SYNTACTIC THEORY

A FORMAL INTRODUCTION

Second Edition

Ivan A. Sag
Thomas Wasow
Emily M. Bender

Syntactic Theory

CSLI Lecture Notes
Number 152

Syntactic Theory

A Formal Introduction

Second Edition

Ivan A. Sag
Thomas Wasow
Emily M. Bender

CSLI PUBLICATIONS
Center for the Study of
Language and Information
Stanford, California

Printed in the United States
10 09 08 07 06   2 3 4 5 6

Library of Congress Cataloging-in-Publication Data

Sag, Ivan A., 1949–
Syntactic theory / by Ivan A. Sag, Thomas Wasow, and Emily M. Bender.— 2nd ed.

p. cm. – (CSLI lecture notes ; no. 152)

Includes bibliographical references and index.
ISBN-13 978-1-57586-399-3 (alk. paper)
ISBN-10 1-57586-399-5 (alk. paper)
ISBN-13 978-1-57586-400-6 (pbk. : alk. paper)
ISBN-10 1-57586-400-2 (pbk. : alk. paper)

1. Grammar, Comparative and general—Syntax.
I. Wasow, Thomas. II. Bender, Emily M., 1973– III. Title. IV. Series.

P291.S25 2003
415–dc21   2003010792
CIP

∞ The acid-free paper used in this book meets the minimum requirements
of the American National Standard for Information Sciences—Permanence
of Paper for Printed Library Materials, ANSI Z39.48-1984.

CSLI was founded in 1983 by researchers from Stanford University, SRI International,
and Xerox PARC to further the research and development of integrated theories
of language, information, and computation. CSLI headquarters and
CSLI Publications are located on the campus of Stanford University.

CSLI Publications reports new developments in the study of language,
information, and computation. Please visit our web site at
http://cslipublications.stanford.edu/
for comments on this and other titles, as well as for changes
and corrections by the authors and publisher.

To the memory of my mother, Helen Ceroli (1917-2002) – I.A.S.

To my wife, Judith Wasow – T.W.

To my husband, Vijay Menon – E.M.B.

Contents

Preface

This textbook grew out of our efforts to develop teaching material for the undergraduate-level Introduction to Syntax course at Stanford University. We wanted our course to emphasize the explicit formulation of precise hypotheses that could be tested against empirical data.

It took us several years of writing, revising, and teaching from drafts before the first edition appeared. On the whole, we have been quite pleased with the result, and most of the feedback we have received – including published reviews – has been favorable. Nevertheless, as we and colleagues at other institutions used the first edition in teaching, good ideas for improvements began to accumulate.

We began the process of preparing a second edition expecting it to be a relatively quick and easy task – a few modifications of specific analyses, some global replaces, and fleshing out explanations where students had exhibited confusion. But we rapidly discovered that we had vastly underestimated the job.

The elements of our grammatical system are highly interconnected. Changes to one piece had ramifications in many other places. Hence, as we worked through the changes we wanted to make, we discovered that almost everything needed at least some reformulation. In the end, every chapter of the book (plus the appendices) underwent substantial revision. We believe that the result is a great improvement.

We have taught from drafts of this edition for two years prior to publication, and several colleagues at other institutions have also taught from those drafts. Feedback from them and from our students has been of immense value to us in shaping the present volume. We consider ourselves fortunate to have such talented and demanding students, whose influence shines through our prose on every page.

Among the major changes between the first and second editions are the following:

- Reconceptualizing the formal foundations, with feature structures defined as functions from feature names to values, and the grammar generating tree structures, rather than descriptions of trees ('structural descriptions').
- Eliminating the need for nonbranching phrasal nodes directly dominating lexical nodes, making our trees more compact.
- Generalizing the earlier analysis of determiner-noun agreement to cover subject-verb agreement as well.
- Grouping together the features SPR, COMPS, and MOD as components of the

value of a feature VAL, allowing simplification and generalization of the Valence Principle.

- Simplifying the Binding Theory, through a reformulation of the definition of 'outrank' and the elimination of the ad hoc P-OBJ feature.
- Formalizing lexical rules as feature structures, facilitating the characterization of different types of lexical rule, and simplifying the formulation of particular rules.
- Formulating passive as a derivational (rather than inflectional) lexical rule.
- Analyzing the infinitival *to* as a verb, rather than a complementizer.
- Generalizing the analysis of sentential negation to cover the reaffirmation uses of *too* and *so*.
- Generalizing the treatment of long-distance dependencies to cover *tough*-constructions, using a new feature, STOP-GAP, as an essential part of the analysis of all LDDs.
- Thoroughly revising the analysis of copula absence in African American Vernacular English, on the basis of arguments in favor of a 'silent verb' analysis.
- Adding grammar summaries or summaries of changes to the grammar at the end of most chapters.

We also did some reorganization, particularly in Chapters 2-4, and we reversed the ordering of Chapters 14 and 15. Problems have been moved to the ends of chapters. A number of new problems have been added, a few old ones have been deleted, and most of the remaining ones have undergone some modification. In addition, we have inserted exercises in various places in the text. Exercises are simple questions about, or applications of, what appears at that point in the book; problems, by contrast, are intended to be more challenging, often requiring students to go well beyond what has been presented. Solutions to the exercises (but not the problems) are included at the end of the book. An 'instructor's manual', including further material (both for instructors and for students) that is relevant to this book and the courses that could be taught from it, is available on-line at: http://hpsg.stanford.edu/book/.

As in the first edition, we have tried to strike a balance between linguistic analysis (centered on the development of increasingly broader grammar fragments) and data-oriented problem solving. We have made an effort to situate our theory in a wider intellectual landscape, through discussion of its history and relationship to other theories (Chapter 1 and Appendix B) and through consideration of its psychological plausibility (Chapter 9) and potential applications to language technologies (Chapter 1). The Glossary, which includes both traditional grammatical terminology and the theoretical vocabulary that we have employed, has been expanded.

The first edition has been successfully used in linguistics courses, both at the undergraduate and graduate level. We have also received positive comments from readers in other disciplines related to linguistics, indicating that that edition had some appeal to researchers working in fields such as psychology, cognitive science, philosophy, mathematics, and computer science. We have endeavored to preserve this appeal in our revisions – this text should be accessible to anyone with an interest in rigorous approaches to the analysis of the grammatical structure of natural languages.

Chapters 1 through 8 develop most of the technical machinery used in the rest of the book. This section of the text is self-contained and can be used as the basis of an abbreviated introductory course. Chapter 9 contains a summary of the grammar developed in Chapters 1 through 8, along with some general discussion of language processing by humans and computers.

