate.



Round 5

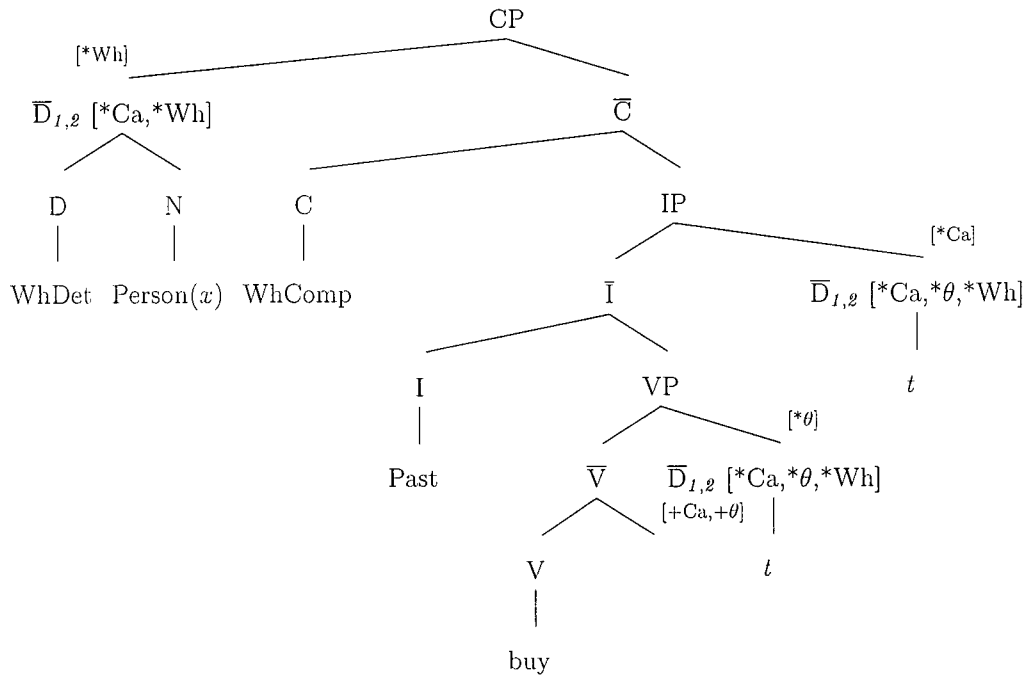
Reception Read **buy** from the input.

Projection Project $[_{VP} \text{ buy}]$ with licensing features $[+\theta]$ in specifier and $[+\theta, +Ca]$ in complement.

Attachment Attach VP as complement of I.

Extension Extend the *wh* chain into the specifier of V to discharge the $+\theta$ feature. Notice that the *wh* phrase cannot be extended into the object position since there is a $[+Ca]$ feature assigned there, and the *wh* chain already has its Case requirement satisfied.

Postulation The object position of the verb is considered as a target for postulation. However, English does not license any chain-heading empty categories in this position, so no empty category is postulated.



Round 6

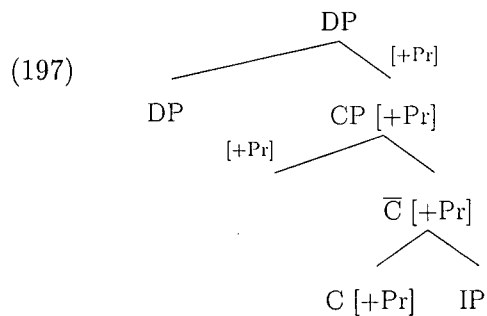
Reception Read **a** from the input.

Projection Project $[\bar{D} \ a]$ with licensing requirement $[+Ca, -Pr, -Wh]$. The head of this phrase functionally selects a projection of N.

Attachment Projected structure becomes the CPPM. The CPPM is attached into the verb's object position. In order for this attachment to proceed, the CPPM must first be augmented with a $[+\theta]$ requirement.

Extension Nothing to extend.

Postulation Nothing to postulate.

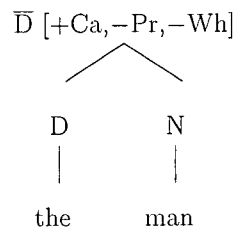


Let us now consider the parsing of the example in (198a), which is tokenized as shown in in (198b).

- (198) a. The man who Ardelia kissed.
 b. the man WhDet Person(x) NullDet Ardelia Past kiss

Rounds 1–2

After reading and processing the tokens **the man** from the input, the parser has built the following structure:



Round 3

Reception Read **WhDet** from the input.

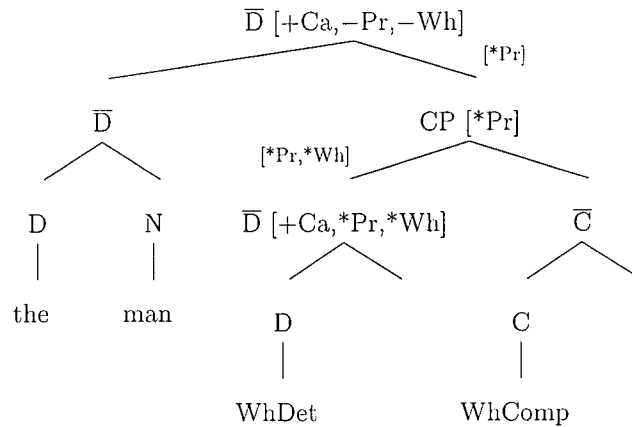
Projection Project [\overline{D} WhDet] with licensing requirement [+Ca, +Wh]. The head of this phrase functionally selects a projection of N.

Attachment [\overline{DP} the man] can optionally license a clause via predication. [\overline{D} WhDet] can be licensed as a relative pronoun. The only way to combine

the two constituents into one structure at this point is for the parser to assume that **WhDet** signals the start of a relative clause predicated of **the man**.

Extension Nothing to extend.

Postulation Nothing to postulate.



Round 4

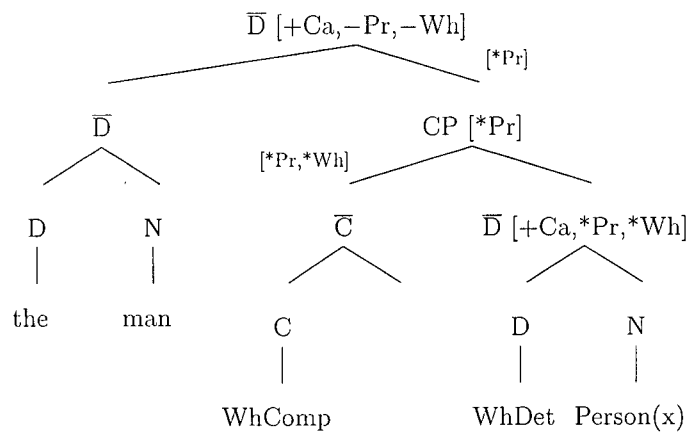
Reception Read **Person**(x) from the input.

Projection Project $[_N \text{ Person}(x)]$ with a negative licensing specification.

Attachment Projected structure becomes the CPPM. Attach CPPM into the structure at the top of the stack to satisfy the functional selection requirement of D.

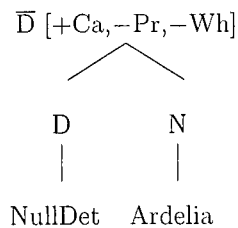
Extension Nothing to extend.

Postulation Nothing to postulate.



Rounds 5–6

Read the two tokens **NullDet** and **Ardelia** from the input. Because these items cannot be incorporated into the existing structure, the existing structure is pushed onto the stack. The following structure is built to house these two tokens.



Round 7

Reception Read **Past** from the input.

Projection Project $[_{IP} \text{ Past}]$.

Attachment Projected structure becomes the CPPM. Attach the structure at the top of the stack into the CPPM to satisfy the common $[+Ca]$ requirement. The resulting structure becomes the CPPM.

Extension Nothing to extend.

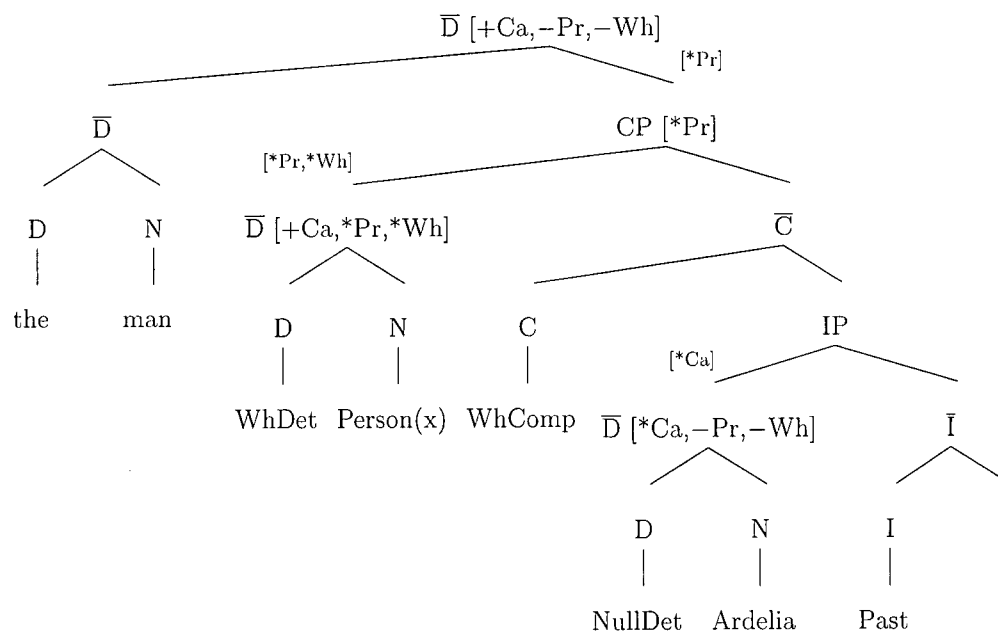
Postulation Nothing to postulate.

Because stack is not empty, attempt attachment again.

Attachment Attach the CPPM into the structure at the top of the stack to satisfy the functional selection requirement of C.

Extension Nothing to extend.

Postulation Nothing to postulate.



Notice the lack of a filled-gap effect with subject position. Since overt material is attached before extension is considered, a gap will never be considered for subject position if overt material is available.

Round 8

Reception Read **kiss** from the input.

Projection Project [_{VP} kiss] with licensing features [$+\theta$] in specifier and [$+\theta, +Ca$] in complement.

Attachment Attach VP as complement of I.

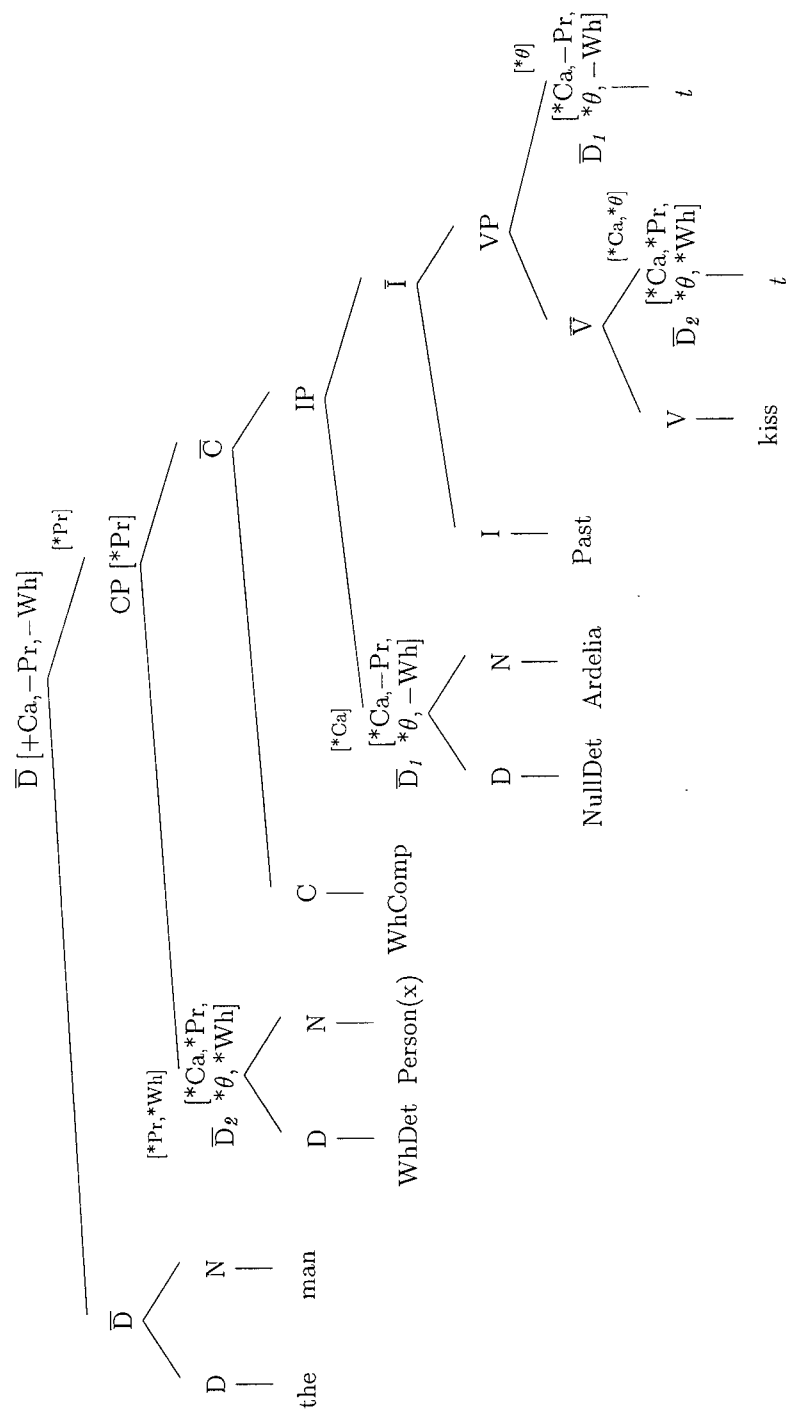
Extension Augment [_{DP} Ardelia] with [$+\theta$] and extend into specifier of VP.

Extend [_{\overline{D}} who] into complement of VP.

Postulation Nothing to postulate.

The resulting tree is shown on the next page.

Finished! Since the input stream is empty and all the licensing requirements of the structure we have built are satisfied, this parse comes to a successful conclusion.



6.1.3 Topicalization

Topicalization (199) and left dislocation (200) are superficially similar processes, yet they differ in their syntax. Topicalization involves an \bar{A} movement dependency, whereas left dislocation does not. In section 7.4 I revisit the processing of these two constructions.

(199) John, Mary admires.

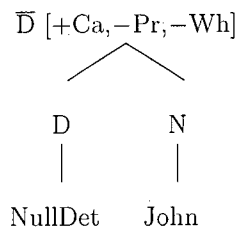
(200) John, Mary admires him.

In this section I simply demonstrate how the algorithm deals with topicalization. (199) is tokenized as shown in (201).

(201) NullDet John NullDet Mary Present admire

Rounds 1–2

The parser reads and processes the tokens **NullDet** and **John** from the input, building the structure shown below.



Round 3

Reception Read **NullDet** from the input.

Projection Project [\bar{D} NullDet] with licensing requirement [+Ca, -Wh, -Pr], functionally selecting for a nominal complement.

Attachment There is no way to license the two constituents with respect to each other at this point. They remain unattached. $[\overline{D} \text{ NullDet John}]$ is pushed onto a stack, and $[\overline{D} \text{ NullDet}]$ becomes the CPPM.

Round 4

Reception Read **Mary** from the input.

Projection Project $[_N \text{ Mary}]$.

Attachment $[_N \text{ Mary}]$ becomes the CPPM. Attach the CPPM into the structure at the top of the stack to satisfy the functional selection requirement of D.

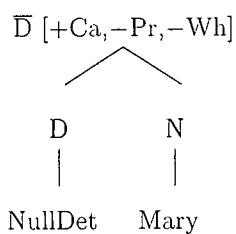
Extension Nothing to extend.

Postulation Nothing to postulate.

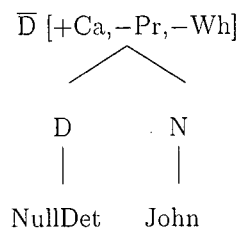
Because stack is not empty, attempt attachment again.

Attachment The two \overline{D} level categories still cannot be attached.

stack 1:



stack 2:



Round 5

Reception Read **Present** from the input.

Projection Project [_{IP} Present] with licensing requirement [+Ca] in specifier position, functionally selecting a projection of V as complement.

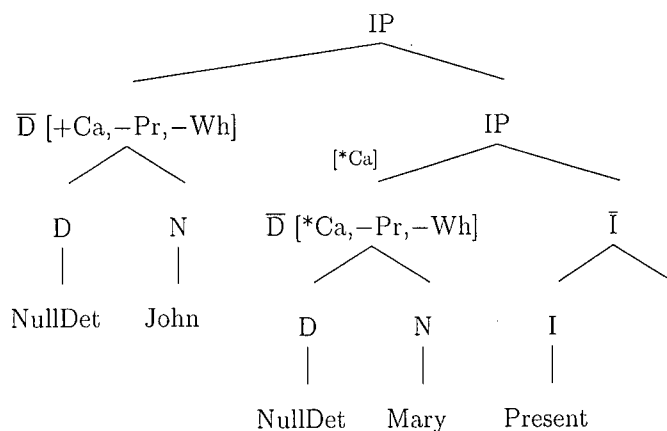
Attachment Attach [_{D̄} NullDet Mary] into specifier position to discharge the [+Ca] licensing feature.

Extension Nothing to extend.

Postulation Nothing to postulate.

Because stack is not empty, attempt attachment again.

Attachment Adjoin [_{D̄} Null John] to IP. This move does not satisfy any of the [\pm Ca, $\pm\theta$, \pm Wh, \pm Pr] licensing requirements of either constituent, but it is licensed by the stress/pause structure.