Chapters 10 through 13 apply the tools developed in the earlier chapters to some well-studied grammatical phenomena. Chapter 14 deals with the basics of the complex topic of long-distance dependencies, and in the process introduces some new analytic devices. Chapter 15 introduces a topic not normally included in theoretical syntax courses, namely (intra-)language variation. Chapter 16 has a special status, introducing a streamlined reformulation of the overall theory in terms of an architecture based on constructions, not rules. Appendix A summarizes the grammar of English developed in Chapters 1–14 (but includes neither the material on dialects discussed in Chapter 15, nor the innovations proposed in Chapter 16).

We have many people to thank for their part in helping us bring this book to completion – through their assistance with the first edition, with the revisions, or with both. First and foremost, we would like to thank three people: Georgia Green (whose painstakingly detailed and helpful comments emerged from her experiences using drafts of the first edition at the University of Illinois at Urbana-Champaign), Bob Carpenter (whose detailed comments on drafts of both editions led to important revisions), and Roussanka Loukanova (whose extensive comments on a near-final draft of the second edition were invaluable, as was her help with the 'squiggly bits' in Chapters 6 and 9). There are a number of other people who gave us detailed comments on various drafts: Luc Baronian, John Beavers, Gertraud Benke, Frank Van Eynde, Dafydd Gibbon, Jeff Good, Tsuneko Nakazawa, Adam Przepiórkowski, and Gregory Stump. Frank, Dafydd, Tsuneko, and Greg also gave us the benefit of their experiences using our text in their own classes. Special thanks are also due to Chuck Fillmore, Dick Hudson, Aravind Joshi, Paul Kay, Dick Oehrle, and Peter Sells for their help with Appendix B. In addition, we want to thank Takao Gunji and Yasunari Harada for the incredible work they put into translating the first edition into Japanese. We learned a lot about what wasn't clear from interacting with them about fine points of the translation.

Others who helped us one way or another are: Farrell Ackerman, Louise Auerhahn, Noah Barish, John Baugh, Renee Blake, Bob Borsley, Ted Briscoe, Ash Brown, Amy Brynolfson, Chris Callison-Burch, Andrew Carnie, Luis Casillas, Myong-hi Chai, Brady Clark, Steve Conley, Ann Copestake, Anne Cortina, Kirstin Cummings, Ben Davidson, Erica Denham, David Denison, Lei Ditch, Colin Drake, Ben Drucker, Penny Eckert, Louis Eisenberg, Dan Flickinger, Itamar Francez, Jesse Froehlich, Ivan Garcia Alvarez, Ryan Ginstrom, Svetlana Godjevac, Mark Goldenson, Lisa Green, Kent Griffin, Scott Guffey, Randy Gullett, Arvel Hernandez, Erhard Hinrichs, Tom Hukari, Jiro Inaba, Florian Jaeger, Sara Jasper, Andreas Kathol, Jongbok Kim, Roger Kim, Sabine Kirchmeier-Andersen, Kazutake Kita, Ewan Klein, Matt Kodoma, Jean-Pierre Koenig, Anubha Kothari, Madhu Krishnan, Yusuke Kubota, Zakiyyah Langford, Alex Lascarides, Chungmin Lee, Hanjung Lee, Bob Levine, Rob Malouf, Gary Marcus, Michael McDaid, Dave McKercher, Laura Michaelis, Brian Milch, Salimah Mussani, Sheba Najmi, Toshiaki Nishihara, Katsuaki Nakanishi, Rachel Nordlinger, Stephan Oepen, Michael Orme, Gerald Penn, Paul Postal, Geoffrey Pullum, Emily Reyna, John Rickford, Susanne Riehemann, Sarah Risken, Jed Rose, Scott Schwenter, Jeramie Scott, Rylan Sekiguchi, Peter Sells, Yael Shrager, Luke Swartz, Ryosuke Takahashi, Stuart Tannock, Shiao Wei Tham, Ida Toivonen, Judith Tonhauser, Louise Vigeant, Rick Warren, Gert Webelhuth, Steve Wechsler, Bill Weigel, and Corinne Yates. We would also like to thank Dikran Karagueuzian, Director of CSLI Publications, for his continued support and patience, as well as Lauri Kanerva, Anubha Kothari, Michael Frank, Christine Sosa, Maureen Burke, and Tony Gee for their help in matters of production.

This book was written at Stanford's Center for the Study of Language and Information – an ideal environment for thought and writing. Thanks to Emma Pease for sustained help with all matters technological. Some of the material in this text is based on research conducted under the auspices of CSLI's Linguistic Grammars Online (LinGO) project. In that connection, we gratefully acknowledge contracts from the Bundesministerium für Bildung, Wissenschaft, Forschung, und Technologie (bmbf), who supported LinGO's participation in the verbmobil project. This material is also based in part upon work supported by the National Science Foundation under Grant No. BCS-0094638. Work on the second edition was completed while Sag was a Fellow at the Center for Advanced Study in the Behavioral Sciences, in which connection we acknowledge the support of a grant (# 2000-5633) to CASBS from The William and Flora Hewlett Foundation.

Finally, we express our deep appreciation to our spouses, Penny Eckert, Judith Wasow, and Vijay Menon, for putting up with us during the long and often tedious revision process.

Note Added at the Second Printing

This book presents an introduction to syntactic theory inspired by the HPSG perspective, but the theory worked out in the book is simplified with respect to HPSG in many ways. We hope that working through the book will put students in a good position to read and understand the current literature in HPSG (as well as appreciate the current literature in other frameworks). For recent proposals in HPSG, see Ginzburg and Sag 2000, Bouma et al. 2001, and the HPSG bibliography at http://www.cl.uni-bremen.de/HPSG-Bib/.

1

Introduction

1.1 Two Conceptions of Grammar

The reader may wonder, why would a college offer courses on grammar – a topic that is usually thought of as part of junior high school curriculum (or even GRAMMAR school curriculum)? Well, the topic of this book is not the same thing that most people probably think of as grammar.

What is taught as grammar in primary and secondary school is what linguists call 'prescriptive grammar'. It consists of admonitions not to use certain forms or constructions that are common in everyday speech. A prescriptive grammar might contain rules like:

> Be sure to never split an infinitive.
>
> Prepositions are bad to end sentences with.

As modern linguists our concerns are very different. We view human language as a natural phenomenon amenable to scientific investigation, rather than something to be regulated by the decrees of authorities. Your seventh grade math teacher might have told you the (apocryphal) story about how the Indiana legislature almost passed a bill establishing the value of π as 3, and everybody in class no doubt laughed at such foolishness. Most linguists regard prescriptive grammar as silly in much the same way: natural phenomena simply cannot be legislated.