Round 6

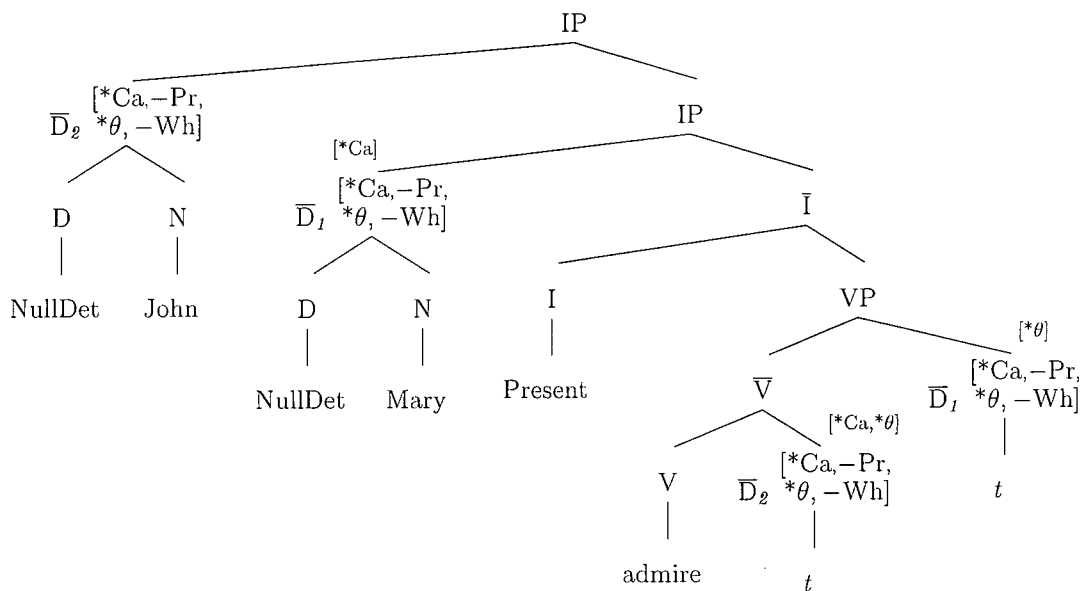
Reception Read **admire** from the input.

Projection Project [_{VP} admire] with licensing requirement [+ θ] in specifier position.

Attachment Attach VP as complement of IP.

Extension Extend $[\bar{D} \text{ NullDet Mary}]$ into specifier of V position. Extend $[\bar{D} \text{ NullDet John}]$ into the object position of V (augmenting with a θ requirement).

Postulation Nothing to postulate.



Finished! Since the input stream is empty and all the licensing requirements of the structure we have built are satisfied, this parse comes to a successful conclusion.

6.2 Covert movements

The next type of movement to consider is that of a phonetically empty operator. In English, relative pronouns can be phonetically empty. In Chinese, relative pronouns are always phonetically empty. English relative pronouns are licensed in operator

position, and are further identified in argument position. Chinese relative pronouns are licensed in argument position, and are further identified in operator position.

6.2.1 Licensed in specifier of C position

In this subsection I will discuss the processing of a relative clause with no overt relative pronoun. This type of relative clause shares its basic word order with topicalization and left dislocation. The topicalized or left dislocated noun phrase is separated from the remainder of the utterance by a pronounced pause and heavy stress.¹ No such separation occurs in the case of the head of a relative clause. I assume that there is no ambiguity to resolve in the case of a relative clause.

The example whose parse is given below is shown in (202a). As usual, the tokenization of the example is shown in (202b).

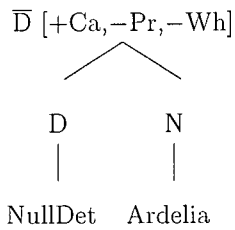
- (202) a. The boy Ardelia tutors graduated.
 b. the boy NullDet Ardelia Present tutor Past graduate

Rounds 1–4

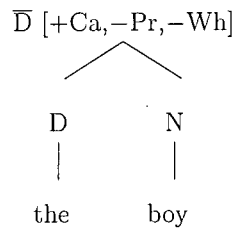
Read the tokens **the boy NullDet Ardelia** from the input. [\overline{D} The boy] is pushed onto the stack. [\overline{D} NullDet Ardelia], the CPPM, is also pushed onto the stack since the two \overline{D} structures cannot be combined.

stack 1:

¹This difference is noted in writing by the use of a comma to separate the topicalized or left dislocated phrase from the main clause.



stack 2:



Round 5

Reception Read **Present** from the input.

Projection Project [_{IP} Present] with licensing requirement [+Ca] in specifier position, and functionally selecting a projection of V as complement.

Attachment Attach [_D NullDet Ardelia] into specifier position to discharge the [+Ca] licensing feature.

Extension Nothing to extend.

Postulation Nothing to postulate.

Because stack is not empty, attempt attachment again.

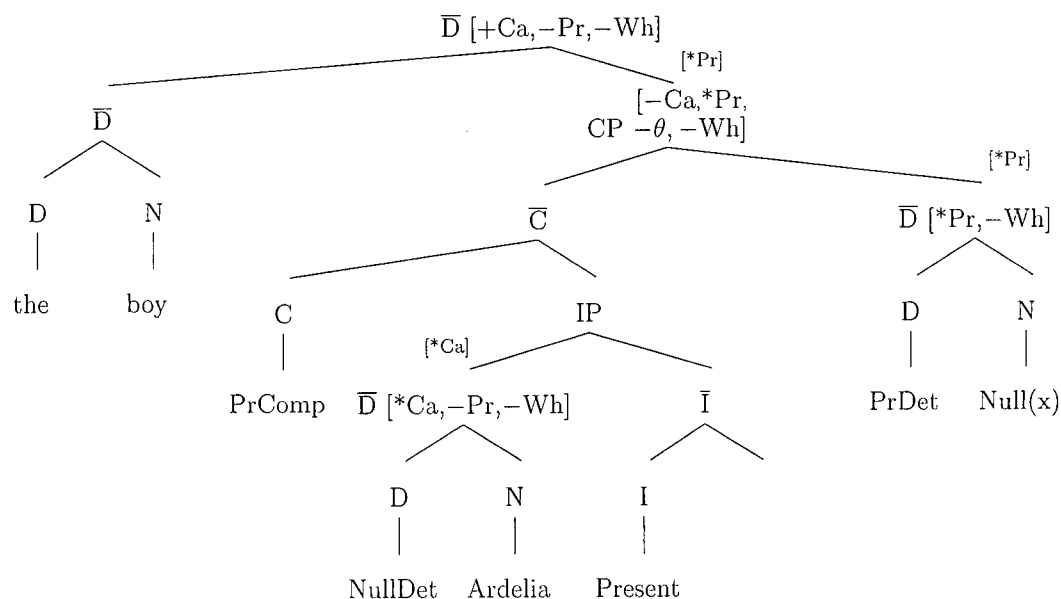
Attachment [_D the boy] does optionally license a clause via predication.

The extended projection of V must be made maximal (i.e. a CP must be projected). Since the licensing mechanism is predication, C must be [+Pr]. Since there is no overt operator, C must be [-Wh].

Extension Nothing to extend.

Postulation A chain-heading empty category is postulated with licensing requirements $[+Pr, -Wh]$.

Notice that the postulated empty operator $[\overline{D} \text{ PrDet Null}(x)]$ is shown on the right in the tree diagram. This is simply an artefact of the way the parser linearizes its output trees. From the parser's perspective the structure does not encode any linear relations.



Round 6

Reception Read **tutor** from the input.

Projection Project $[\text{VP tutor}]$ with licensing requirements $[+\theta]$ in specifier position, $[+\theta, +Ca]$ in the complement position.

Attachment Attach VP as complement of IP.

Extension There are two positions in the VP to consider. $[\overline{D} \text{ NullDet Ardelia}]$ can only be extended into the specifier, because of the $[+Ca]$ specification

in the complement. The empty relative operator can be augmented with both $[+Ca]$ and $[+\theta]$ and extended into the complement.

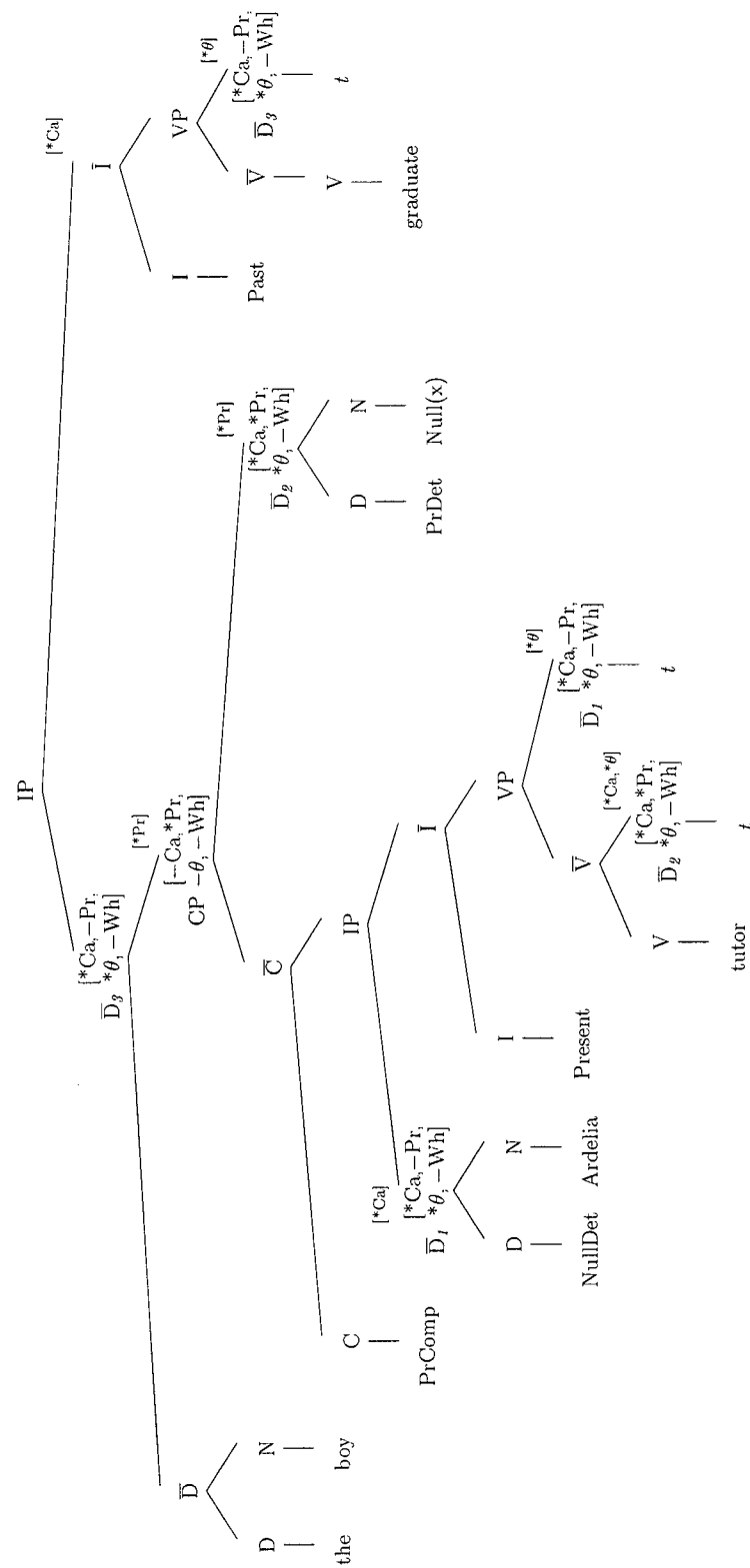
Postulation Nothing to postulate.

The resulting tree is shown on the next page.



Rounds 7–8

Read and process the remaining two tokens, **Past graduate**. The resulting tree is shown on the next page.



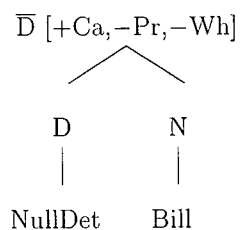
6.2.2 Licensed in argument position

As noted above, in English a *wh* word like *who* can be used in both relative and interrogative clauses. Chinese, in contrast, behaves differently. There is no movement in Chinese question formation. Chinese does not have overt relative pronouns, and relative clauses are also prenominal, rather than postnominal as in English.² Consider an example like that shown in (203). Since Chinese licenses empty pronouns, the parser postulates one in the object position of the subordinate clause. This empty pronoun can be augmented to become a relative operator when the relative complementizer is encountered.

- (203) a. Bill du de zheben shu.
 Bill read that this book
 The book that Bill read
- b. NullDet Bill du NullASP de zheben shu

Rounds 1–2

The parser reads and processes the tokens **NullDet** and **Bill** from the input stream. The following structure is built.



Round 3

²I discuss the relationship between the order of the head and the relative clause and the use of overt versus covert relative pronouns in chapter 7.

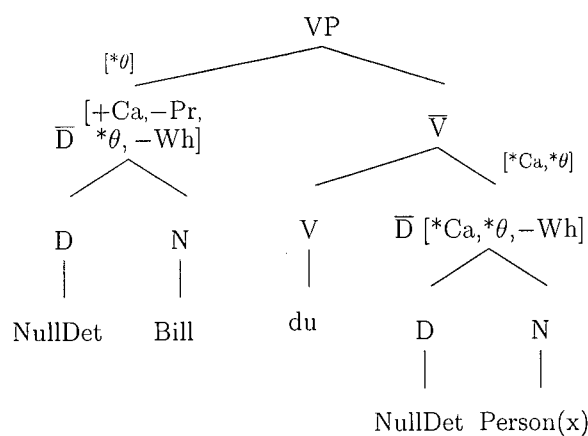
Reception Read the verb **du** from the input.

Projection Project $[\text{VP du}]$ with licensing requirements $[+\theta]$ in specifier position, $[+\theta, +\text{Ca}]$ in the complement position.

Attachment The constituent $[\overline{\text{D}} \text{ NullDet Bill}]$ has only a requirement that it be assigned Case. Both of the available attachment sites within the VP structure require that the $\overline{\text{D}}$ be augmented with θ . In a situation like this the parser makes a choice based on height in tree. This heuristic for choosing an attachment site is somewhat arbitrary. While not very satisfying, this solution does make the correct decision for Chinese.

Extension Nothing to extend.

Postulation Because the object position in the VP is still unfilled, and because Chinese licenses empty object pronouns, the parser postulates an empty pronoun $[\overline{\text{D}} \text{ NullDet Person}(x)]$ with licensing specification $[+\text{Ca}, +\theta]$.



Round 4

Reception Read the null aspectual **NullASP** from the input. As far as the parser is concerned, the null aspectual plays the same role in Chinese as

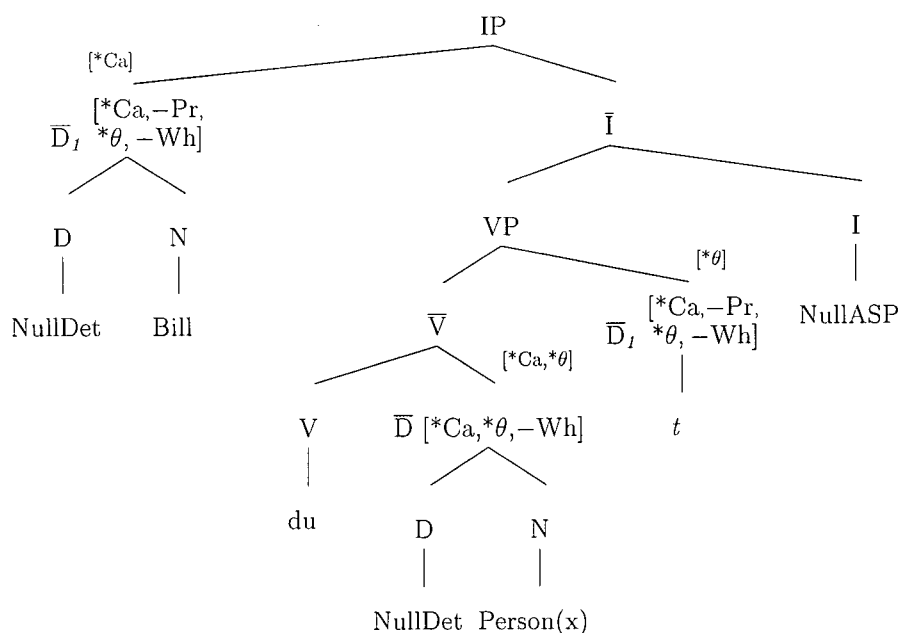
a tense marking does in English: it signals the existence of a projection of Inflection.

Projection Project $[_{IP} \text{ NullASP}]$. Case is assigned to the specifier position, and the head functionally selects for a projection of V.

Attachment The VP is attached as the complement of I.

Extension The \bar{D} in the VP-internal subject position is extended to occupy the specifier of I position in order to get Case.

Postulation Nothing to postulate.



Round 5

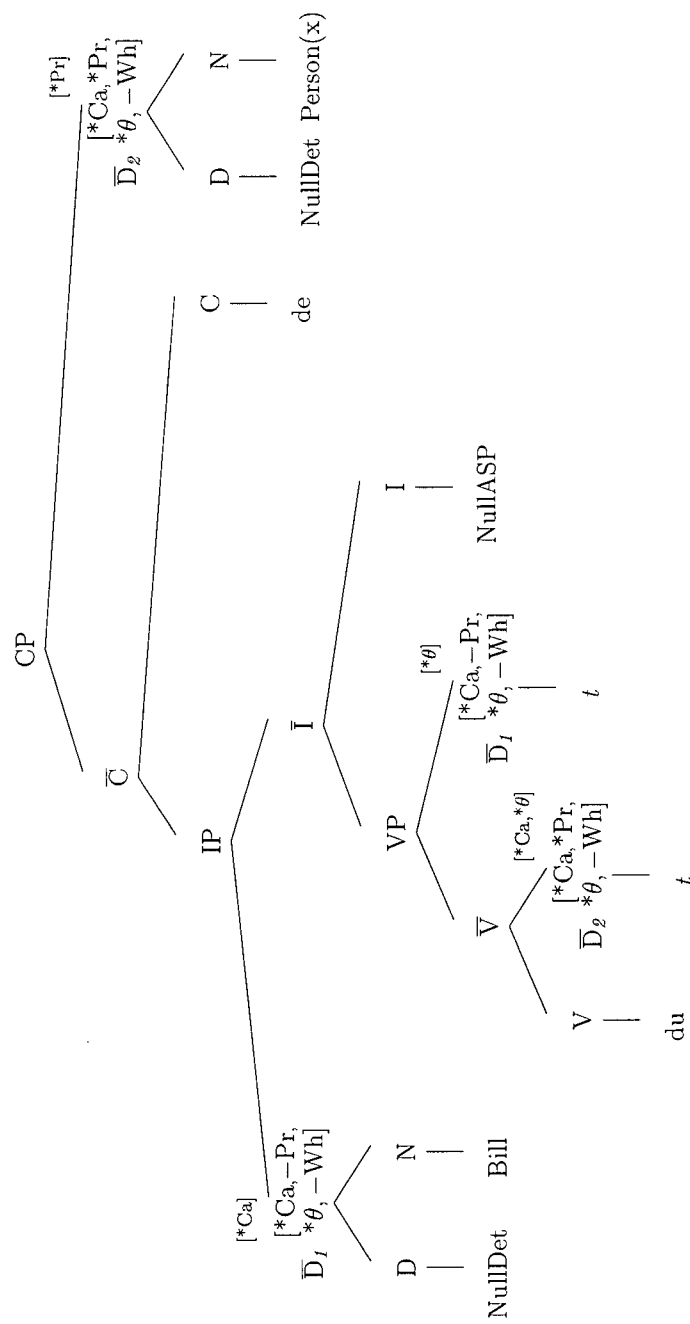
Reception Read the relative complementizer **de** from the input.