Of course, unlike the value of π, the structure of language is a product of human activity, and that can be legislated. And we do not deny the existence of powerful social and economic reasons for learning the grammatical norms of educated people.[1] But how these norms get established and influence the evolution of languages is a (fascinating) question for sociolinguistics and/or historical linguistics, not for syntactic theory. Hence, it is beyond the scope of this book. Similarly, we will not address issues of educational policy, except to say that in dismissing traditional (prescriptive) grammar instruction, we are not denying that attention to linguistic structure in the classroom can turn students into more effective speakers and writers. Indeed, we would welcome more enlightened grammar instruction in the schools. (See Nunberg 1983 and Cameron 1995 for insightful discussion of these issues.) Our concern instead is with language as it is used in everyday communication; and the rules of prescriptive grammar are of little help in describing actual usage.

[1] By the same token, there may well be good economic reasons for standardizing a decimal approximation to π (though 3 is almost certainly far too crude an approximation for most purposes).

So, if modern grammarians don't worry about split infinitives and the like, then what do they study? It turns out that human languages are amazingly complex systems, whose inner workings can be investigated in large part simply by consulting the intuitions of native speakers. We employ this technique throughout this book, using our own intuitions about English as our principal source of data. In keeping with standard linguistic practice, we will use an asterisk to mark an expression that is not well-formed – that is, an expression that doesn't 'sound good' to our ears. Here are some examples from English:

Example 1 The adjectives *unlikely* and *improbable* are virtually synonymous: we talk about unlikely or improbable events or heroes, and we can paraphrase *It is improbable that Lee will be elected* by saying *It is unlikely that Lee will be elected.* This last sentence is synonymous with *Lee is unlikely to be elected.* So why does it sound so strange to say **Lee is improbable to be elected*?

Example 2 The sentences *They saw Pat with Chris* and *They saw Pat and Chris* are near paraphrases. But if you didn't catch the second name, it would be far more natural to ask *Who did they see Pat with?* than it would be to ask **Who did they see Pat and?* Why do these two nearly identical sentences differ with respect to how we can question their parts? Notice, by the way, that the question that sounds well-formed (or 'grammatical' in the linguist's sense) is the one that violates a standard prescriptive rule. The other sentence is so blatantly deviant that prescriptive grammarians would never think to comment on the impossibility of such sentences. Prescriptive rules typically arise because human language use is innovative, leading languages to change. If people never use a particular construction – like the bad example above – there's no point in bothering to make up a prescriptive rule to tell people not to use it.

Example 3 The two sentences *Something disgusting has slept in this bed* and *Something disgusting has happened in this bed* appear on the surface to be grammatically completely parallel. So why is it that the first has a passive counterpart: *This bed has been slept in by something disgusting*, whereas the second doesn't: **This bed has been happened in by something disgusting*?

These are the sorts of questions contemporary grammarians try to answer. The first two will eventually be addressed in this text, but the third will not.[2] The point of introducing them here is to illustrate a fundamental fact that underlies all modern work in theoretical syntax:

> Every normal speaker of any natural language has acquired an immensely rich and systematic body of unconscious knowledge, which can be investigated by consulting speakers' intuitive judgments.

In other words, knowing a language involves mastering an intricate system full of surprising regularities and idiosyncrasies. Languages are objects of considerable complexity, which can be studied scientifically. That is, we can formulate general hypotheses about linguistic structure and test them against the facts of particular languages.

The study of grammar on this conception is a field in which hypothesis-testing is particularly easy: the linguist can simply ask native speakers whether the predictions

---

[2]For extensive discussion of the third question, see Postal 1986.

regarding well-formedness of crucial sentences are correct.[3] The term 'syntax' is often used instead of 'grammar' in technical work in linguistics. While the two terms are sometimes interchangeable, 'grammar' may also be used more broadly to cover all aspects of language structure; 'syntax', on the other hand, refers only to the ways in which words combine into phrases, and phrases into sentences – the form or structure of well-formed expressions.

Linguists divide grammar into 'syntax', 'semantics' (the study of linguistic meaning), 'morphology' (the study of word structure), and 'phonology' (the study of the sound patterns of language). Although these distinctions are conceptually clear, many phenomena in natural languages involve more than one of these components of grammar.

1.2 An Extended Example: Reflexive and Nonreflexive Pronouns

To get a feel for the sort of research syntacticians conduct, consider the following question:[4]

> In which linguistic environments do English speakers normally use reflexive pronouns (i.e. forms like *herself* or *ourselves*), and where does it sound better to use a nonreflexive pronoun (e.g. *her*, *she*, *us*, or *we*)?

To see how to approach an answer to this question, consider, first, some basic examples:

(1) a.*We like us.
b. We like ourselves.
c. She likes her. [where, she $\neq$ her]
d. She likes herself.
e. Nobody likes us.
f.*Leslie likes ourselves.
g.*Ourselves like us.
h.*Ourselves like ourselves.

These examples suggest a generalization along the following lines:

Hypothesis I: *A reflexive pronoun can appear in a sentence only if that sentence also contains a preceding expression that has the same referent (i.e. a preceding* COREFERENTIAL *expression); a nonreflexive pronoun cannot appear in a sentence that contains such an expression.*

[3]This methodology is not without its pitfalls. Judgments of acceptability show considerable variation across speakers. Moreover, they can be heavily influenced by context, both linguistic and nonlinguistic. Since linguists rarely make any serious effort to control for such effects, not all of the data employed in the syntax literature should be accepted without question. On the other hand, many judgments are so unequivocal that they can clearly be relied on. In more delicate cases, many linguists have begun to supplement judgments with data from actual usage, by examining grammatical patterns found in written and spoken corpora. The use of multiple sources and types of evidence is always a good idea in empirical investigations. See Schütze 1996 for a detailed discussion of methodological issues surrounding the use of judgment data in syntactic research.

[4]The presentation in this section owes much to the pedagogy of David Perlmutter; see Perlmutter and Soames (1979: chapters 2 and 3).

The following examples are different from the previous ones in various ways, so they provide a first test of our hypothesis:

(2) a. She voted for her. [she $\neq$ her]
b. She voted for herself.
c. We voted for her.
d.*We voted for herself.
e.*We gave us presents.
f. We gave ourselves presents.
g.*We gave presents to us.
h. We gave presents to ourselves.
i.*We gave us to the cause.
j. We gave ourselves to the cause.
k.*Leslie told us about us.
l. Leslie told us about ourselves.
m.*Leslie told ourselves about us.
n.*Leslie told ourselves about ourselves.

These examples are all predicted by Hypothesis I, lending it some initial plausibility. But here are some counterexamples:

(3) a. We think that Leslie likes us.
b.*We think that Leslie likes ourselves.

According to our hypothesis, our judgments in (3a,b) should be reversed. Intuitively, the difference between these examples and the earlier ones is that the sentences in (3) contain subordinate clauses, whereas (1) and (2) contain only simple sentences.

Exercise 1: Some Other Subordinate Clauses

Throughout the book we have provided exercises designed to allow you to test your understanding of the material being presented. Answers to these exercises can be found beginning on page 543.