Projection Project a predication clause headed by the relative complementizer: $[_{CP} \text{ de}]$.

Attachment The IP is attached as the complement of C.

Extension The \bar{D} in the VP-internal object position is augmented with operator features and then extended to occupy the specifier of C position.

Postulation Nothing to postulate.



Round 6

Reception Read the token **zheben** from the input.

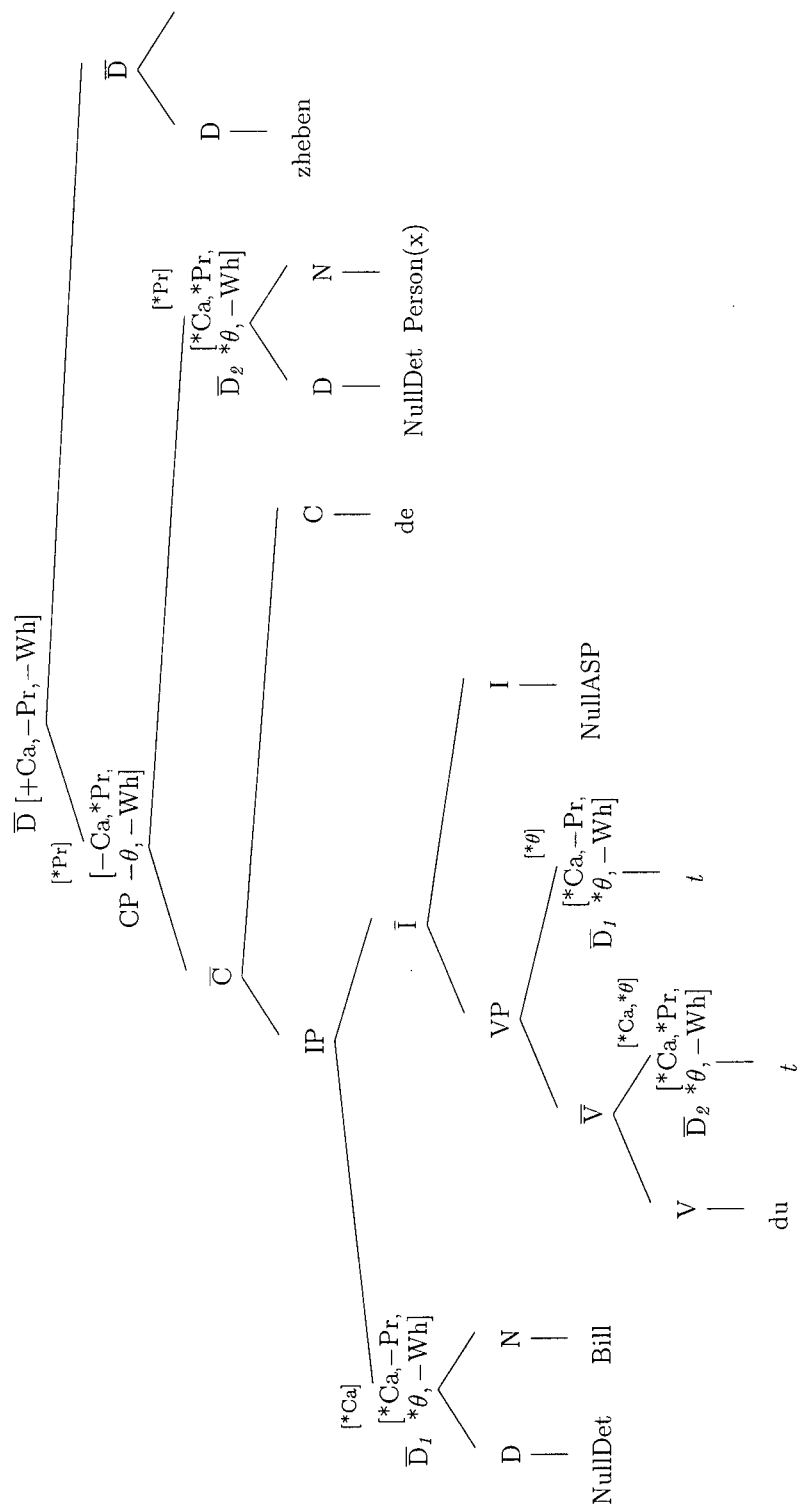
Projection Project $[\overline{D}$ zheben].

Attachment $[\overline{D}$ zheben] does optionally license a clause via predication.

Attach the existing CP as a relative clause modifying $[\overline{D}$ zheben].

Extension Nothing to extend.

Postulation Nothing to postulate.



Round 7

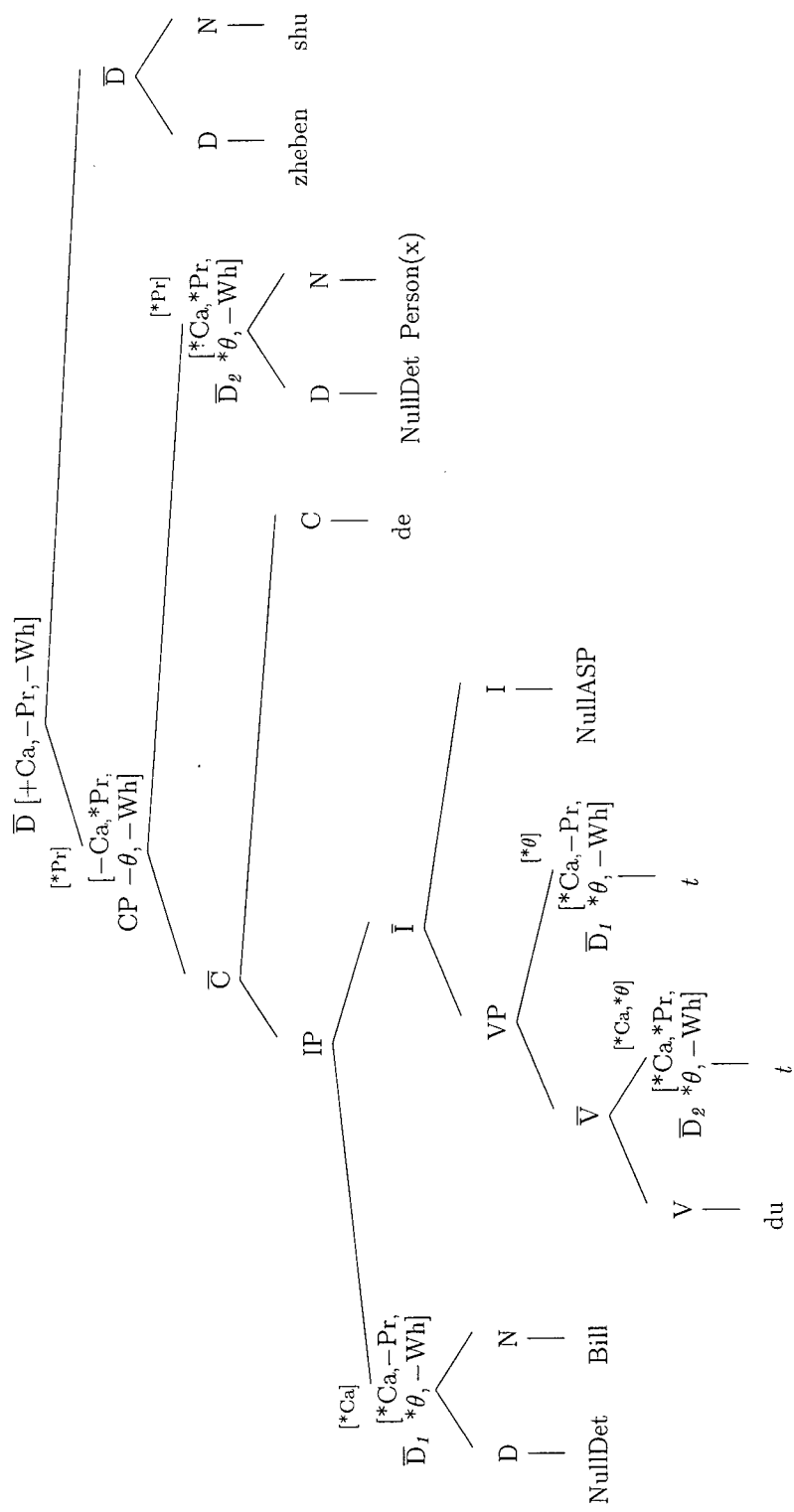
Reception Read the token **shu** from the input.

Projection Project $[_N \text{ shu }]$.

Attachment $[_N \text{ shu }]$ as complement of $[_{\overline{D}} \text{ zheben }]$.

Extension Nothing to extend.

Postulation Nothing to postulate.



Finished! Since the input stream is empty and all the licensing requirements of the structure we have built are satisfied, this parse comes to a successful conclusion.

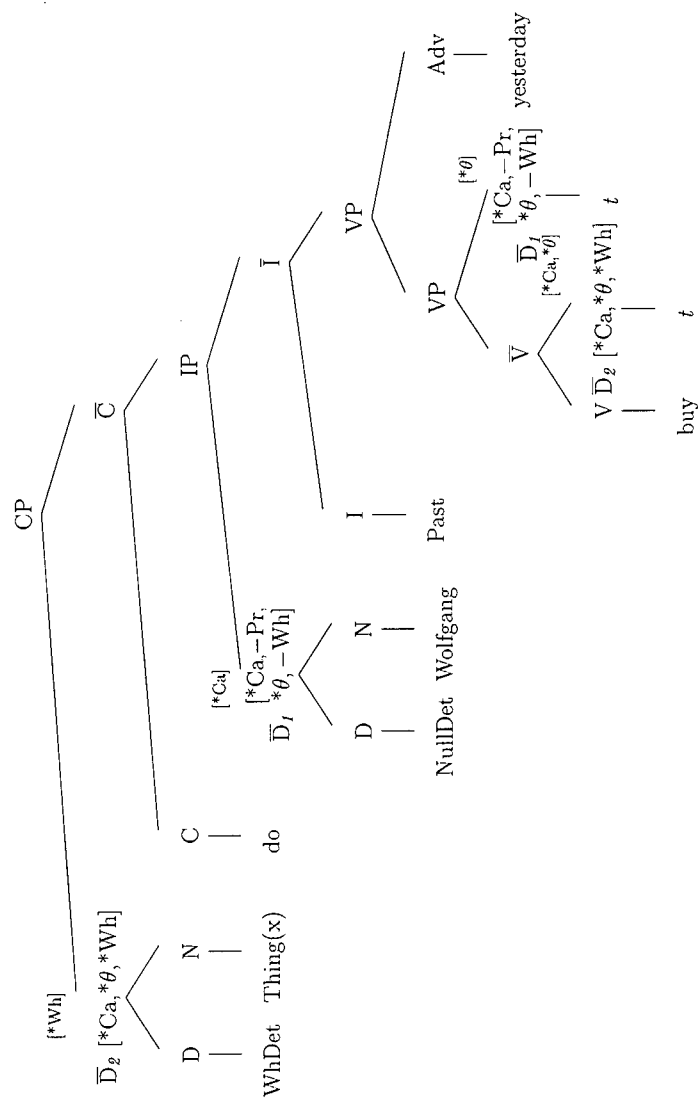
6.3 Impossible movements

Having established how the parser recovers a variety of leftward movements, I must now address why the parser cannot recover a rightward movement configuration.

Suppose the parser were presented with (204b) rather than (204a), where *what* in (204b) does not bear the heavy stress which would license it as an in-situ *wh* phrase with echo question interpretation.

- (204) a. What did Wolfgang buy yesterday?
b. Did Wolfgang buy yesterday what?

Both have the following hierarchical structure:



Yet the instructions given to the parser in the second sentence are not sufficient to give rise to the above structure. At the point where the parser considers the position where a gap must be postulated, the overt *wh* phrase has not yet been encountered. Thus, the information that this is a question arrives too late for the

We have already worked through an example like (204a) in section 6.1.1. Let us now work through (204b), the unparsable rightward movement example, in detail.

- ## Rounds 1–5

CP

[+Wh]

\bar{C}

C

do

IP

[*Ca]

\bar{I}

I

Past

VP

VP

Adv

yesterday

\bar{V}

V

buy

\bar{D}_I

NullDet

Wolfgang

\bar{D}_I

[*Ca, -Pr, *θ, -Wh]

D

N

a car

\bar{D}_I

[*Ca, -Pr, *θ, -Wh]

t

Round 6

Reception Read **WhDet** from the input.

Projection Project $[\overline{D} \text{ WhDet}]$ with licensing requirements $[+Ca, +Wh]$, and functionally selecting a constituent of category N.

Attachment Since an in-situ interpretation is not licensed for this constituent, the parser attaches it into the specifier of CP.

Extension The parser considers as targets for extension gaps in the constituent which was just attached. The constituent which was just attached is $[\overline{D} \text{ WhDet}]$. There is nothing available to extend into the functionally selected position within this constituent.

Postulation Nothing to postulate.



Round 7

Reception Read **Thing(x)** from the input.

Projection Project $[_N \text{ Thing}(x)]$.

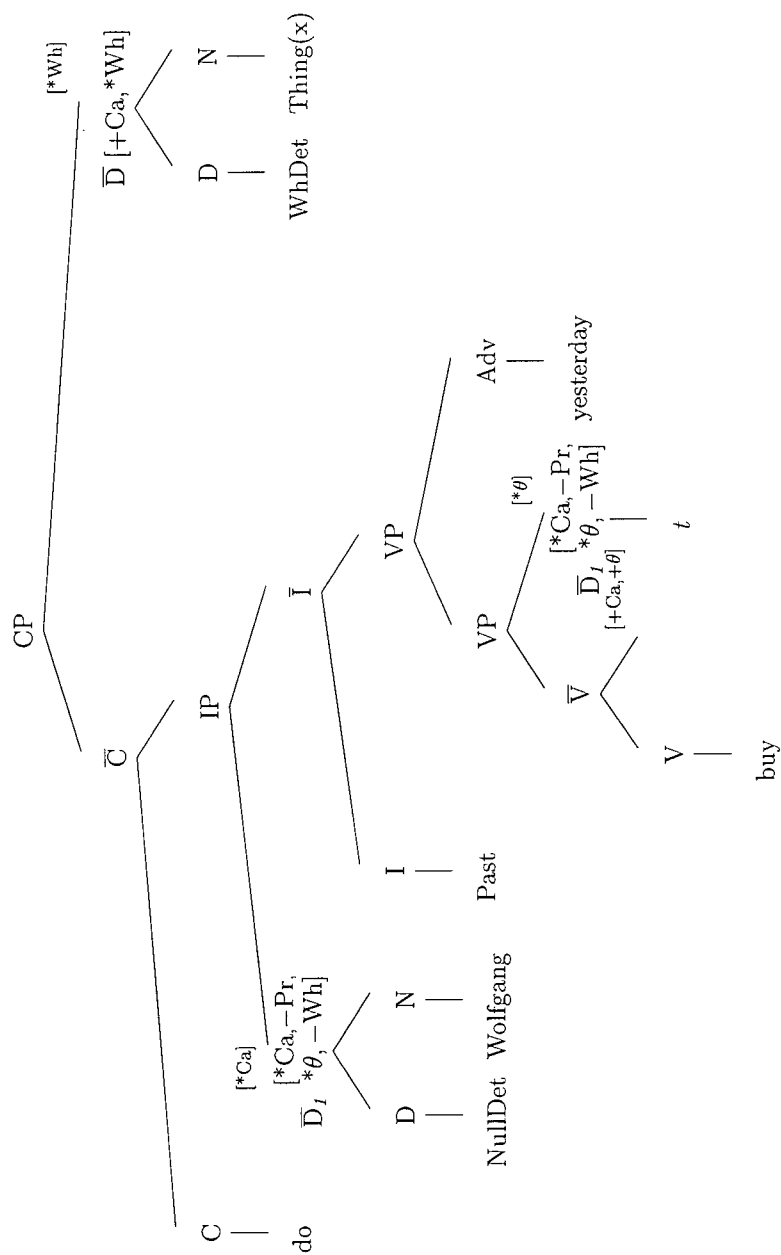
Attachment Attach as a complement of $[_D \text{ WhDet}]$.

Extension Nothing to extend.

Postulation Nothing to postulate.

The resulting tree is shown on the next page.

Finished! Since the input stream is empty, the structure shown is the output of the parser. However, since there are unsatisfied licensing requirements, this structure will be ruled out by the grammar.



When the *wh* phrase is encountered, it can be attached into the specifier of CP position, but it cannot be linked to the empty object position. Because this operator does not bind any variables, it is illicit. Because the Case and θ marking

assigned to the object position of the verb have not been discharged, the VP is illicit. The whole structure is rule out as syntactically ill-formed.

6.4 Unexplained phenomena

At this juncture it is important to make a clear distinction between those phenomena which I seek to explain and those which my approach does not address. I propose an account of why *overt* movement cannot be rightward but must instead be leftward. The parsing algorithm I present offers no explanation of why a language might choose a particular mechanism to express a long-distance relationship. In other words, the algorithm does not address why one language may choose overt movement and while another language utilizes covert movement, or even why one language may choose to use a movement relationship when some other language eschews the use of movement altogether when expressing the same relationship.

For example, consider that French permits a *wh* phrase to *optionally* remain in-situ in matrix clauses and still receive a (non-echo) question interpretation. English does not sanction such an interpretation, and requires that *wh* phrases move overtly. While I do not have any insights to offer as to why French grammar admits an optional movement, this is not a counterexample to my thesis. My thesis is that *overt* movements must be leftward.

In the French case the question word is in-situ (i.e. it is presented to the parser in-situ), and so it can be moved just as any in-situ operator, via a chain-creating process (which builds the chain from foot to head). Non-overt movements, whether of overt or covert material, results in foot-to-head chain-building.

As a further example, since my approach does not address *why* movement is overt or covert, I have nothing to say about cases of multiple *wh* interpretations in

English. These are cases as shown below

- (207) a. Who remembers what we bought where?
b. Who remembers who bought what?

These cases, which are ambiguous, contain an unmoved *wh* phrase. This unmoved *wh* phrase can be interpreted with respect to either of the moved ones. In other words, possible answers to (a) include

- (208) a. Bill does.
b. Bill remembers what we bought at the supermarket, but Mary remembers what we bought at the toy store.

6.5 Summary

In this chapter I have demonstrated that my algorithm can process a variety of leftward movement constructions, including examples involving overt movement of an overt phrase and movements of empty categories, licensed initially in both operator and argument positions.

The common element in all these structures is that the constituent which is to serve as the head of the movement chain is identified *before* any other position of the chain. The algorithm identifies a chain head as a singleton chain, and not as part of an existing chain. An existing chain is extended only when a gap site has been identified. Since a trace is a silent copy of its filler, the filler must exist in the structure at the time the trace is created.

In a rightward movement configuration the parser would need to identify a trace position of a chain before the head position, or create a trace of the head once the head is encountered, and reinsert the trace into the extraction site. However,

a copy of the filler cannot be made when the filler is first identified, as this would be a filler-driven rather than a gap-driven mechanism. Hence, rightward movement cannot be parsed.

Chapter 7

Consequences

In order for the parsing algorithm presented in chapter 5 to parse an overt movement structure, the filler must be encountered before the filler's extraction site. Therefore the algorithm initially treats an overt DP as a singleton chain, not as an extension of an existing chain. The algorithm also initially treats any non-trace empty category which it postulates as a single chain. In this chapter I explore some of the consequences of the chain building algorithm.

In much of this chapter's discussion it is important to identify in which position an empty nominal was initially postulated and where it is situated in the final analysis. Recall that a labelled bracketing is simply a linear representation of a hierarchical structure. The linear order imposed on the bracketed representation does not reflect a linear ordering in the hierarchical structure. I will adopt the following convention when giving labelled-bracketing representations of parse trees:¹

¹Notice that upward extension of a chain during parsing cannot involve overt movement. Overt movement (like all movement) must position the chain head higher in the tree than any trace in the chain. Overt movement must be leftward. If the chain head moves higher in the tree by overt movement, the chain head would be positioned to the left of where it

- The position in which an empty nominal is postulated is shown as the leftmost element of a chain.
- When a chain is extended, any new chain position is shown to the right of existing chain positions. Since the Proper Binding Condition requires that an antecedent c-command its traces, this implies that:
 - when a chain is extended downwards in the tree, the new position is a trace;
 - when a chain is extended upwards in the tree, the new position is the chain head.

7.1 Post-verbal subject interpretations

Italian is a *pro* drop language, meaning that it allows the subject of tensed clauses to be unexpressed. In such cases, the subject is interpreted as third person singular (he/she/it). While the basic word order in Italian is SVO, Italian allows post-verbal subjects. A possible word order is thus VOS.

Consider what happens in the case of an optionally transitive verb. Since a subject may be phonetically empty, the structure V DP is ambiguous. The DP can be interpreted as either the object of a transitive clause, with an unexpressed subject, or it could be the postverbal subject of an intransitive clause.

Consider the concrete example in (209). This sentence is ambiguous, with both an intransitive (210a) and a transitive (210b) reading. The transitive reading involves a personal *pro*, and the presence of personal *pro* is always discourse conditioned to some extent. In the absence of a context there is no discourse context

currently is, contrary to fact.

which the personal *pro* can make reference to. A priori the intransitive reading should be preferred.

(209) ha chiamato Giovanni
has called Giovanni

(210) a. ha chiamato Giovanni
pro_i called Giovanni_i
"Giovanni called"

b. ha chiamato Giovanni
pro called Giovanni
"He/she/it called Giovanni"

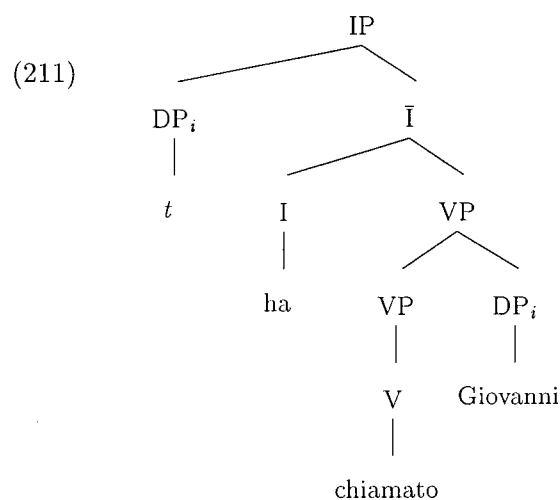
De Vincenzi (De Vincenzi 1991) investigated sentences like (209) and found that the transitive interpretation is preferred to the intransitive one. This finding is all the more surprising as the sentences were presented in the absence of context. This can be interpreted to suggest that there is a strong processing factor which overrides the discourse requirements of personal *pro*.

The sentences which are consistent with the theoretically preferred pro-subject analysis of the first clause are read faster than sentences whose semantics force a revision of the pro-subject analysis. This result is especially surprising because a '*pro*' has a contextually-given antecedent in normal usage. In this experimental setting there was no possible antecedent for the '*pro*'. The prediction is then that if we put the same sentences in context, there should be an even stronger preference for the pro-subject in cases of ambiguity than the preference we found here.

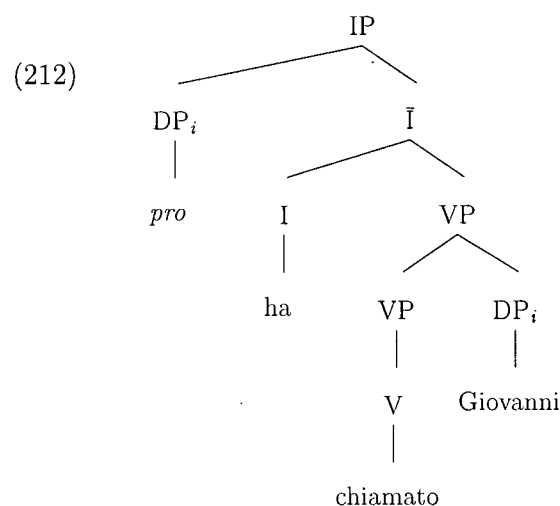
(De Vincenzi 1991, p. 38)

In her account of these cases De Vincenzi adopts an analysis developed by Burzio (1986) and Belletti (1988). Burzio and Belletti both assume that subjects

originate in pre-verbal position, and that post-verbal subjects are lowered to be adjoined to VP, leaving behind a trace, as shown in (211). Null subjects originate, under this view, in the pre-verbal position.



De Vincenzi adopts a slightly different analysis of the postverbal subject cases, shown in (212). Here an expletive *pro* occupies the specifier of IP position, and is in a chain with [DP Giovanni].

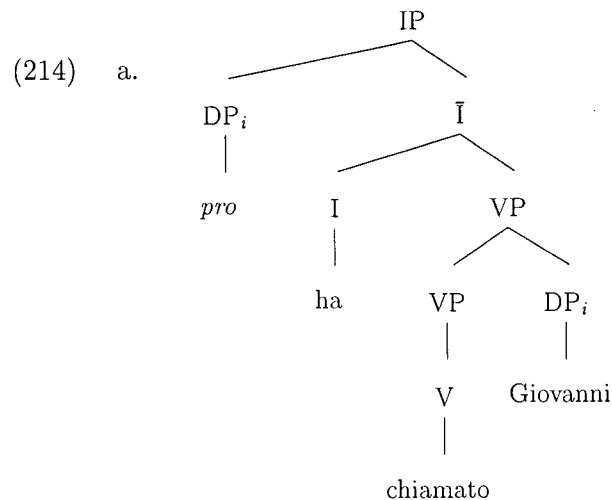


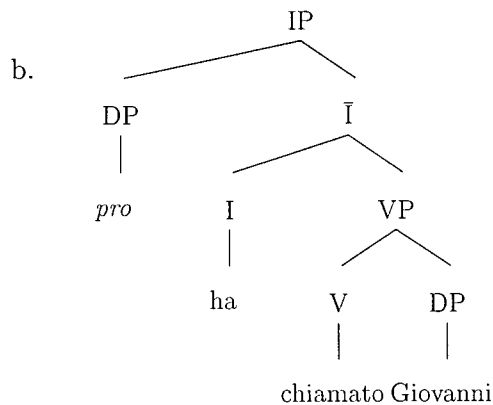
De Vincenzi proposes a parsing-based explanation to account for the observed preference: the Minimal Chain Principle, an extension of Frazier's Active Filler Strategy.

(213) **Minimal Chain Principle (MCP)**

- a. do not postulate potentially unnecessary chain members, but
- b. do not delay postulating required chain members.

Given the Burzio/Belletti-style analysis of these structures, the Minimal Chain Principle correctly predicts the observed preferences. The intransitive interpretation is derived by lowering (a dubious operation which is in violation of the Proper Binding Condition) or by forming a chain consisting of an expletive *pro* and a full DP. It therefore involves one two-membered chain (in addition to the singleton chain of *Giovanni*). This is shown in (214a). The transitive interpretation only involves two singleton chains, since subject *pro* is base-generated in the subject position, as shown in (214b). The transitive interpretation is therefore favoured by the MCP.

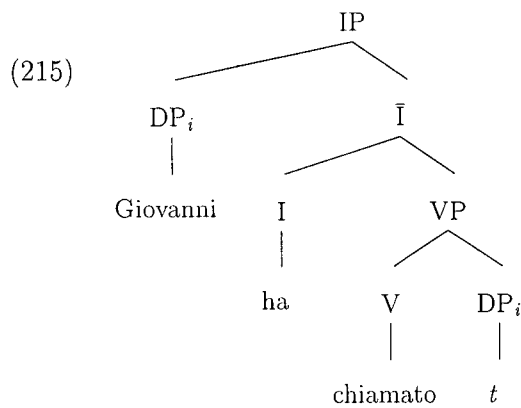




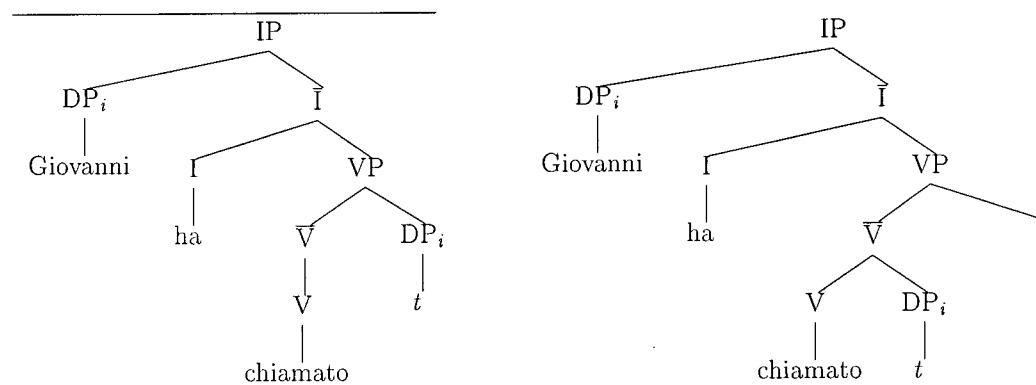
Adopting some form of the VP-internal subject hypothesis, a structural analysis different from that of Burzio and Belletti is motivated. A plausible analysis of a pre-verbal overt subject construction, under a VP-internal subject assumption, is shown in (215).²

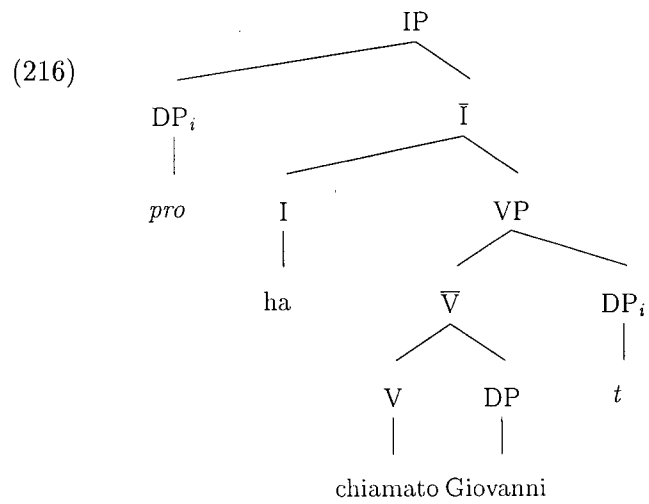
²An additional assumption which needs to be made at this point is that the parser operates under general minimal commitment principles, and treats the basic structural relation during structure building as *dominance* rather than *immediate dominance*. This does not affect the processing of examples already discussed, but is significant for the current discussion. As mentioned in section 2.4.1, underspecification is a general mechanism for dealing with local ambiguities which do not give rise to processing difficulties.

In the structure shown in example 215, the parse tree description shown must be in a final analysis be resolved. The given parse tree description is compatible with two structural analyses. Because of the subcategorization properties of the verb, only the second, with the addition of a null subject, is a possible analysis.

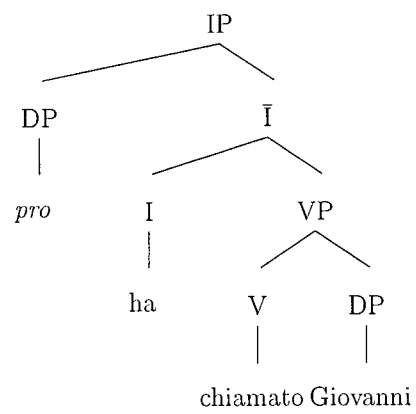


Furthermore, a common analysis of expletive-associate constructions involves raising of the associate to adjoin to the expletive at the syntactic level of Logical Form. In this case the above examples can be argued to have the structures in (216) and (217). Assuming these structures, De Vincenzi's evidence no longer supports her analysis. The intransitive case now has one two-element chain, whereas the transitive case has one two-element chain and one singleton chain. If these structures are indeed motivated, then the MCP no longer makes the correct prediction regarding these structures.

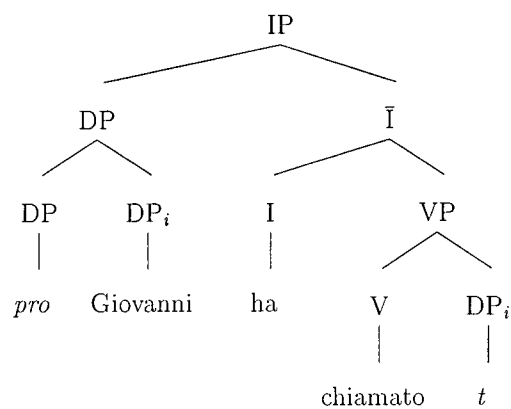




(217) a. S-structure



b. LF



The algorithm of chapter 5 readily accounts for the interpretive preference that De Vincenzi found, given the structures shown in (216) and (217). Let us now work through the processing of (218a), tokenized as in (218b), to show how the

transitive interpretation is given preference over the intransitive interpretation.^{3,4}

- (218) a. Ha chiamato Giovanni.
b. ha chiamato NullDet Giovanni

Round 1

Reception Read **ha** from the input.

Projection Project [_{IP} [_I ha]].

³As noted earlier, in this example I assume that the parser operates under general minimal commitment principles: the structural relation used is not *immediate dominance* but simply *dominance*. Since the implemented parser in fact uses an immediate dominance relation in its structure building, the example here is hand-constructed, showing how a parser following my algorithm would parse the given input. Because the verb is optionally transitive, the parser initially interprets it as being intransitive, but comes to reinterpret it as transitive by inserting an additional branch into the existing structure. This move is permitted if the structural relation is *dominance*, but not if it is *immediate dominance*. Notice that this does not violate determinism but is in fact a means by which a deterministic parser can accommodate certain types of local ambiguities.

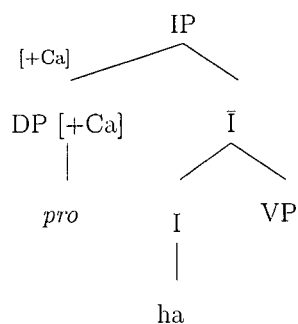
Determinism requires that decisions taken not be abandoned. Processing difficulty is taken to result from the need to reanalyse a structure. Ambiguities which can be accommodated without such reanalysis are those which are predicted to not result in garden-path type effects. This approach is discussed by Marcus et al. (1983), Pritchett (1992) and Gorrell (1995).

⁴Recall from the discussion of feature augmentation in section 5.2.1 that it applies only to the licensing needs of DPs. Feature augmentation does not apply in the case of a choice between two different subcategorization frames for an optionally transitive verb. This is instead handled as mentioned in the previous footnote, by interpreting the structural relation between nodes as *dominance* rather than *immediate dominance*.

Attachment The projected structure becomes the CPPM.

Extension Nothing to extend.

Postulation Postulate an empty expletive in the specifier of I. For the sake of readability, I represent the empty expletive simply as [DP *pro*].



Round 2

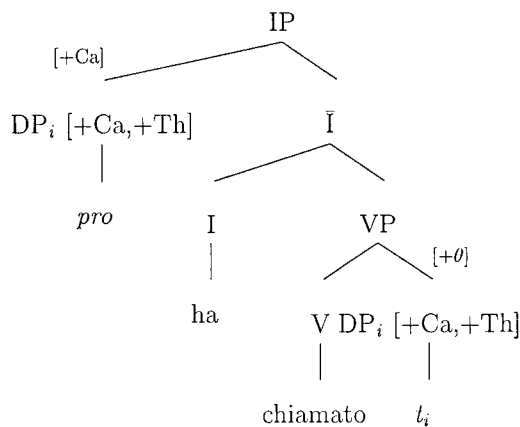
Reception Read **chiamato** from the input.

Projection Project [_{VP} [_V *chiamato*]]. There are two possible subcategorizations, since *chiamato* can be either intransitive or transitive. Assume the minimal configuration, which is the intransitive one.