It isn't actually the mere presence of the subordinate clauses in (3) that makes the difference. To see why, consider the following, which contain subordinate clauses but are covered by Hypothesis I.

(i) We think that she voted for her. [she $\neq$ her]
(ii) We think that she voted for herself.
(iii)*We think that herself voted for her.
(iv)*We think that herself voted for herself.

A. Explain how Hypothesis I accounts for the data in (i)-(iv).
B. What is it about the subordinate clauses in (3) that makes them different from those in (i)-(iv) with respect to Hypothesis I?

Given our investigation so far, then, we might revise Hypothesis I to the following:

Hypothesis II: *A reflexive pronoun can appear in a clause only if that clause also contains a preceding, coreferential expression; a nonreflexive pronoun cannot appear in any clause that contains such an expression.*

For sentences with only one clause (such as (1)-(2)), Hypothesis II makes the same predictions as Hypothesis I. But it correctly permits (3a) because *we* and *us* are in different clauses, and it rules out (3b) because *we* and *ourselves* are in different clauses.

However, Hypothesis II as stated won't work either:

(4) a. Our friends like us.
b.*Our friends like ourselves.
c. Those pictures of us offended us.
d.*Those pictures of us offended ourselves.
e. We found your letter to us in the trash.
f.*We found your letter to ourselves in the trash.

What's going on here? The acceptable examples of reflexive pronouns have been cases (i) where the reflexive pronoun is functioning as an object of a verb (or the object of a preposition that goes with the verb) and (ii) where the ANTECEDENT – that is, the expression it is coreferential with – is the subject or a preceding object of the same verb. If we think of a verb as denoting some sort of action or state, then the subject and objects (or prepositional objects) normally refer to the participants in that action or state. These are often called the ARGUMENTS of the verb. In the examples in (4), unlike many of the earlier examples, the reflexive pronouns and their antecedents are not arguments of the same verb (or, in other words, they are not COARGUMENTS). For example in (4b), *our* is just part of the subject of the verb *like*, and hence not itself an argument of the verb; rather, it is *our friends* that denotes participants in the liking relation. Similarly, in (4e) the arguments of *found* are *we* and *your letter to us*; *us* is only part of an argument of *found*.

So to account for these differences, we can consider the following:

Hypothesis III: *A reflexive pronoun must be an argument of a verb that has another preceding argument with the same referent. A nonreflexive pronoun cannot appear as an argument of a verb that has a preceding coreferential argument.*

Each of the examples in (4) contains two coreferential expressions (*we, us, our*, or *ourselves*), but none of them contains two coreferential expressions that are arguments of the same verb. Hypothesis III correctly rules out just those sentences in (4) in which the second of the two coreferential expressions is the reflexive pronoun *ourselves*.

Now consider the following cases:

(5) a. Vote for us!
b.*Vote for ourselves!
c.*Vote for you!
d. Vote for yourself!

In (5d), for the first time, we find a well-formed reflexive with no antecedent. If we don't want to append an *ad hoc* codicil to Hypothesis III,[5] we will need to posit a hidden subject (namely, *you*) in imperative sentences.

Similar arguments can be made with respect to the following sentences.

(6) a. We appealed to them_1 to vote for them_2. [$\text{them}_1 \neq \text{them}_2$]
b. We appealed to them to vote for themselves.
c. We appealed to them to vote for us.

(7) a. We appeared to them to vote for them.
b.*We appeared to them to vote for themselves.
c. We appeared to them to vote for ourselves.

In (6), the pronouns indicate that *them* is functioning as the subject of *vote*, but it looks like it is the object of the preposition *to*, not an argument of *vote*. Likewise, in (7), the pronouns suggest that *we* should be analyzed as an argument of *vote*, but its position suggests that it is an argument of *appeared*. So, on the face of it, such examples are problematical for Hypothesis III, unless we posit arguments that are not directly observable. We will return to the analysis of such cases in later chapters.

You can see that things get quite complex quite fast, requiring abstract notions like 'coreference', being 'arguments of the same verb', and 'phantom arguments' that the rules for pronoun type must make reference to. And we've only scratched the surface of this problem. For example, all the versions of the rules we have come up with so far predict that nonreflexive forms of a pronoun should appear only in positions where their reflexive counterparts are impossible. But this is not quite true, as the following examples illustrate:

(8) a. We wrapped the blankets around us.
b. We wrapped the blankets around ourselves.
c. We admired the pictures of us in the album.
d. We admired the pictures of ourselves in the album.

It should be evident by now that formulating precise rules characterizing where English speakers use reflexive pronouns and where they use nonreflexive pronouns will be a difficult task. We will return to this task in Chapter 7. Our reason for discussing it here was to emphasize the following points:

- Normal use of language involves the mastery of an intricate system, which is not directly accessible to conscious reflection.
- Speakers' tacit knowledge of language can be studied by formulating hypotheses and testing their predictions against intuitive judgments of well-formedness.
- The theoretical machinery required for a viable grammatical analysis could be quite abstract.

[5]For example, an extra clause that says: 'unless the sentence is imperative, in which case a second person reflexive is well-formed and a second person nonreflexive pronoun is not.' This would rule out the offending case but not in any illuminating way that would generalize to other cases.

1.3 Remarks on the History of the Study of Grammar

The conception of grammar we've just presented is quite a recent development. Until about 1800, almost all linguistics was primarily prescriptive. Traditional grammar (going back hundreds, even thousands of years, to ancient India and ancient Greece) was developed largely in response to the inevitable changing of language, which is always (even today) seen by most people as its deterioration. Prescriptive grammars have always been attempts to codify the 'correct' way of talking. Hence, they have concentrated on relatively peripheral aspects of language structure. On the other hand, they have also provided many useful concepts for the sort of grammar we'll be doing. For example, our notion of parts of speech, as well as the most familiar examples (such as *noun* and *verb*) come from the ancient Greeks.

A critical turning point in the history of linguistics took place at the end of the eighteenth century. It was discovered at that time that there was a historical connection among most of the languages of Europe, as well as Sanskrit and other languages of India (plus some languages in between).[6] This led to a tremendous flowering of the field of historical linguistics, centered on reconstructing the family tree of the Indo-European languages by comparing the modern languages with each other and with older texts. Most of this effort concerned the systematic correspondences between individual words and the sounds within those words. But syntactic comparison and reconstruction was also initiated during this period.

In the early twentieth century, many linguists, following the lead of the Swiss scholar Ferdinand de Saussure, turned their attention from the historical (or 'diachronic'[7]) study to the 'synchronic'[8] analysis of languages – that is, to the characterization of languages at a given point in time. The attention to synchronic studies encouraged the investigation of languages that had no writing systems, which are much harder to study diachronically since there is no record of their earlier forms.