Attachment Attach VP as complement of I to satisfy functional selection requirement of I.

Extension Extend empty expletive to [+Ca,+ θ], and satisfy the external argument of the VP.

Postulation Nothing to postulate.



Round 3

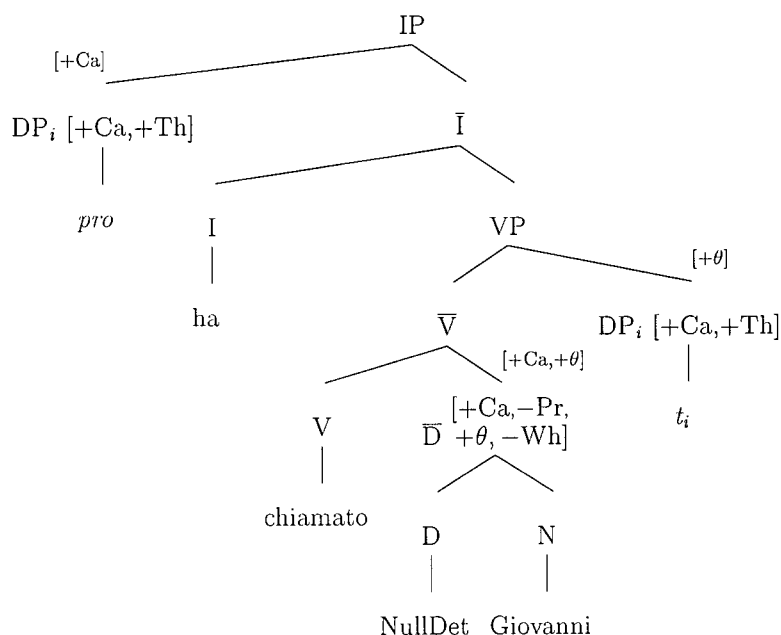
Reception Read **NullDet Giovanni** from the input.

Projection Project $[\overline{D}$ Giovanni] with licensing requirement [+Ca].

Attachment There are two possible ways of incorporating $[\overline{D}$ NullDet Giovanni] into the current structure. The parser can undo the last step, and attach $[\overline{D}$ NullDet Giovanni] into the VP-internal subject position. This should lead to a garden path effect, or at least a filled-gap effect. The parser can also monotonically extend the structure by attaching Giovanni as the object. This involves adding structure to the VP, but not removing any structure which is already there. Because the second alternative does not involve revising structure already built, this alternative is “cost free” and thus preferred.

Extension Nothing to extend.

Postulation Nothing to postulate.



If context requires reanalysis, then some cost will be associated with the reanalysis. Given an optionally transitive verb, it is predicted that transitive reading (with null pronominal subject) will be preferred to the the intransitive reading with an overt postverbal subject.

De Vincenzi assumes that the parser does “commit itself to the choice of an empty category in the absence of disambiguating information” (De Vincenzi 1991, p. 31) and that the parser chooses “the full-‘pro’ ” (De Vincenzi 1991, p. 32) or an argument *pro* bearing both Case and θ features.

I assume, instead, that the parser does not commit fully to a choice of empty category. Following my algorithm, a parser will commit to the existence of an empty category, but will not commit to the full identification of that empty category in the absence of disambiguating information.

As far as chain building behavior is concerned, my algorithm gives preference to singleton chains: all chains are constructed initially as singleton chains. Multiple-

membered chains are built only if the extension of an existing chain helps to satisfy the licensing requirements of an as-yet unfilled position.

7.2 Relative clauses

In this section I consider some interesting properties of restrictive relative clauses. A (restrictive) relative clause is a clause which restricts the reference of the nominal head which it modifies.⁵ I begin by considering the contrast in relative clause structure exhibited in English and Japanese. I then consider how my typology of nominals and my parsing algorithm can accommodate relative clause structures as seen in languages such as Chinese and Hebrew. This latter discussion is speculative in nature, and further exploration of the topic is left for future research.

Relative clauses vary in form cross-linguistically, but they all share certain structural properties. A relative clause contains an operator-variable chain, and is licensed via predication with respect to a DP.⁶ It is beyond the scope of this dissertation to examine the gamut of relativization strategies found cross-linguistically. I focus on the relationship between the linear order of the head nominal and the relative clause itself on the one hand, and whether the relative operator is overt or covert on the other.

The first property to observe is that a relative clause may either precede or follow the nominal which it modifies. In the former case the relative clause is termed *prenominal*, and in the latter *postnominal*. In English relative clauses are postnom-

⁵Unless stated otherwise I use the term “relative clause” to refer to a restrictive relative clause throughout this section.

⁶I put aside internally-headed relative clauses, as found in Lakhota for example, even though Culy (1990) proposes an analysis along the lines sketched in this section.

inal while in Japanese they are prenominal, as shown in (219a,b) respectively.⁷

- (219) a. [Head the child] [Relative Clause who the teacher scolded]
 b. [Relative Clause Mary-ni Bill-o shookai sita] [Head otoko-ga]
 Mary-Dat Bill-Acc introduced man-Nom
 The man who introduced Bill to Mary ...

Examples of how processing proceeds in these cases can be found in sections 6.1.2 and 6.2. Because relative clauses are postnominal in English, disambiguation of the operator in a relative clause happens as soon as the clause is licensed with respect to the nominal. In Japanese relative clauses are prenominal. Because the clause is processed before it is licensed as a relative clause, the extraction site of the operator is processed before the fact that the clause is a relative clause can be determined.

A curious fact about natural language which should be explained is that overt relative pronouns appear only with prenominal relative clauses.⁸

Let us note first that relative pronouns (RPROS) are limited to postnominal RCS; we know of no prenominal relatives in any language which clearly present relative pronouns.

(Keenan 1985, p. 149)

Prenominal relative clauses are most common in head-final languages, such as Japanese. Japanese relative clauses do not provide overt cues to either their

⁷Example (219b) is from (Mazuka 1991).

⁸Following standard terminology, Keenan's use of the term *relative pronoun* refers to an overt relative pronoun. What I call a *relative pro* or a *relative PRO* is generally referred to as an *empty operator*, not a relative pronoun.

subordinate status or their relativizing function. It is expected therefore that a relative clause in Japanese could easily be misinterpreted as a non-relative clause. Indeed, this is the case during on-line processing. Consider examples (220a–b) from Mazuka (1991). (220a) shows that the string of words *Mary-ni Bill-o shokai sita* can be interpreted as an independent clause with a *pro* subject. It is also possible for the utterance to continue as in (220b). In this case *Mary-ni Bill-o shokai sita* must be interpreted as a relative clause.

- (220) a. *pro* Mary-ni Bill-o shokai sita
 Mary-Dat Bill-Acc introduced
 He/she/it introduced Bill to Mary.
- b. [CP *O_i* [IP *t_i* Mary-ni Bill-o shokai sita]] otoko-ga ...
 Mary-Dat Bill-Acc introduced man-N
 The man who introduced Bill to Mary ...

My conjecture is that prenominal relative clauses in a language such as this are only possible in case a personal pronoun can be reanalysed as a relative pronoun. Japanese permits *pro*-drop quite freely. Because the licensing feature structure personal *pro* [+Ca, + θ] can be augmented to that of relative *pro* [+Ca, + θ , +Pr, -Wh], the parser is able to reinterpret personal *pro* as relative *pro* (an empty relative operator) without incurring a perceptual cost.

In English the postulation of a predicative operator in a relative clause is triggered because the relative clause is predicated of the DP which it modifies. A prenominal linearization of relative clauses in a language like English would not present the necessary cue for the parser to postulate a predicative operator. I claim this is the reason that English does not permit prenominal relative clauses. Let us now work through several examples which present what could be prenominal relative clauses in English to see when and why processing breaks down. First note

the possibility of headless relative clauses in English:

(221) John recognized what Mary said as the truth.

(222) John found what Mary was looking for.

(223) Wolfgang met who Mary talked to yesterday.

(224) Wolfgang knows who kissed Mary.

Suppose now that we try to form a prenominal relative clause using an overt relative pronoun, as in (225).

(225) Wolfgang met who Mary talked to yesterday the man.

The string *who Mary talked to yesterday* is interpreted as a headless relative. The structure projected from the string *the man* cannot be incorporated into the structure already built. I contend that (225) is not ungrammatical, simply unprocessable, as indicated in (226). I return to this in section (7.5).

(226) a. \surd Wolfgang met the man who Mary talked to yesterday.

b. $\surd\otimes$ Wolfgang met who Mary talked to yesterday the man.

Consider the case where the relative pronoun is omitted, but an overt complementizer is used, as in (227a-b). The headless relative interpretation is not available in this case.

(227) a. Wolfgang knows that Mary hit yesterday the man.

b. Wolfgang met that Mary hit yesterday the man.

Notice that *that* is lexically ambiguous between a clausal complementizer and a demonstrative determiner. As demonstrated in (228a) both interpretations are possible at the point that *that* is being parsed:

(228) Wolfgang knows that ...

a. [NP song]

b. [IP Montréal has cold winters]

As the parse of (227a) continues the ambiguity is resolved in favour of a clausal complementizer. Processing breaks down when *yesterday* is parsed, because no object DP is available to discharge the Case and θ marking of the verb.

In (227b) only the demonstrative pronoun interpretation is available for *that*, since the verb *meet* cannot take a clausal complement. This interpretation of *that* becomes untenable, and parsing quickly breaks down.

No cue is available in either example which would lead the parser to postulate a relative operator. Hence, neither example is parsable as a relative clause.⁹

(229) a. \checkmark Wolfgang knows the man that Mary hit yesterday.

b. $\checkmark \otimes$ Wolfgang knows that Mary hit yesterday the man.

(230) a. \checkmark Wolfgang met the man that Mary hit yesterday.

b. $\checkmark \otimes$ Wolfgang met that Mary hit yesterday the man.

If also the relative pronoun is omitted, unparsability results as soon as the adverb *yesterday* is encountered in (231a), and as soon as the embedded verb *hit*

⁹It is interesting to note that (227a) is marginally acceptable interpreted as a Heavy NP Shift structure. This interpretation becomes more accessible if the sentence final DP is made heavier:

(1) Wolfgang knows that Mary hit yesterday the very frail-looking man in the wheelchair.

This reinforces the idea that the parser attempts to construct a well-licensed structure as best it can.

is encountered in (231b). There is no way for the parser to build a well-formed structure.

- (231) a. Wolfgang knows Mary hit yesterday the man.
b. Wolfgang met Mary hit yesterday the man.

Let us consider as a final example a case of an infinitival relative clause in English, as in (232).

- (232) Wolfgang and Mary met to prepare tomorrow's show the man.

The infinitival subordinate clause is interpreted as a purpose clause rather than as a relative clause. The structure projected above the string *the man* cannot be incorporated into the structure.

- (233) a. $\sqrt{}$ Wolfgang and Mary met the man to prepare tomorrow's show.
b. $\sqrt{\otimes}$ Wolfgang and Mary met to prepare tomorrow's show the man.

How is a purpose clause licensed as such? It is licensed as a modifier of a VP, much as a relative clause is licensed as a modifier of a DP. Consider the examples in (234)–(237). (234) is ambiguous between the two readings shown in (234a–b). To quote Browning (1987, 108), “In this sentence the chicken may have been brought home as either the guest of honor or the main dish.” (234a) shows the subject relative reading, while (234b) shows the object relative reading.

- (234) John brought the chicken home to eat. (Browning 1987, 108)
a. John brought the chicken home to eat dinner.
b. John brought the chicken home to eat as dinner.

Example (235) is ambiguous in a different way. *to eat* can be interpreted as either a subject relative, as in (235a), or as a purpose clause, as in (235b).

(235) John brought the chicken to eat.

a. John brought the perfect chicken to eat on an occasion such as this.

b. John brought the chicken to eat it.

To highlight the two possible ways of licensing the clause *to eat*, notice that it can only be licensed as a relative clause when adjacent to a DP (236), and only as a purpose clause when adjacent to a VP (237) (=234).

(236) John brought the chicken to eat home.

(237) John brought the chicken home to eat.

In particular, *to eat* cannot be licensed as an extraposed relative clause in (237), even though this is possible in (238).

(238) John brought the chicken home which he intended to eat.

Since there is an ambiguity as to how *to eat* should be licensed in (237), the unavailability of the extraposed relative reading could be due to the parser being garden pathed in resolving the ambiguity towards the purpose clause reading.¹⁰

There are a variety of ways in which a language might indicate to the parser that a clause should be interpreted as a relative clause. If none of these mechanisms is used, then a relative clause can only be parsed as such if the parser can insert a personal *pro* in the structure, to be augmented into a relative *pro* at a later point in the parse.

¹⁰I use the phrase “garden pathed” in a non-standard way here. Following a garden path standardly implies that the parser has resolved a local ambiguity in a manner which is untenable globally, in spite of the availability of a globally viable resolution. Here I use the phrase to refer to the masking of one viable interpretation by the presence of an alternative preferred interpretation.

In English, a relative operator (empty or not) can only be postulated if there is a predication relationship for which there is no overt operator. The juxtaposition of a DP and a clause is the cue for the parser to license a predicative operator.

In a language such as Japanese, *pro* is available. *pro* can be postulated in an empty argument position. When the head of the relative clause is encountered, the licensing feature structure of *pro* may be augmented to include operatorhood in the specifier of CP position. In this way, prenominal relative clauses can be parsed as relative clauses. In fact, if personal pronouns can in general be underspecified, then it should be possible to have prenominal relative clauses in languages which retain overt personal pronouns in the relativized position.

In the remainder of this section I speculate about the structure of pronominal systems in natural language, and indicate how this can explain some of the data concerning relative clauses observed above.

Consider the three categories of pronouns *interrogative*, *relative* and *personal*. In English the form of overt interrogative and overt relative pronouns is the same, while the form of overt personal pronouns is distinct. Thus, *who* can function as either an interrogative or a relative pronoun, but *she* can function only as a personal pronoun. These different pronouns are clearly specified differently with respect to the features [Wh] and [Pr]. *Who* is specified [+Wh], but is underspecified with respect to the [Pr] feature. The value of the [Pr] feature is determined by context, and is therefore not specified in the lexicon. *She* is specified [-Wh,-Pr], as personal pronouns can never function as relative or interrogative operators in English.

Following the idea that overt and covert nominals are alike, covert personal pronouns should also be fully specified with respect to the [Wh] and [Pr] features. Supposing for a moment that English licensed *pro*, it would have to be postulated

with licensing requirements [-Wh,-Pr].

What is the import of the above? Any language which had a split in its pronominal system like English, in which there was overlap between interrogative and relative pronouns, should exhibit the same split in its covert nominals. An example is a language like Italian. Italian does license *pro*, and yet does not permit postnominal relative clauses like those found in Japanese. This is expected if in fact a *pro* in Italian is specified as [-Wh,-Pr], whereas a *pro* in Japanese, underspecified with respect to the [Pr] feature, is specified simply [-Wh].

In order to explore this idea a little bit further, let us consider Chinese and Hebrew. These languages are interesting because they permit personal pronouns to appear in the gap position of relative clauses.

Chinese does not have overt relative pronouns, as we have seen. Chinese relative clauses can be formed with a null relative pronoun. Chinese also permits personal pronouns to appear in the position of gaps in relative clauses.

It very rarely happens that pronominal RCS regularly retain personal pronouns in the NP_{rel} position. The only clear counterexample we know to this claim is Mandarin, as illustrated in (23) below:

- (23) (a) wo da-le (ta) yidum de neige nanhaizi
 I hit-PERF him once REL that boy
 'the boy that I hit once'
- (b) wo bei ta da-le yidum de neige nanhaizi
 I by him hit-PERF once REL that boy
 'the boy by whom I was hit once'

(The pattern of pronoun retention in Mandarin is like that of Hebrew: not normal for NP_{rel} the subject of s_{rel}, optional if NP_{rel} is the direct object of s_{rel}, and generally obligatory elsewhere.)

(Keenan 1985, p. 148-9)

It would appear that Chinese has a different split in its pronominal system than does English: relative pronouns have the same form as personal pronouns. This implies that personal/relative pronouns are underspecified with respect to the [Pr] feature. I have no explanation of why pronoun retention is rare in prenominal relative clauses.

In Hebrew we observe a similar phenomenon: personal pronouns can act as relative pronouns. Demirdache (1991) proposes that personal pronouns appearing in the relativized position in relative clauses are not simply resumptive pronouns (i.e. phonetically non-null traces). She proposes instead that the personal pronouns are in fact relative operators which move covertly to a c-commanding operator position. Demirdache's evidence for this analysis comes from crossover facts, which indicates that movement of an operator is taking place at LF. She proposes that

... a resumptive pronoun behaves like a gap because *there is an \bar{A} -trace in the S-structure position of the pronoun at the level of LF*. In particular, I propose that, at this level, there is a trace in the site of relativization, and that this trace is created by movement of the pronoun to an operator position. This LF-movement of resumptive pronouns, in Hebrew relatives for instance, is exactly on-par with the syntactic movement of *wh*-words or empty operators in English. Thus, the English S-structure shown in (9) corresponds to the Hebrew LF-representation in (10), (the bold *e* stands for an empty Comp, and \emptyset for a null operator).

- (9) a. the girl who lives here.

S-structure: [NP [CP wh_i [C^o e] [IP t_i]]]

LF [NP [CP wh_i [C^o e] [IP t_i]]]

- b. the girl that lives here.

S-structure: [NP [CP \emptyset_i [C^o that] [IP t_i]]]

LF [NP [CP \emptyset_i [C^o that] [IP t_i]]]

- (10) ha-?iŠ_i še pagašti ?oto_i

the-man that met-I him

'the man that I met'

S-structure: [NP [CP [C^o še] [IP ?oto]]]

LF [NP [CP ?oto_i [C^o še] [IP t_i]]]

(Demirdache 1991, p. 17)

I tentatively suggest that in languages such as this personal pronouns are underspecified with respect to the [Pr] feature, and can play the role of either a personal or a relative pronoun, depending on context. In such a case, any interrogative pronoun which a language has is unambiguously marked [+Wh,-Pr].

In this regard, Givón (1975) notes that the use of interrogative pronouns as relative pronouns is becoming possible in Hebrew (emphasis as in original):

Of considerable interest is the diachronic process through which WH-pronouns invade the signalling system of relativization and eventually fuse with the erstwhile relative subordinator. This is occurring now in Hebrew and has presumably occurred in English and Romance. Since this diachronic trend has a certain measure of universality, natural explanations for it should be sought. One which I would like to REJECT at the very start is the one which suggests

THAT THE ORIGIN OF THE SEEMING ETYMOLOGICAL RELATION BETWEEN RELATIVE AND INTERROGATIVE PRONOUNS LIES IN THE FACT THAT RELATIVIZATION (OR PRESUPPOSITION, IN SEMANTIC TERMS) IS ALWAYS INVOLVED IN THE UNDERLYING STRUCTURE OF WH-QUESTIONS. So that interrogative pronouns resemble relative pronouns because they were “borrowed” from relativization. As attractive as this analysis may sound at first, I think it is not supported by facts. To begin with, the etymology in both Hebrew and English support the opposite direction of borrowing. Further, in all languages in which this development has taken place, I believe one should show that at an earlier stage no relative pronouns existed at all. Rather, there existed a relative subordinator and anaphoric resumptive pronouns in relativization, and a separate set of WH-pronouns for questions.

(Givón 1975, p. 14)

This supports the idea that the relativizing function of a pronoun is not the unique function of the pronoun, but is determined by its context of appearance. The fact that Hebrew permits personal pronouns in relative clauses is surprising given that the relative clauses are postnominal and have an initial relative complementizer *še*. Further investigation of these issues is left for future research.

7.3 Non-movements

Recall from the discussion in section 5.1.3 that not all languages employ movement in the formation of questions. Let us now turn to the case of languages which make use of *wh* indefinites, bound by a c-commanding question operator, in forming questions. In languages which make use of this question-formation strategy, *wh* indefinites are not moved to an operator position. Instead, the indefinites introduce a variable which must be bound by a c-commanding operator. If no overt

operator is present, *wh* indefinites are interpreted with respect to a default question operator with matrix scope. The contrast between these two types of languages is shown in (239) and (240), which show the strategy used by English and Chinese respectively.

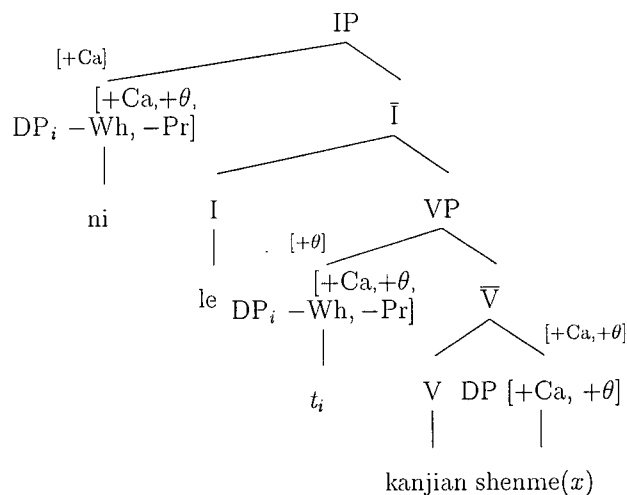
(239) [CP [PP/DP *wh*(*x*)-Op_{*x*}[Q]]_{*k*} [IP... *t_k*...]]

(240) [CP Op_{*x*}[Q] [IP... *wh*(*x*)...]]

The type of variable-binding mechanism used in Chinese is not central to my thesis, since it does not involve movement. To show how my parsing mechanism can handle this type of variable-binding mechanism, let us now work through (241), an example from Chinese.

(241) Ni kanjian le shenme
 you see ASP something
What did you see?

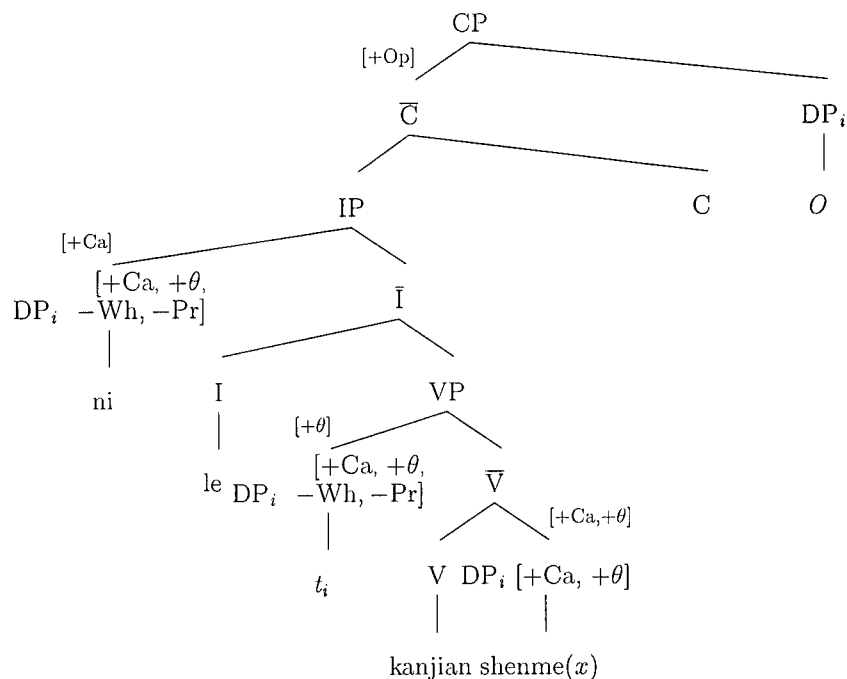
This is similar to French optional *wh* movement case, except that the indefinite does not move. In earlier work Davis and Alphonse (1992) analysed this case exactly as the French optional movement case. While Huang's work proposed that *wh* indefinites move covertly to parallel the English case, recent linguistic research suggests that they introduce variables which are bound by a c-commanding operator. Processing proceeds as in the French example just considered, to this point:



The *wh* indefinite contains a variable which must be bound. If no overt operator binds it, then the parser must insert an empty question operator to yield the default interpretation.¹¹

At the end of a parse, the parser checks the structure for free variables. If there are any, the parser inserts a default question operator which unselectively binds all free variables in the structure.

¹¹As with French, it is natural to assume that all matrix clauses in Chinese are CPs, so that the specifier of CP is available at the point where a default operator must be inserted. Note that an assumption such as this could be made in languages with overt *wh* movement as well, such as English. Such an assumption would confer no particular advantage for the processor, but would also not be detrimental.

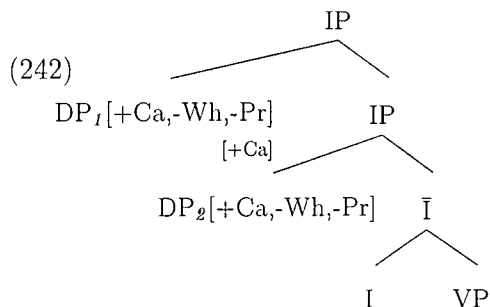


7.4 Left dislocation

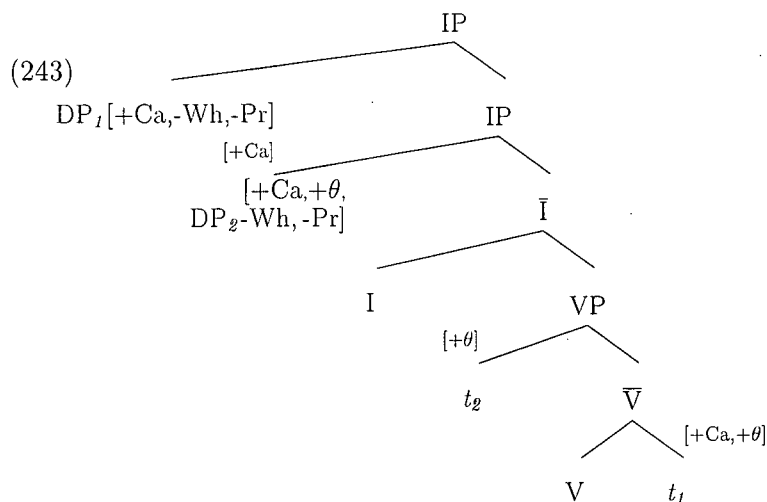
In section 5.2.1 I discussed the initial similarity in topicalization and left dislocation structures. In this section I wish to discuss a prediction which my parsing algorithm makes with respect to left dislocation from an object position.

Consider the case of a fronted DP. As previously discussed, all the parser knows is that an overt DP requires Case. The parser cannot immediately determine whether a DP also requires a θ role, because only arguments require θ roles. A left-dislocated DP, for example, is not in an argument position. A priori the parser cannot determine whether a DP will be associated with an argument position or not.

Consider the point in the processing of a simple transitive clause when this structure has been built.



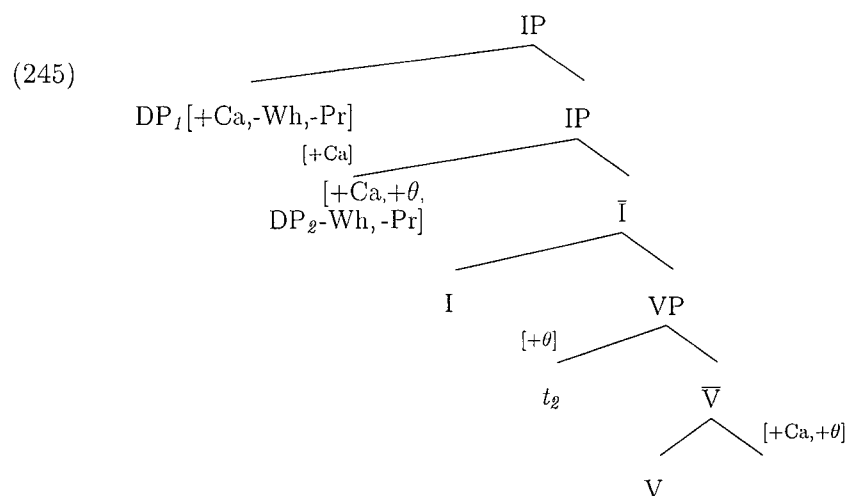
Once the verb is processed the VP-internal subject position, to which a θ -role is assigned, and the object position, to which both Case and θ is assigned, are made available. Since DP_2 is already Case-marked, it cannot be extended to the object position. DP_2 can only be extended to the VP-internal subject position. In order to satisfy its Case-marking requirement, DP_1 must be extended to the object position. So that the verb may discharge its θ role, the licensing feature structure of DP_1 is augmented with a θ requirement.



The processing of left dislocation proceeds in a fashion very similar to topicalization. Consider again (168a), repeated here as (244).

(244) Ardelia, I think that Wolfgang really admires her.

Up to and including the point where the verb is received, processing proceeds in the same way in the two constructions. When the verb is processed the parser should try to build the topicalized structure.

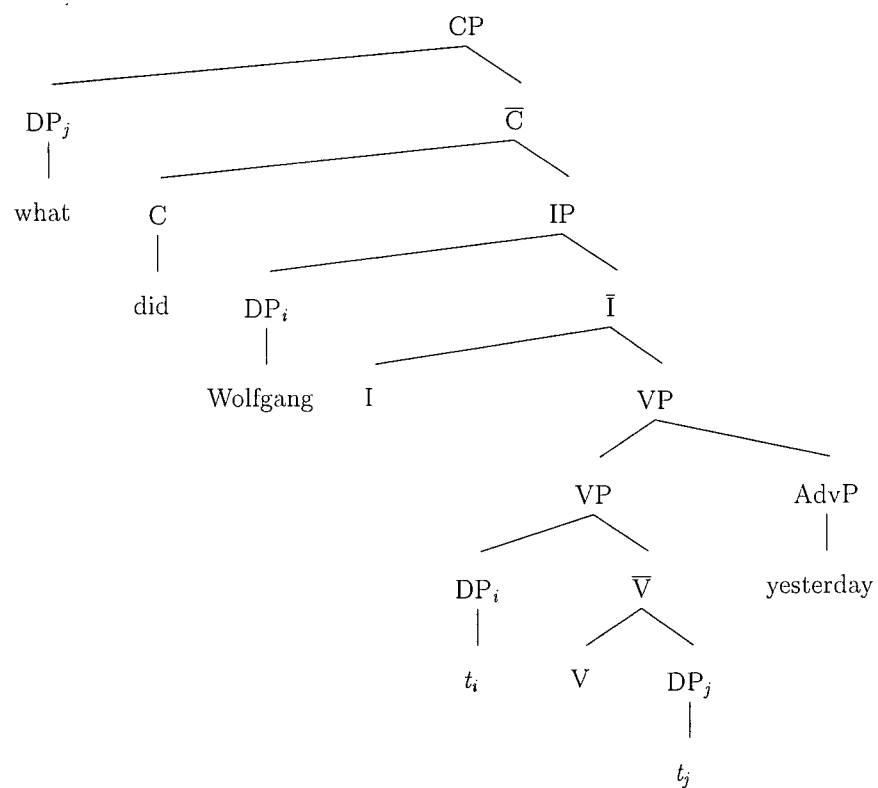


When the overt pronoun is encountered the parser is forced to reanalyse its answer. The misanalysis seen here should result in a filled-gap effect. Filled-gap effects are not very strong – they are not consciously noticeable, but are certainly measurable experimentally. I know of no experiment which tests this prediction; it is an interesting experiment, which is left for future investigation.

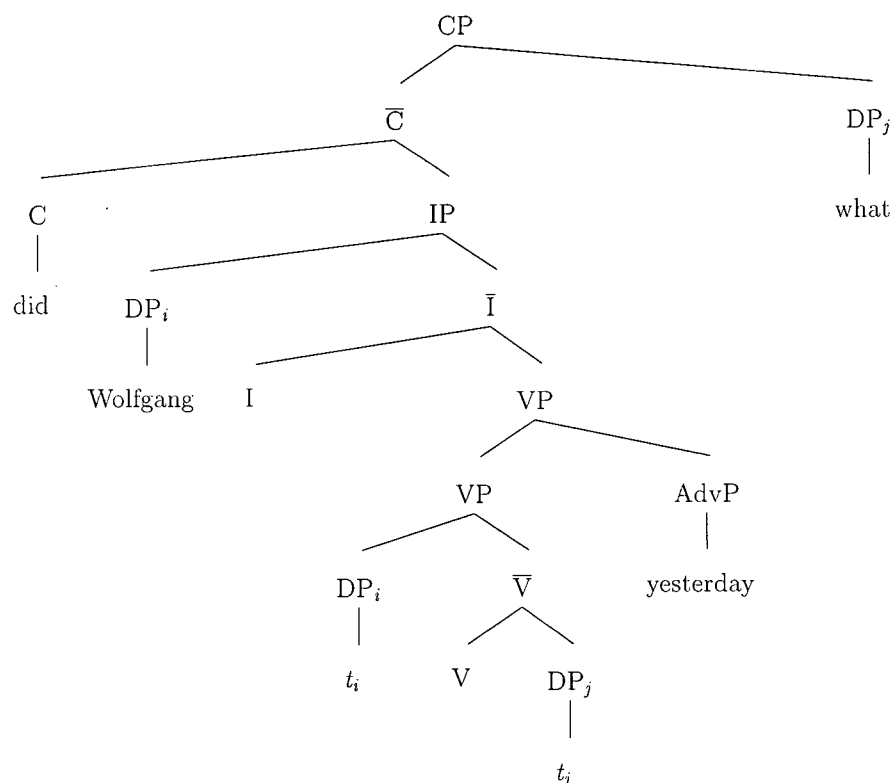
7.5 Garden paths and *culs-de-sac*

Throughout this dissertation I have been concerned with how the linearization of an hierarchical structure affects its processability. I claim that both the structures (246a–b) are grammatical, but that the human parser can only recover the hierarchical structure shown when presented with the linearization of (246a).

(246) a. What did Wolfgang buy yesterday?



b. Did Wolfgang buy yesterday what?



What is the status of (246b)? I claim that it is unprocessable but not ungrammatical. It is not unprocessable in the same sense as a centre-embedded structure, however: it is not the case that the parser exhausts the available supply of some resource. Neither is it unprocessable in the same sense as a garden path sentence: there is no local ambiguity which fools the parser. If this were the case then with practice the structure should become easier to process. Furthermore, it should be possible to embed (246b) in a disambiguating context to bias the parser towards the licit structure. This is not the case.

The problem for the parser, under my analysis, is that the parser only has a wrong path to follow. There is no alternative for the parser at any step of processing,

and the path which it follows leads to a dead end, or a *cul-de-sac*. To the best of my knowledge this type of processing difficulty has not been discussed or named in the processing literature. I propose to call such sentences *culs-de-sac*.

Summarizing, a garden path is a globally unambiguous sentence containing a local ambiguity. Resolving the ambiguity in the wrong way does not allow the globally viable structure to be recovered. A *cul-de-sac* is a sentence which cannot be parsed, but not because of misanalysis of a local ambiguity. Rather, the processing difficulty arises from crucial information becoming available too late in the sentence for the parser to make use of it. The parser perceives no local ambiguity at any point during the parse, and simply gets into a state from which it cannot continue processing.

Chapter 8

Coda

In this final chapter of the dissertation I summarize my findings, revisit the work of Kayne (1994) and Hawkins (1994), and discuss briefly some avenues for future research.

8.1 Summary

This dissertation has explored an interesting linguistic problem from the perspective of on-line parsing. The question I have addressed is why overtly displaced constituents appear to the left of their extraction sites. In short, my answer to this question is that it is only when the parser encounters a filler before its gap that the parser receives the appropriate cues to build a proper parse tree.

In more detail, I claim that the directionality of overt movement derives from the parsing of chains and that the parsing of chains is intimately connected with the processing of empty categories. A problem with parsing empty categories is that they offer no phonetic clue as to their identity. I view this problem as one of ambiguity resolution, and develop a parsing algorithm which initially builds underspecified

licensing specifications. The algorithm incrementally and monotonically further specifies these licensing specifications, fully identifying the empty category in the process.

The proposed parsing algorithm focuses on how empty categories are processed. It is argued that the standard typology of empty categories is not fine-grained enough, and is based on inherent features of nominals. The typology I put forward identifies several more empty categories. This typology is based on contextual features available to the parser during on-line processing, and is structured in such a way as to permit incremental interpretation of empty categories through the use of underspecified licensing feature structures.