In the United States, these developments led linguists to pay far more attention to the indigenous languages of the Americas. Beginning with the work of the anthropological linguist Franz Boas, American linguistics for the first half of the twentieth century was very much concerned with the immense diversity of languages. The Indo-European languages, which were the focus of most nineteenth-century linguistic research, constitute only a tiny fraction of the approximately five thousand known languages. In broadening this perspective, American linguists put great stress on developing ways to describe languages that would not forcibly impose the structure of a familiar language (such as Latin or English) on something very different; most, though by no means all, of this work emphasized the differences among languages. Some linguists, notably Edward Sapir and Benjamin Lee Whorf, talked about how language could provide insights into how people think. They tended to emphasize alleged differences among the thought patterns of speakers of different languages. For our purposes, their most important claim is that the structure of language can provide insight into human cognitive processes. This idea has

[6] The discovery is often attributed to Sir William Jones who announced such a relationship in a 1786 address, but others had noted affinities among these languages before him.

[7] From the Greek: *dia* 'across' plus *chronos* 'time'

[8] *syn* 'same, together' plus *chronos*.

wide currency today, and, as we shall see below, it constitutes one of the most interesting motivations for studying syntax.

In the period around World War II, a number of things happened to set the stage for a revolutionary change in the study of syntax. One was that great advances in mathematical logic provided formal tools that seemed well suited for application to studying natural languages. A related development was the invention of the computer. Though early computers were unbelievably slow and expensive by today's standards, some people immediately saw their potential for natural language applications, such as machine translation or voice typewriters.

A third relevant development around mid-century was the decline of behaviorism in the social sciences. Like many other disciplines, linguistics in America at that time was dominated by behaviorist thinking. That is, it was considered unscientific to posit mental entities or states to account for human behaviors; everything was supposed to be described in terms of correlations between stimuli and responses. Abstract models of what might be going on inside people's minds were taboo. Around 1950, some psychologists began to question these methodological restrictions, and to argue that they made it impossible to explain certain kinds of facts. This set the stage for a serious rethinking of the goals and methods of linguistic research.

In the early 1950s, a young man named Noam Chomsky entered the field of linguistics. In the late '50s, he published three things that revolutionized the study of syntax. One was a set of mathematical results, establishing the foundations of what is now called 'formal language theory'. These results have been seminal in theoretical computer science, and they are crucial underpinnings for computational work on natural language. The second was a book called *Syntactic Structures* that presented a new formalism for grammatical description and analyzed a substantial fragment of English in terms of that formalism. The third was a review of B. F. Skinner's (1957) book *Verbal Behavior*. Skinner was one of the most influential psychologists of the time, and an extreme behaviorist. Chomsky's scathing and devastating review marks, in many people's minds, the end of behaviorism's dominance in American social science.

Since about 1960, Chomsky has been the dominant figure in linguistics. As it happens, the 1960s were a period of unprecedented growth in American academia. Most linguistics departments in the United States were established in the period between 1960 and 1980. This helped solidify Chomsky's dominant position.

One of the central tenets of the Chomskyan approach to syntax, known as 'generative grammar', has already been introduced: hypotheses about linguistic structure should be made precise enough to be testable. A second somewhat more controversial one is that the object of study should be the unconscious knowledge underlying ordinary language use. A third fundamental claim of Chomsky's concerns the biological basis of human linguistic abilities. We will return to this claim in the next section.

Within these general guidelines there is room for many different theories of grammar. Since the 1950s, generative grammarians have explored a wide variety of choices of formalism and theoretical vocabulary. We present a brief summary of these in Appendix B, to help situate the approach presented here within a broader intellectual landscape.

1.4 Why Study Syntax?

Students in syntax courses often ask about the point of such classes: why should one study syntax?

Of course, one has to distinguish this question from a closely related one: why DO people study syntax? The answer to that question is perhaps simpler: exploring the structure of language is an intellectually challenging and, for many people, intrinsically fascinating activity. It is like working on a gigantic puzzle – one so large that it could occupy many lifetimes. Thus, as in any scientific discipline, many researchers are simply captivated by the complex mysteries presented by the data themselves – in this case a seemingly endless, diverse array of languages past, present and future.

This reason is, of course, similar to the reason scholars in any scientific field pursue their research: natural curiosity and fascination with some domain of study. Basic research is not typically driven by the possibility of applications. Although looking for results that will be useful in the short term might be the best strategy for someone seeking personal fortune, it wouldn't be the best strategy for a society looking for long-term benefit from the scientific research it supports. Basic scientific investigation has proven over the centuries to have long-term payoffs, even when the applications were not evident at the time the research was carried out. For example, work in logic and the foundations of mathematics in the first decades of the twentieth century laid the theoretical foundations for the development of the digital computer, but the scholars who did this work were not concerned with its possible applications. Likewise, we don't believe there is any need for linguistic research to be justified on the basis of its foreseeable uses. Nonetheless, we will mention three interrelated reasons that one might have for studying the syntax of human languages.

1.4.1 A Window on the Structure of the Mind

One intellectually important rationale for the study of syntax has been offered by Chomsky. In essence, it is that language – and particularly, its grammatical organization – can provide an especially clear window on the structure of the human mind.[9]

Chomsky claims that the most remarkable fact about human language is the discrepancy between its apparent complexity and the ease with which children acquire it. The structure of any natural language is far more complicated than those of artificial languages or of even the most sophisticated mathematical systems. Yet learning computer languages or mathematics requires intensive instruction (and many students still never master them), whereas every normal child learns at least one natural language merely through exposure. This amazing fact cries out for explanation.[10]

Chomsky's proposed explanation is that most of the complexity of languages does not have to be learned, because much of our knowledge of it is innate: we are born knowing about it. That is, our brains are 'hardwired' to learn certain types of languages.

[9] See Katz and Postal 1991 for arguments against the dominant Chomskyan conception of linguistics as essentially concerned with psychological facts.

[10] Chomsky was certainly not the first person to remark on the extraordinary facility with which children learn language, but, by giving it a central place in his work, he has focused considerable attention on it.

More generally, Chomsky has argued that the human mind is highly modular. That is, we have special-purpose 'mental organs' that are designed to do particular sorts of tasks in particular ways. The language organ (which, in Chomsky's view, has several largely autonomous submodules) is of particular interest because language is such a pervasive and unique part of human nature. All people use language, and (he claims) no other species is capable of learning anything much like human language. Hence, in studying the structure of human languages, we are investigating a central aspect of human nature.

This idea has drawn enormous attention not only from linguists but also from people outside linguistics, especially psychologists and philosophers. Scholars in these fields have been highly divided about Chomsky's innateness claims. Many cognitive psychologists see Chomsky's work as a model for how other mental faculties should be studied, while others argue that the mind (or brain) should be regarded as a general-purpose thinking device, without specialized modules. In philosophy, Chomsky provoked much comment by claiming that his work constitutes a modern version of Descartes' doctrine of innate ideas.