The main consequences of this parsing algorithm are that:

- overt movement must be leftward in order for the algorithm to (re)construct a movement chain;
- in a language like Italian, postverbal DPs are preferentially interpreted as objects rather than subjects when they appear after optionally transitive verbs;
- a prenominal relative clause is possible only in a language which permits personal pronouns to be underspecified with respect to the $[\pm\text{Pr}]$ feature, something I hypothesize happens only in languages without overt relative pronouns;
- there is predicted to be a filled-gap effect with structures, like left dislocation in English, for which there is an ambiguity as to whether movement has taken place and in which the licensor precedes the licensee;
- it supports the view that the human language processing mechanism constructs a single underspecified representation, and resolves ambiguities incrementally as the parse progresses.

I have argued that the constraint on overt rightward movement cannot be adequately represented as a constraint in grammar, but that there is a psycholinguistically plausible model for parsing which does offer an explanation. I claim that it is the combination of no lookahead, the gap-driven mechanism for filler-gap processing and the structure building mechanism which explains why overt rightward movement is unparsable as such.

0

8.2 Kayne's Linear Correspondence Axiom (LCA)

Kayne (1994) presents a theory of syntactic structure in which all languages share a common underlying word order, namely Specifier-Head-Complement. This universal word order is derived from Kayne's *Linear Correspondence Axiom (LCA)*. The LCA is a constraint on the mapping from a hierarchical syntactic structure to a linear sequences of words.

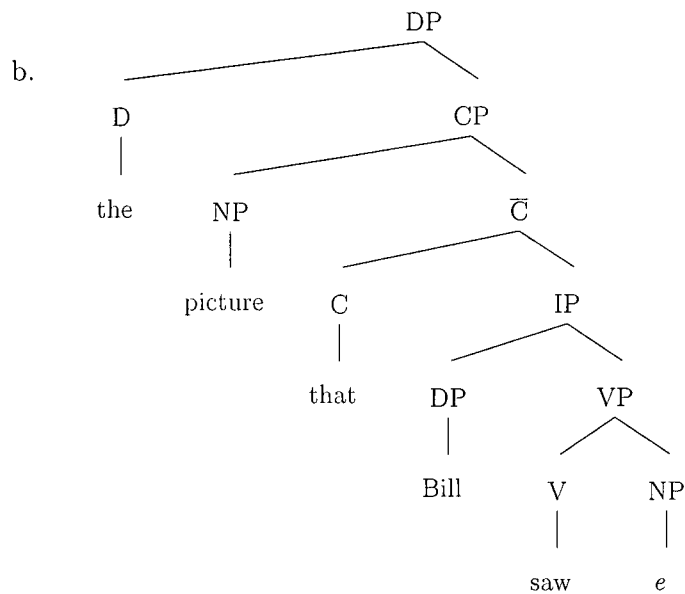
Any word order which deviates from the prescribed universal order must be derived via movement. Under an LCA analysis most clauses contain many more movement chains than has traditionally been assumed.

Consider now some of the implications of this proposal. Kayne argues that the structure of a postnominal relative clause is as shown in (247).¹

(247) (Kayne 1994, p. 94)

a. the picture that Bill saw

¹It is unclear to me what prevents the NP *picture* from appearing as the complement of a D, to yield either *the a picture that Bill saw* or *the picture that Bill saw a*.



A prenominal relative clause has the general structure shown in (248).² Notice that the trace t_I appears before its antecedent [NP picture]. This violates the Proper Binding Condition on traces (but see also the discussion below regarding remnant fronting).

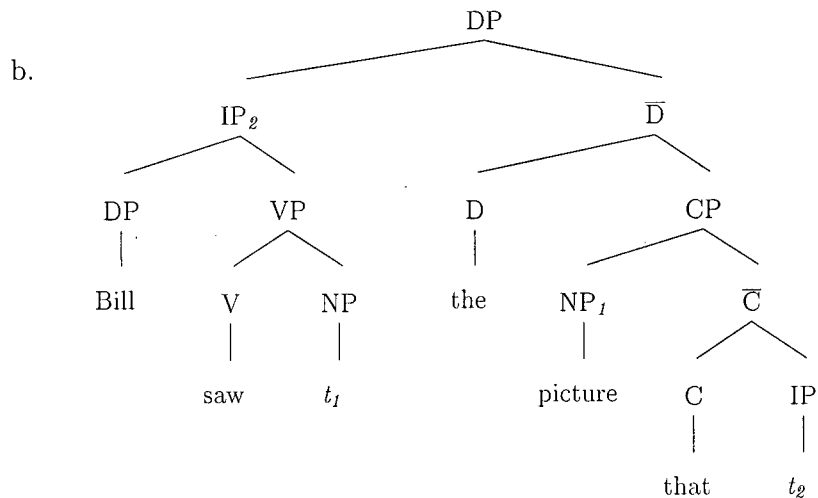
(248) a. Bill saw the picture that

²Kayne states that the structure given reflects that of a prenominal relative clause in Amharic. Kayne further notes that,

... in languages where a relative particle follows the relative proper and precedes N (or D), that relative particle cannot be C^0 , but only some kind of I^0 (whose complement has been raised to its specifier).

(Kayne 1994, p. 157)

These comments apply to a language like Chinese, as the relative particle *de*, commonly analysed as a complementizer, appears before the semantic head.



While the LCA may appear to enforce quite rigid word orders on languages, it does nothing of the sort. The LCA merely supplies a constraint on the mapping from hierarchical structure to linear order. In fact, any word order is achievable under the LCA, given an appropriate hierarchical structure as starting point.

Under Kayne's view hierarchical structure determines word order. Hierarchical structure can deviate from the universal underlying structure only by application of movement. Hence, the theory which governs movement also governs word order in a language.

While it is strictly speaking true that Kayne's framework does not permit rightward movement, the rules of the game have changed and so the question being asked must be reformulated. Rather than asking why rightward movement is illicit, the question must now be why the following word order is ruled out:

(249) saw Mary who?

(250) Mary gave to Bill the book. (where "the book" is topicalized)

Under Kayne's analysis movement of a phrase always targets a specifier position. Many such targets are the specifiers of functional heads. In order to rule out move-

ment which produces the orders in (249) and (250) we need:

- a theory of possible functional heads and their relative positions;
- a theory of what motivates movement to these specifier positions.

Kayne (1994) does not develop either of these theories in enough detail for issues such as this to be addressed. I leave it to future research to explore these questions in Kayne's framework.

8.3 Hawkins' Early Immediate Constituents Principle

Hawkins (1994) presents an account of how processing constraints become grammaticalized over time. Hawkins' theory addresses word order preferences, but his approach does not make predictions in individual cases. Rather, it explains tendencies in word order patterns across languages.

The main explanatory principle of Hawkins' theory is the Early Immediate Constituents (EIC) principle, which is informally characterized in the following quote:

Let me begin with the basic intuition that underlies the major principle of this chapter: Early Immediate Constituents (EIC). I believe that words and constituents occur in the order they do so that syntactic groupings and their immediate constituents (ICs) can be recognized (and produced) as rapidly and efficiently as possible in language performance.

(Hawkins 1994, p. 57)

The EIC Principle has the effect of favoring orders where longer constituents are situated at the periphery of a sentence. In a language like English longer con-

stituents tend to be placed at the right periphery, while in a language like Japanese they tend to be placed at the left periphery.

The EIC does not address why movement is *always* leftward if overt. In English, for example, the EIC would seem to suggest that rightward reordering is possible with long constituents, such as *wh* phrases. Such rightward placement of moved constituents does not occur.

I agree with Hawkins that,

...order is not ultimately a grammatical phenomenon, it is explained by performance. The conventionalized orders of particular grammars are the result of general processing mechanisms that are innate, in conjunction with general efficiency and complexity considerations applied to the structural representations that they produce in language use.

(Hawkins 1994, p. 110)

My goal is much more modest than that of Hawkins, however. My goal has been to argue that one particular linear asymmetry, the leftwardness of overt movement, is best explained as an on-line processing constraint, not a grammatical constraint. I do not make any attempt to tie this processing constraint to grammaticalization. Notice, though, that the leftwardness constraint on overt movement is not variable across languages, and does not have a gradient effect within a language, unlike the phenomena Hawkins considers. A question worthy of further investigation is whether there are different classes of processing effects, those that result in a gradual change in a malleable grammar, and those that are fixed products of the architecture of the language processor and cannot be incorporated into the grammar.

8.4 Future work

In this section I identify some issues which are left for future research.

8.4.1 Linguistic issues

I discuss three topics for future research. The first concerns a study of the structure of pronominal systems across languages. The second and third are more focussed topics. For each of these two I elucidate a prediction of my algorithm, which can be tested in future research.

Pronominal systems

In section 7.2 I speculated about the structure of pronominal systems in natural language. In particular I suggested that *relative* pronouns may pattern with either *interrogative*, or *personal* pronouns, in which case the type of pronoun that relatives pattern with should be underspecified with respect to the $[\pm\text{Pr}]$ feature. An in-depth study of pronominal systems, addressing issues relevant to the on-line processing of overt and covert pronominals, would constitute a valuable contribution.

Remnant fronting and *pro*-drop

Remnant fronting refers to the preposing of a constituent from which material has been extracted. The preposed constituent therefore contains a trace. Such a trace may violate the Proper Binding Condition, which requires that traces must be bound (co-indexed with a c-commanding antecedent). A violation will occur if the remnant is moved to a position higher in the tree than the material extracted from the remnant. I now discuss how the algorithm might accommodate these cases.

VP fronting occurs frequently in German. If some constituent has been extracted from the VP before it is fronted, then the the fronted VP is a remnant. Accepting some form of VP-internal subject hypothesis, any fronted VP is a remnant. The English example in (251) serves to illustrate the problem.

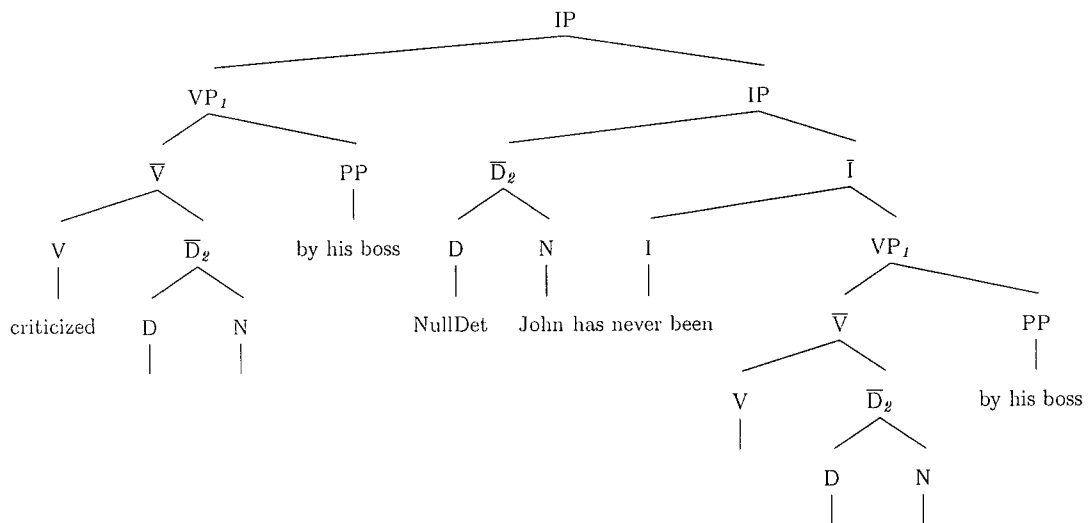
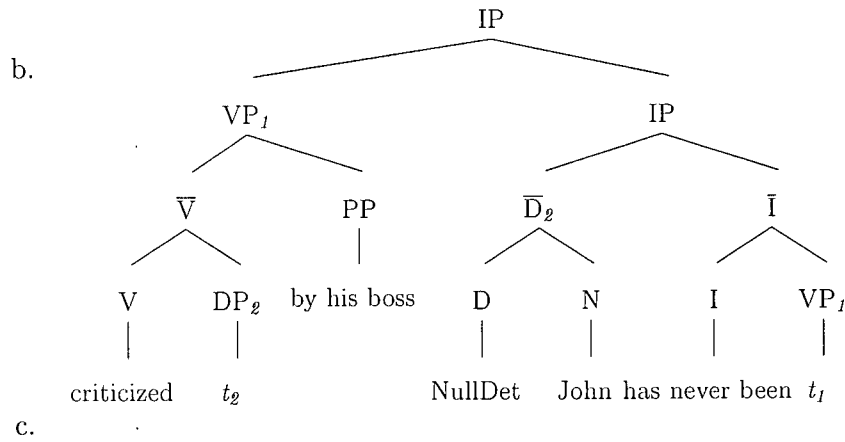
(251) [VP t_i like Mary] [IP I think John_{*i*} does]

The problem for my algorithm is that in a case such as this there is a trace which occurs before its antecedent. This should be unprocessable, since the parser cannot postulate a trace before having identified its antecedent. This type of construction is, however, completely well-formed.

I suggest that a possible solution to this problem is to take seriously the idea that a trace is a silent copy of its antecedent. Adopting such a view, a trace lacks only the phonetic matrix of its antecedent, but maintains the structure of the antecedent. This can be achieved through some form of structure sharing representation of chains, in which only the uppermost chain member has access to phonetic information.

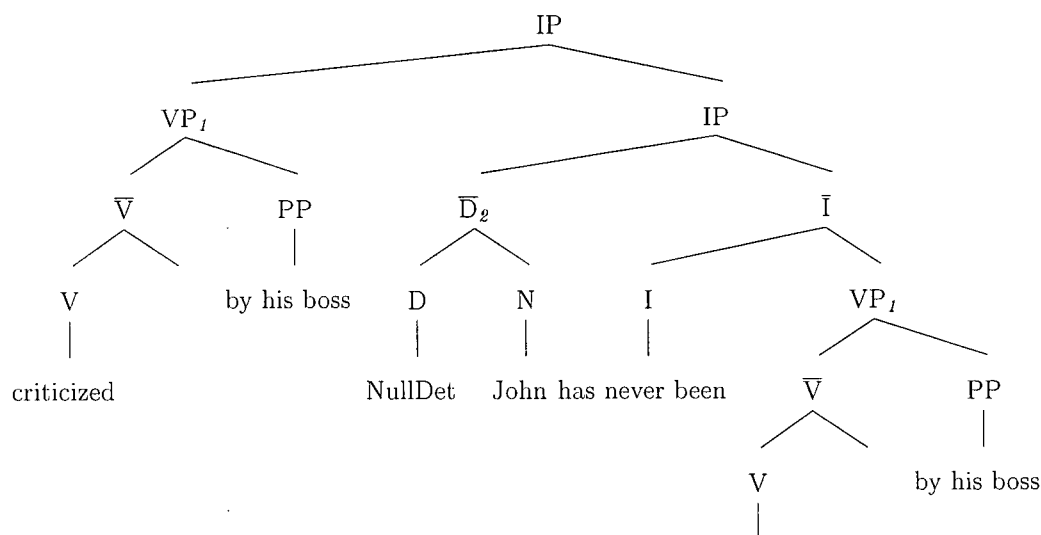
Consider as an example (252a). I propose that the proper representation of this sentence is not (252b) but rather (252c), in which each trace really is a silent copy of its antecedent.

(252) a. [VP criticized t_2 by his boss]_{*I*} John_{*2*} has never been t_1 . (Müller 1998, p. 7)



The proposal that traces are silent copies of their antecedents solves the problem at hand in the following manner. When the parser attaches the trace of VP as a complement of I, the structure built is as shown in (253). When attachment of the trace of the VP takes place, the parser does what it always does when it attaches material into the CPPM: it considers each available attachment point within the newly attached constituent can be the source of a movement (a target for extension).

(253)



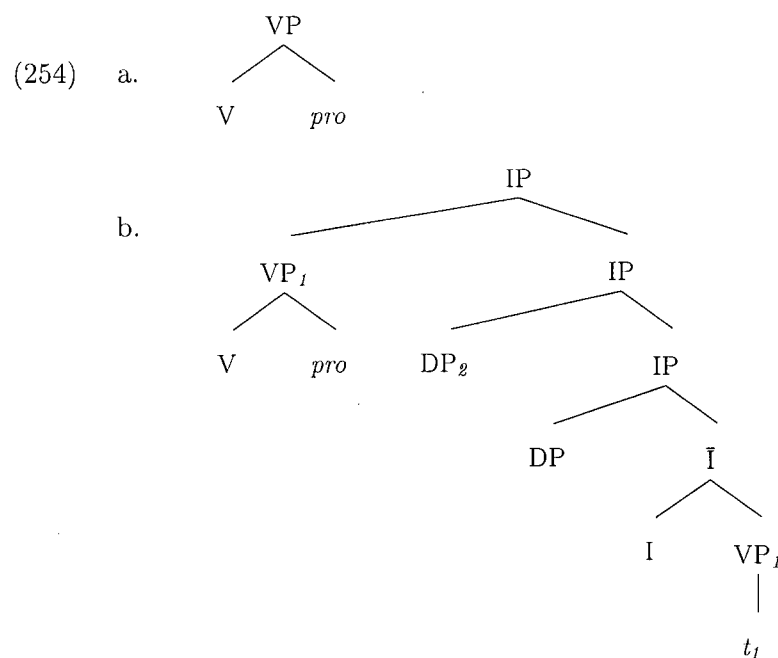
Taking this approach the possibility of remnant fronting follows naturally from the operation of the parser. While it seems appealing, further study of the implications of this move is required.³

³For example why does the parser not postulate *PRO* in the VP internal subject position in English? It would seem that the right environment exists for postulation of *PRO*, since there is a position (the VP-internal subject position) which must be filled. There is no overt material available to fill this position. There is a chain-heading empty category which is licensed in this position in English: *PRO*. It would seem that the parser should postulate *PRO* here. Doing so would be incorrect in this case, since the VP-internal subject position is, in the final analysis, filled by a trace. If revision is required, a garden path effect is expected, yet none is felt.

There are two obvious avenues to explore. One is that the parser does postulate a *PRO*, but that abandonment of a *PRO* hypothesis is costless (there is no garden path) or very cheap (there is a weak garden path, not consciously noticeable). The other possibility is

Recall that many of the movements required by Kayne's approach involve remnant fronting. If these movements are all licit from the parser's and the grammar's perspective, then an argument against Kayne's approach is removed.

The parsing algorithm predicts that in a language which permits *pro*-drop, remnant fronting should lead to a strong garden path. The parser postulates *pro* in the position where it should postulate a trace, as in (254a). The extracted DP cannot be associated with its extraction site, and an ill formed structure results, as in (254b). DP_2 , a scrambled object DP, does not receive Case marking. I ignore here the status of the subject DP.



that the parser for some reason does not postulate *PRO* in this environment. For instance, it might be that postulation of *PRO* occurs only in the environment of an untensed I node. The implications of either choice must be explored.

Rightward positioning and *pro*-drop

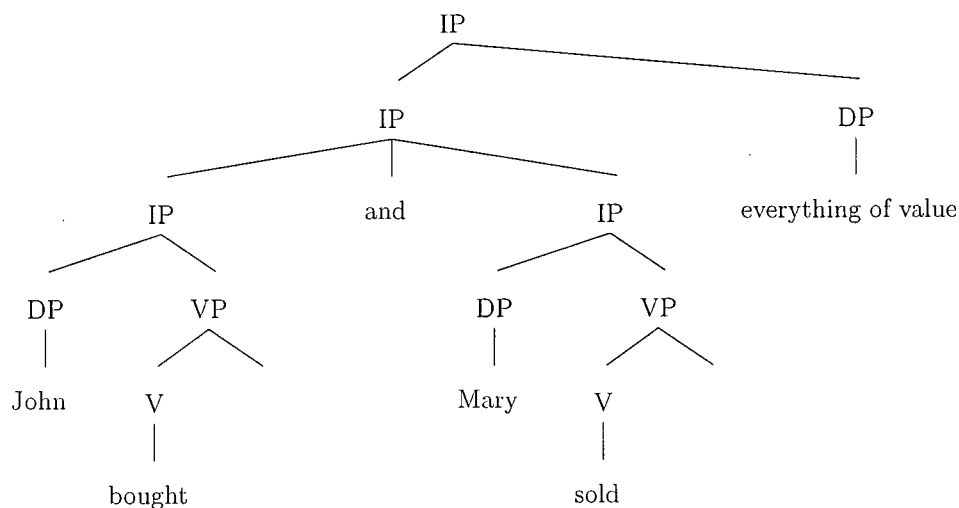
In a language which licenses *pro* in an argument position, the parser will postulate *pro* as a last resort in order to discharge the licensing requirements of the position. In other words, as long as there is no overt material which can fill the position, and there is no constituent in the structure which can be construed as having been moved from this position, the parser will postulate *pro*.

Heavy NP Shift (HNPS) and Right Node Raising (RNR) are two constructions which in English position arguments to the right of their canonical positions. As I argued in chapter 3, I assume that neither of these constructions involves movement. I do not address what the proper analysis of these constructions should be. At this point I wish simply to state a prediction of my parsing algorithm. Recall first what RNR looks like in English.

(255) a. (Rochemont 1992, 393)

John bought and Mary sold everything of value.

b.



The prediction is that a language which permits *pro*-drop should not have

constructions, like English HNPS or RNR, which positions arguments to the right of their canonical positions.

8.4.2 Psycholinguistic issues

In section 7.4 I indicated that the parsing algorithm predicts the existence of a filled-gap effect in left dislocations from object position in English. The more general prediction is that those constructions in which the licenser for a position precedes the licensee are candidates for giving rise to filled-gap effects. This prediction should be tested experimentally.

8.4.3 Computational issues

Ambiguity is pervasive in natural language, and many different approaches to dealing with ambiguity exist.

One approach to dealing with ambiguity involves computing and maintaining multiple representations, one for each possible interpretation. However, the difficulty in computing and cost of storing multiple representations militate against such an approach.

Marcus' (1980) determinism approach suggests that only one representation should be computed and stored during syntactic processing. This approach is too restrictive, however, and does not accord with psycholinguistic data which shows that people do not always disambiguate ambiguous utterances but instead allow them to remain vague.

Vagueness can be represented through the use of underspecified representations, and has the advantage of being a compact representation which can be more fully specified as more information becomes available.

Using underspecified representations which can be monotonically augmented as a technique for dealing with ambiguity should be investigated further, not only within syntactic processing of natural language utterances. (van Deemter and Peters 1996) is a collection of articles dealing with underspecification as a means of dealing with semantic ambiguity.

A difficulty in using underspecified structures to represent ambiguity or vagueness lies in constructing a compact but usable representation which support incremental interpretation. Developing such representations, both in general and for specific domains, and developing ways of processing such representations are both important areas to address.

Bibliography

- Abbot, B. (1976). Right node raising as a test for constituenthood, *Linguistic Inquiry* 7(4): 639-642.
- Abney, S. (1987). Licensing and parsing, in J. McDonough and B. Plunkett (eds), *Proceedings of NELS 17*, University of Massachusetts at Amherst, Amherst, Massachusetts.
- Abney, S. and Cole, J. (1986). A government-binding parser, in S. Berman, J.-W. Choe and McDonough (eds), *Proceedings of NELS 16*, University of Massachusetts at Amherst, Amherst, Massachusetts.
- Alshawi, H. and Crouch, R. (1992). Monotonic semantic interpretation, *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics (ACL-92)*.
- Belletti, A. (1988). The case of unaccusatives, *Linguistic Inquiry* 19: 1-34.
- Browning, M. (1987). *Null Operator Constructions*, PhD thesis, MIT.
- Burzio, L. (1986). *Italian Syntax*, Reidel.
- Cheng, L. L. S. (1991). *On the Typology of Wh-Questions*, PhD thesis, MIT.
- Chomsky, N. (1976). Conditions on rules of grammar, *Linguistic Analysis*.

- Chomsky, N. (1981). *Lectures on Government and Binding*, Studies in Generative Grammar 9, Foris Publications, Dordrecht, The Netherlands.
- Chomsky, N. (1982). *Some concepts and consequences of the theory of government and binding*, Linguistic Inquiry Monographs 6, The MIT Press.
- Chomsky, N. (1986). *Knowledge of Language*, Convergence, Praeger Publishers.
- Chomsky, N. (1995). *The minimalist program*, Current studies in linguistics 28, The MIT Press, Cambridge, Massachusetts.
- Chomsky, N. and Lasnik, H. (1992). Principles and parameters theory, in J. Jacobs, A. von Stechow and W. Sternefeld (eds), *Syntax: An International Handbook of Contemporary Research*, Walter de Gruyter, Berlin.
- Cinque, G. (1999). *Adverbs and functional heads: a cross-linguistic perspective*, Oxford Studies in Comparative Syntax, Oxford University Press.
- Crain, S. and Fodor, J. D. (1985). How can grammars help parsers?, in D. Dowty, L. Karttunen and A. Zwicky (eds), *Natural Language Parsing: Psychological, Computational and Theoretical Perspectives*, Cambridge University Press, Cambridge.
- Culicover, P. W. and Rochemont, M. S. (1990). Extraposition and the complement principle, *Linguistic Inquiry*.
- Culy, C. (1990). *The syntax and semantics of internally headed relative clauses*, PhD thesis, Stanford University.

- Davis, H. and Alphonse, C. (1992). Parsing, WH-movement and linear asymmetry, in K. Broderick (ed.), *Proceedings of the North East Linguistic Society 22*, GLSA, University of Massachusetts, Amherst.
- De Vincenzi, M. (1991). *Syntactic parsing strategies in Italian*, Studies in Theoretical Psycholinguistics 12, Kluwer Academic Publishers.
- Demirdache, H. K. (1991). *Resumptive chains in restrictive relatives, appositives and dislocation structures*, PhD thesis, MIT.
- den Dikken, M. (1995). Binding, expletives, and levels, *Linguistic Inquiry* 26(2): 347–354.
- Dobrovie-Sorin, C. (1990). Clitic doubling, *wh*-movement, and quantification in romanian, *Linguistic Inquiry* 3: 351–397.
- Fodor, J. D. (1978). Parsing strategies and constraints on transformations, *Linguistic Inquiry* 9: 427–474.
- Frank, R. (1992). *Syntactic locality and Tree Adjoining Grammar: Grammatical, Acquisition and Processing perspectives*, PhD thesis, University of Pennsylvania.
- Frazier, L. (1985). Syntactic complexity, in D. Dowty, L. Karttunen and A. Zwicky (eds), *Natural Language Parsing: Psychological, Computational and Theoretical Perspectives*, Cambridge University Press, Cambridge.
- Frazier, L. (1987a). Syntactic processing: Evidence from Dutch, *Natural Language and Linguistic Theory* 5: 519–559.

- Frazier, L. (1987b). Theories of sentence processing, in J. L. Garfield (ed.), *Modularity in Knowledge Representation and Natural-Language Understanding*, The MIT Press.
- Frazier, L. and Flores D'Arcais, G. B. (1989). Filler driven parsing: A study of gap filling in Dutch, *Journal of Memory and Language* **28**(3): 331-344.
- Gazdar, G., Klein, E., Pullum, G. and Sag, I. (1985). *Generalized Phrase Structure Grammar*, Basil Blackwell Publisher Ltd., Oxford, UK.
- Gibson, E., Hickock, G. and Schütze, C. T. (1994). Processing empty categories: A parallel approach, *Journal of Psycholinguistic Research* **23**(5): 381-405.
- Givón, T. (1975). On the role of perceptual clues in hebrew relativization, *Afroasiatic Linguistics* **2**(8): 131-147.
- Gorrell, P. (1995). *Syntax and parsing*, Cambridge University Press.
- Grimshaw, J. (1991). Extended projection, Manuscript, Brandeis University.
- Groat, E. M. (1995). English expletives: A minimalist approach, *Linguistic Inquiry* **26**(2): 354-365.
- Grosu, A. (1973). On the status of the so-called right roof constraint, *Language* **49**: 294-311.
- Haegeman, L. M. V. (1991). *Introduction to Government and Binding Theory*, B. Blackwell.
- Hawkins, J. A. (1994). *A performance theory of order and constituency*, Cambridge Studies in Linguistics 73, Cambridge University Press.

- Higginbotham, J. (1980). Pronouns and bound variables, *Linguistic Inquiry* **11**(4): 679–708.
- Huang, C.-T. J. (1982). Move wh in a language without wh movement, *The Linguistic Review* **1**: 369–416.
- Kayne, R. S. (1994). *The Antisymmetry of Syntax*, Linguistic Inquiry Monographs 25, The MIT Press.
- Keenan, E. (1985). Relative clauses, in T. Shopen (ed.), *Language typology and syntactic description*, Vol. II, Cambridge University Press, pp. 141–170.
- Koopman, H. and Sportiche, D. (1982). Variables and the bijection principle, *The Linguistic Review* **2**: 139–160.
- Koopman, H. and Sportiche, D. (1991). The position of subjects, *Lingua* **85**: 211–258.
- Kornai, A. and Pullum, G. K. (1990). The x-bar theory of phrase structure, *Language* **66**(1): 24–50.
- Kuroda, S. Y. (1988). Whether we agree or not, in W. J. Poser (ed.), *Papers from the Second International Conference on Japanese Syntax*, Center for the Study of Language and Information, Stanford University.
- Lasnik, H. (1995). Case and expletives revisited: On greed and other human failings, *Linguistic Inquiry* **26**(4): 615–633.
- Lasnik, H. and Uriagereka, J. (1988). *A Course in GB Syntax: Lectures on Binding and Empty Categories*, The MIT Press.

- Lee, M.-W. (1999). Is there a filled-gap effect in the pre-verbal subject position in embedded questions in english?, Poster presented at *Architectures and Mechanisms for Language Processing*.
- Levine, R. D. (1985). Right-node (non)-raising, *Linguistic Inquiry* **16**(3): 492-497.
- Marcus, M. P. (1980). *A Theory of Syntactic Recognition for Natural Language*, The MIT Press.
- Marcus, M. P., Hindle, D. and Fleck, M. (1983). D-theory: Talking about talking about trees, *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*.
- Marslen-Wilson, W. (1985). Speech shadowing and speech comprehension, *Speech Communication* **4**: 55-73.
- Mazuka, R. (1991). Processing of empty categories in Japanese, *Journal of Psycholinguistic Research* **20**(3): 215-232.
- McDaniel, D., Chiu, B. and Maxfield, T. L. (1995). Parameters for wh-movement types: evidence from child english, *Natural Language and Linguistic Theory* **13**(4): 709-753.
- Müller, G. (1998). *Incomplete Category Fronting*, Studies in Natural Language and Linguistic Theory, Kluwer Academic Publishers.
- Postal, P. M. (1994). Parasitic and pseudoparasitic gaps, *Linguistic Inquiry* **25**(1): 63-117.
- Pritchett, B. (1992). *Grammatical Competence and Parsing Performance*, The University of Chicago Press.

- Rizzi, L. (1986). Null objects in Italian and the theory of *pro*, *Linguistic Inquiry* **17**(3): 501–557.
- Rizzi, L. (1990). *Relativized Minimality*, Linguistic Inquiry Monographs 16, The MIT Press.
- Rochement, M. S. (1992). Bounding rightward \bar{a} -dependencies, in H. Goodluck and M. Rochement (eds), *Island constraints : theory, acquisition, and processing*, Kluwer.
- Rochement, M. S. and Culicover, P. W. (1990). *English Focus Constructions and the Theory of Grammar*, Cambridge Studies in Linguistics, Cambridge University Press.
- Ross, J. R. (1967). *Constraints on variables in syntax*, PhD thesis, MIT. Distributed by the Indiana University Linguistics Club, Bloomington.
- Ross, J. R. (1986). *Infinite Syntax!*, Language and being, ALEX Publishing Corporation. Originally presented as the author's Ph.D. dissertation (MIT, 1967).
- Safir, K. (1984). Multiple variable binding, *Linguistic Inquiry* **15**(4): 603–638.
- Schriefers, H., Friederici, A. D. and Kühn, K. (1995). The processing of locally ambiguous relative clauses in German, *Journal of Memory and Language* **34**: 499–520.
- Siewierska, A. (1988). *Word Order Rules*, Croom Helm linguistics series, Croom Helm.
- Stowe, L. A. (1984). *Models of Gap-Location in the Human Language Processor*, PhD thesis, University of Rochester. Indiana University Linguistics Club.

- Stowe, L. A. (1986). Parsing WH-constructions: evidence for on-line gap location, *Language and Cognitive Processes* **1**(3): 227–245.
- Tsai, W.-T. (1994). *On Economizing the theory of A-bar dependencies*, PhD thesis, MIT.
- van Deemter, K. and Peters, S. (eds) (1996). *Semantic ambiguity and underspecification*, CSLI Publications.
- van Riemsdijk, H. and Williams, E. (1986). *Introduction to the Theory of Grammar*, The MIT Press.
- Williams, E. (1980). Predication, *Linguistic Inquiry* **11**: 203–238.

Index

\overline{X} , *see* X-bar

$*$, 9

\surd , 10

$?$, 9

\otimes , 10

AFS, *see* Active Filler Strategy

agreement, 23

anaphoric, 32, 95

argument, 98

attachment, 49, 51, 125

augmentation, 119, 126, 129, 132

binding, 31, 100

binding conditions, 32

c-command, 31

case

 marking, 99

Case filter, 24

case marking, 24

category

 empty, 95

functional, 18

governing, 32

head (X^0), 30

lexical, 18

non-lexical, 18

chain, 27, 115

 argument (A), 27

 foot of, 27

 head of, 27

 identification, 115

 licensing, 115

 non-argument (\overline{A}), 29

complement, 16

constituent, 12

determinism, 44

 hypothesis, 44

domination, 31

 immediate, 31

Early Immediate Constituents Principle, 247

- ECP, *see* Empty Category Principle
- empty category, *see* category, empty
- Empty Category Principle, 35
- expletive, 97
- expletive-associate raising, 215
- extension, 49, 51, 52, 129
- filled-gap effect, 38, 90, 91
- filler-gap dependency, 40
- filler-gap processing
 - Active Filler Strategy, 91, 133
 - filler-driven, 40, 41
 - first resort, 42, 43, 89, 90
 - gap-driven, 40, 89, 90, 132
 - last resort, 42, 43
 - second resort, 90, 133
- functional
 - category, 18
 - selection, 19
- functional selection, 126
- garden path, 36
- governing category, 32
- government
 - proper, 35
- GPSG, *see* grammar, Generalized Phrase
 - Structure
 - grammar
 - context free, 11, 73
 - Generalized Phrase Structure, 77
 - Universal, 71
- head, 30
- identification
 - of a chain, 115
- immediate dominance (ID) statement,
 - 77
- interrogative pronoun, *see* pronoun, in-
 - terrogative
- Italian, 210
- LCA, *see* Linear Correspondence Ax-
 - iom
- left dislocation, 129, 236
- licensing
 - of a chain, 115
 - parser, 44
 - parsing, 49
 - requirements, 20
 - structure, 49
- Linear Correspondence Axiom, 81, 82,
 - 244

- linear precedence (LP) statement, 77
- lookahead, 47, 90
 - bounded, 47
- MCP, *see* Minimal Chain Principle
- Minimal Chain Principle, 213
- movement, 12, 26, 158, 182
 - wh*, 158
 - argument (A), 27
 - head, 30
 - impossible, 199
- non-expletive, 97
- notational conventions, 9
- operator, 29, 100, 233
- operators
 - wh*, 101
- PBC, *see* Proper Binding Condition
- PFI, *see* Principle of Full Interpretation
- pleonastic, 97
- position
 - \bar{A} , 100
 - argument (A), 29
 - non-argument (\bar{A}), 30
- predication, 126
- priming, 37
- Principle of Full Interpretation, 20, 48
- PRO, 95, 101, 106
 - personal, 106
 - relative, 106
- pro, 95, 99, 105
 - expletive, 212
 - personal, 105, 210, 211, 224, 228
 - relative, 105
- pronominal, 32, 95
- pronoun
 - interrogative, 101, 229
 - personal, 105, 106, 224, 229
 - relative, 101, 105, 106, 172, 182, 229
- Proper Binding Condition, 34
- proper government, 35
- relative clause, 222
 - infinitival, 105
 - object, 105
 - subject, 105
- postnominal, 222
- prenominal, 222
- relative pronoun, *see* pronoun, relative

shadowing, 36

specifier, 16

thematic (θ) role, 20

 marking, 97

topicalization, 129, 179, 236

trace, 27, 32, 41, 52, 141

typology

 binding theory, 32, 95

 clausal complementizer, 103

 nominal chain, 114, 124

UG, *see* grammar, Universal

underspecification, 48, 94, 119–121, 124

variable, 100, 101

VP-internal subject hypothesis, 21, 214

word order

 basic, 12, 13

X-bar (\bar{X}) theory, 14