Chomsky's innateness thesis and the interdisciplinary dialogue it stimulated were major factors in the birth of the new interdisciplinary field of cognitive science in the 1970s. (An even more important factor was the rapid evolution of computers, with the concomitant growth of artificial intelligence and the idea that the computer could be used as a model of the mind.) Chomsky and his followers have been major contributors to cognitive science in the subsequent decades.

One theoretical consequence of Chomsky's innateness claim is that all languages must share most of their structure. This is because all children learn the languages spoken around them, irrespective of where their ancestors came from. Hence, the innate knowledge that Chomsky claims makes language acquisition possible must be common to all human beings. If this knowledge also determines most aspects of grammatical structure, as Chomsky says it does, then all languages must be essentially alike. This is a very strong universal claim.

In fact, Chomsky often uses the term 'Universal Grammar' to mean the innate endowment that makes language acquisition possible. A great deal of the syntactic research since the late 1960s has been concerned with identifying linguistic universals, especially those that could plausibly be claimed to reflect innate mental structures operative in language acquisition. As we proceed to develop the grammar in this text, we will ask which aspects of our grammar are peculiar to English and which might plausibly be considered universal.

If Chomsky is right about the innateness of the language faculty, it has a number of practical consequences, especially in fields like language instruction and therapy for language disorders. For example, since there is evidence that people's innate ability to learn languages is far more powerful very early in life (specifically, before puberty) than later, it seems most sensible that elementary education should have a heavy emphasis on language, and that foreign language instruction should not be left until secondary school, as it is in most American schools today.

1.4.2 A Window on the Mind's Activity

If you stop and think about it, it's really quite amazing that people succeed in communicating by using language. Language seems to have a number of design properties that get in the way of efficient and accurate communication of the kind that routinely takes place.

First, it is massively ambiguous. Individual words, for example, often have not just one but a number of meanings, as illustrated by the English examples in (9).

(9) a. Leslie used a *pen.* ('a writing implement')
  b. We put the pigs in a *pen.* ('a fenced enclosure')
  c. We need to *pen* the pigs to keep them from getting into the corn. ('to put in a fenced enclosure')
  d. They should *pen* the letter quickly. ('to write')
  e. The judge sent them to the *pen* for a decade. ('a penitentiary')

(10) a. The cheetah will *run* down the hill. ('to move fast')
  b. The president will *run.* ('to be a political candidate')
  c. The car won't *run.* ('to function properly')
  d. This trail should *run* over the hill. ('to lead')
  e. This dye will *run.* ('to dissolve and spread')
  f. This room will *run* $200 or more. ('to cost')
  g. She can *run* an accelerator. ('to operate')
  h. They will *run* the risk. ('to incur')
  i. These stockings will *run.* ('to tear')
  j. There is a *run* in that stocking. ('a tear')
  k. We need another *run* to win. ('a score in baseball')
  l. Fats won with a *run* of 20. ('a sequence of successful shots in a game of pool')

To make matters worse, many sentences are ambiguous not because they contain ambiguous words, but rather because the words they contain can be related to one another in more than one way, as illustrated in (11).

(11) a. Lee saw the student with a telescope.
  b. I forgot how good beer tastes.

(11a) can be interpreted as providing information about which student Lee saw (the one with a telescope) or about what instrument Lee used (the telescope) to see the student. Similarly, (11b) can convey either that the speaker forgot how GOOD beer (as opposed to bad or mediocre beer) tastes, or else that the speaker forgot that beer (in general) tastes good. These differences are often discussed in terms of which element a word like *with* or *good* is modifying (the verb or the noun).

These two types of ambiguity interact to produce a bewildering array of (often comical) ambiguities, like these:

(12) a. Visiting relatives can be boring.
  b. If only Superman would stop flying planes!
  c. That's a new car dealership.
  d. I know you like the back of my hand.

e. An earthquake in Romania moved buildings as far away as Moscow and Rome.
f. The German shepherd turned on its master.
g. I saw that gas can explode.
h. Max is on the phone now.
i. The only thing capable of consuming this food has four legs and flies.
j. I saw her duck.

This is not the end of the worrisome design properties of human language. Many words are used to refer to different things on different occasions of utterance. Pronouns like *them*, *(s)he*, *this*, and *that* pick out different referents almost every time they are used. Even seemingly determinate pronouns like *we* don't pin down exactly which set of people the speaker is referring to (compare *We have two kids/a city council/a lieutenant governor/50 states/oxygen-based life here*). Moreover, although certain proper names like *Sally Ride*, *Sandra Day O'Connor*, or *Condoleezza Rice* might reliably pick out the same person almost every time they are used, most conversations are full of uses of names like *Chris*, *Pat*, *Leslie*, *Sandy*, etc. that vary wildly in their reference, depending on who's talking to whom and what they're talking about.

Add to this the observation that some expressions seem to make reference to 'covert elements' that don't exactly correspond to any one word. So expressions like *in charge* and *afterwards* make reference to missing elements of some kind – bits of the meaning that have to be supplied from context. Otherwise, discourses like the following wouldn't make sense, or would at best be incomplete:

(13) a. I'm creating a committee. Kim – you're in charge. [in charge of what? – the committee]
b. Lights go out at ten. There will be no talking afterwards. [after what? – after ten]

The way something is said can also have a significant effect on the meaning expressed. A rising intonation, for example, on a one word utterance like *Coffee?* would very naturally convey 'Do you want some coffee?' Alternatively, it might be used to convey that 'coffee' is being offered as a tentative answer to some question (say, *What was Columbia's former number-one cash crop?*). Or even, in the right context, the same utterance might be used in seeking confirmation that a given liquid was in fact coffee.

Finally, note that communication using language leaves a great deal unsaid. If I say to you *Can you give me a hand here?* I'm not just requesting information about your abilities, I'm asking you to help me out. This is the unmistakable communicative intent, but it wasn't literally said. Other examples of such inference are similar, but perhaps more subtle. A famous example[11] is the letter of recommendation saying that the candidate in question has outstanding penmanship (and saying nothing more than that!).

Summing all this up, what we have just seen is that the messages conveyed by utterances of sentences are multiply ambiguous, vague, and uncertain. Yet somehow, in spite of this, those of us who know the language are able to use it to transmit messages to one

[11] This example is one of many due to the late H. Paul Grice, the philosopher whose work forms the starting point for much work in linguistics on problems of PRAGMATICS, how people 'read between the lines' in natural conversation; see Grice 1989.

another with considerable precision – far more precision than the language itself would seem to allow. Those readers who have any experience with computer programming or with mathematical logic will appreciate this dilemma instantly. The very idea of designing a programming language or a logical language whose predicates are ambiguous or whose variables are left without assigned values is unthinkable. No computer can process linguistic expressions unless it 'knows' precisely what the expressions mean and what to do with them.

The fact of the matter is that human language-users are able to do something that modern science doesn't understand well enough to replicate via computer. Somehow, people are able to use nonlinguistic information in such a way that they are never even aware of most of the unwanted interpretations of words, phrases, and sentences. Consider again the various senses of the word *pen*. The 'writing implement' sense is more common – that is, more frequent in the language you've been exposed to (unless you're a farmer or a prisoner) – and so there is an inherent bias toward that sense. You can think of this in terms of 'weighting' or 'degrees of activation' of word senses. In a context where farm animals are being discussed, though, the weights shift – the senses more closely associated with the subject matter of the discourse become stronger in this case. As people direct their attention to and through a given dialogue, these sense preferences can fluctuate considerably. The human sense selection capability is incredibly robust, yet we have only minimal understanding of the cognitive mechanisms that are at work. How exactly does context facilitate our ability to locate the correct sense?

In other cases, it's hard to explain disambiguation so easily in terms of affinity to the domain of discourse. Consider the following contrast:

(14) a. They found the book on the table.
b. They found the book on the atom.

The preposition *on* modifies the verb in (14a) and the noun in (14b), yet it seems that nothing short of rather complex reasoning about the relative size of objects would enable someone to choose which meaning (i.e. which modification) made sense. And we do this kind of thing very quickly, as you can see from (15):

(15) After finding the book on the atom, Sandy went into class, confident that there would be no further obstacles to getting that term paper done.

When you finish reading this sentence, you do not need to go back and think about whether to interpret *on* as in (14a) or (14b). The decision about how to construe *on* is made by the time the word *atom* is understood.

When we process language, we integrate encyclopedic knowledge, plausibility information, frequency biases, discourse information, and perhaps more. Although we don't yet know exactly how we do it, it's clear that we do it very quickly and reasonably accurately. Trying to model this integration is probably the most important research task now facing the study of language.

Syntax plays a crucial role in all this. It imposes constraints on how sentences can or cannot be construed. The discourse context may provide a bias for the 'fenced enclosure' sense of *pen*, but it is the syntactic context that determines whether *pen* occurs as a noun or a verb. Syntax is also of particular importance to the development of language-processing models, because it is a domain of knowledge that can be characterized more precisely than some of the other kinds of knowledge that are involved.

When we understand how language processing works, we probably will also understand quite a bit more about how cognitive processes work in general. This in turn will no doubt enable us to develop better ways of teaching language. We should also be better able to help people who have communicative impairments (and more general cognitive disorders). The study of human language-processing is an important sub-area of the study of human cognition, and it is one that can benefit immensely from precise characterization of linguistic knowledge of the sort that syntacticians seek to provide.

1.4.3 Natural Language Technologies

Grammar has more utilitarian applications, as well. One of the most promising areas for applying syntactic research is in the development of useful and robust natural language technologies. What do we mean by 'natural language technologies'? Roughly, what we have in mind is any sort of computer application that involves natural languages[12] in essential ways. These include devices that translate from one language into another (or perhaps more realistically, that provide translation assistance to someone with less than perfect command of a language), that understand spoken language (to varying degrees), that automatically retrieve information from large bodies of text stored on-line, or that help people with certain disabilities to communicate.

There is one application that obviously must incorporate a great deal of grammatical information, namely, grammar checkers for word processing. Most modern word processing systems include a grammar checking facility, along with a spell-checker. These tend to focus on the concerns of prescriptive grammar, which may be appropriate for the sorts of documents they are generally used on, but which often leads to spurious 'corrections'. Moreover, these programs typically depend on superficial pattern-matching for finding likely grammatical errors, rather than employing in-depth grammatical analysis. In short, grammar checkers can benefit from incorporating the results of research in syntax.

Other computer applications in which grammatical knowledge is clearly essential include those in which well-formed natural language output must be generated. For example, reliable software for translating one language into another must incorporate some representation of the grammar of the target language. If it did not, it would either produce ill-formed output, or it would be limited to some fixed repertoire of sentence templates.

Even where usable natural language technologies can be developed that are not informed by grammatical research, it is often the case that they can be made more robust by including a principled syntactic component. For example, there are many potential uses for software to reduce the number of keystrokes needed to input text, including facilitating the use of computers by individuals with motor disabilities or temporary impairments such as carpal tunnel syndrome. It is clear that knowledge of the grammar of English can help in predicting what words are likely to come next at an arbitrary point in a sentence. Software that makes such predictions and offers the user a set of choices for the next word or the remainder of an entire sentence – each of which can be

[12] That is, English, Japanese, Swahili, etc. in contrast to programming languages or the languages of mathematical logic.

inserted with a single keystroke – can be of great value in a wide variety of situations. Word prediction can likewise facilitate the disambiguation of noisy signals in continuous speech recognition and handwriting recognition.

But it's not obvious that all types of natural language technologies need to be sensitive to grammatical information. Say, for example, we were trying to design a system to extract information from an on-line database by typing in English questions (rather than requiring use of a special database query language, as is the case with most existing database systems). Some computer scientists have argued that full grammatical analysis of the queries is not necessary. Instead, they claim, all that is needed is a program that can extract the essential semantic information out of the queries. Many grammatical details don't seem necessary in order to understand the queries, so it has been argued that they can be ignored for the purpose of this application. Even here, however, a strong case can be made for the value of including a syntactic component in the software.

To see why, imagine that we are using a database in a law office, containing information about the firm's past and present cases, including records of witnesses' testimony. Without designing the query system to pay careful attention to certain details of English grammar, there are questions we might want to ask of this database that could be misanalyzed and hence answered incorrectly. For example, consider our old friend, the rule for reflexive and nonreflexive pronouns. Since formal database query languages don't make any such distinction, one might think it wouldn't be necessary for an English interface to do so either. But suppose we asked one of the following questions:

(16) a. Which witnesses testified against defendants who incriminated them?
b. Which witnesses testified against defendants who incriminated themselves?

Obviously, these two questions will have different answers, so an English language 'front end' that didn't incorporate some rules for distinguishing reflexive and nonreflexive pronouns would sometimes give wrong answers.

In fact, it isn't enough to tell reflexive from nonreflexive pronouns: a database system would need to be able to tell different reflexive pronouns apart. The next two sentences, for example, are identical except for the plurality of the reflexive pronouns:

(17) a. List all witnesses for the defendant who represented himself.
b. List all witnesses for the defendant who represented themselves.

Again, the appropriate answers would be different. So a system that didn't pay attention to whether pronouns are singular or plural couldn't be trusted to answer correctly.

Even features of English grammar that seem useless – things that appear to be entirely redundant – are needed for the analysis of some sentences that might well be used in a human-computer interaction. Consider, for example, English subject-verb agreement (a topic we will return to in some detail in Chapters 2–4). Since subjects are marked as singular or plural – *the dog* vs. *the dogs* – marking verbs for the same thing – *barks* vs. *bark* – seems to add nothing. We would have little trouble understanding someone who always left subject agreement off of verbs. In fact, English doesn't even mark past-tense verbs (other than forms of *be*) for subject agreement. But we don't miss agreement in the past tense, because it is semantically redundant. One might conjecture, therefore, that an English database querying system might be able simply to ignore agreement.

However, once again, examples can be constructed in which the agreement marking on the verb is the only indicator of a crucial semantic distinction. This is the case with the following pair:

(18) a. List associates of each witness who speaks Spanish.
b. List associates of each witness who speak Spanish.

In the first sentence, it is the witnesses in question who are the Spanish-speakers; in the second, it is their associates. These will, in general, not lead to the same answer.

Such examples could be multiplied, but these should be enough to make the point: Building truly robust natural language technologies – that is, software that will allow you to interact with your computer in YOUR language, rather than in ITS language – requires careful and detailed analysis of grammatical structure and how it influences meaning. Shortcuts that rely on semantic heuristics, guesses, or simple pattern-matching will inevitably make mistakes.

Of course, this is not to deny the value of practical engineering and statistical approximation. Indeed, the rapid emergence of natural language technology that is taking place in the world today owes at least as much to these as it does to the insights of linguistic research. Our point is rather that in the long run, especially when the tasks to be performed take on more linguistic subtlety and the accuracy of the performance becomes more critical, the need for more subtle linguistic analysis will likewise become more acute.

In short, although most linguists may be motivated primarily by simple intellectual curiosity, the study of grammar has some fairly obvious uses, even in the relatively short term.

1.5 Phenomena Addressed

Over the next fifteen chapters, we develop theoretical apparatus to provide precise syntactic descriptions. We motivate our formal machinery by examining various phenomena in English. We also address the applicability of our theory to other languages, particularly in some of the problems.

The following is a brief overview of the most important phenomena of English that we deal with. We omit many subtleties in this preliminary survey, but this should give readers a rough sense of what is to come.

- Languages are infinite. That is, there is no limit to the length of sentences, and most utterances have never been uttered before.
- There are different types of words – such as nouns, verbs, etc. – which occur in different linguistic environments.
- There are many constraints on word order in English. For example, we would say *Pat writes books*, not **Writes Pat books*, **Books writes Pat*, or **Pat books writes.*
- Some verbs require objects, some disallow them, and some take them optionally. So we get: *Pat devoured the steak*, but not **Pat devoured*; *Pat dined*, but not **Pat dined the steak*; and both *Pat ate the steak*, and *Pat ate.*
- Verbs agree with their subjects, so (in standard English) we wouldn't say **Pat write books* or **Books is interesting.*

- There is also a kind of agreement within noun phrases; for example, *this bird* but not **this birds*; *these birds* but not **these bird*; and *much water* but not **much bird* or **much birds.*
- Some pronouns have a different form depending on whether they are the subject of the verb or the object: *I saw them* vs. **Me saw them* or **I saw they.*
- As was discussed in Section 1.2, reflexive and nonreflexive pronouns have different distributions, based on the location of their antecedent.
- Commands are usually expressed by sentences without subjects, whose verbs show no agreement or tense marking, such as *Be careful!*
- Verbs come in a variety of forms, depending on their tense and on properties of their subject. Nouns usually have two forms: singular and plural. There are also cases of nouns and verbs that are morphologically and semantically related, such as *drive* and *driver.*
- Sentences with transitive verbs typically have counterparts in the passive voice, e.g. *The dog chased the cat* and *The cat was chased by the dog.*
- The word *there* often occurs as the subject of sentences expressing existential statements, as in *There is a unicorn in the garden.*
- The word *it* in sentences like *It is clear that syntax is difficult* does not refer to anything. This sentence is synonymous with *That syntax is difficult is clear*, where the word *it* doesn't even appear.
- Certain combinations of words, known as idioms, have conventional meanings, not straightforwardly inferable from the meanings of the words within them. Idioms vary in their syntactic versatility. Examples of idioms are *keep tabs on* and *take advantage of.*
- Pairs of sentences like *Pat seems to be helpful* and *Pat tries to be helpful*, though superficially similar, are very different in the semantic relationship between the subject and the main verb. This difference is reflected in the syntax in several ways; for example, *seems* but not *tries* can have the existential *there* as a subject: *There seems to be a unicorn in the garden* vs. **There tries to be a unicorn in the garden.*
- There is a similar contrast between the superficially similar verbs *expect* and *persuade*: *We expected several students to be at the talk* and *We persuaded several students to be at the talk* vs. *We expected there to be several students at the talk* but **We persuaded there to be several students at the talk.*
- Auxiliary ('helping') verbs in English (like *can, is, have*, and *do*) have a number of special properties, notably:
  - fixed ordering (*They have been sleeping* vs **They are having slept*)
  - occurring at the beginning of yes-no questions (*Are they sleeping?*)
  - occuring immediately before *not* (*They are not sleeping*)
  - taking the contracted form of *not*, written *n't* (*They aren't sleeping*)
  - occurring before elliptical (missing) verb phrases (*We aren't sleeping, but they are*)

- There is considerable dialectal variation in the English auxiliary system, notably British/American differences in the use of auxiliary *have* (*Have you the time?*) and the existence of a silent version of *is* in African American Vernacular English (*She the teacher*).
- A number of constructions (such as '*wh*-questions') involve pairing a phrase at the beginning of a sentence with a 'gap' – that is, a missing element later in the sentence. For example, in *What are you talking about? what* functions as the object of the preposition *about*, even though it doesn't appear where the object of a preposition normally does.

These are some of the kinds of facts that a complete grammar of English should account for. We want our grammar to be precise and detailed enough to make claims about the structure and meanings of as many types of sentence as possible. We also want these descriptions to be psychologically realistic and computationally tractable. Finally, despite our focus on English, our descriptive vocabulary and formalization should be applicable to all natural languages.

1.6 Summary

In this chapter, we have drawn an important distinction between prescriptive and descriptive grammar. In addition, we provided an illustration of the kind of syntactic puzzles we will focus on later in the text. Finally, we provided an overview of some of the reasons people have found the study of syntax inherently interesting or useful. In the next chapter, we look at some simple formal models that might be proposed for the grammars of natural languages and discuss some of their shortcomings.

1.7 Further Reading

An entertaining (but by no means unbiased) exposition of modern linguistics and its implications is provided by Pinker (1994). A somewhat more scholarly survey with a slightly different focus is presented by Jackendoff (1994). For discussion of prescriptive grammar, see Nunberg 1983, Cameron 1995, and Chapter 12 of Pinker's book (an edited version of which was published in *The New Republic*, January 31, 1994). For an overview of linguistic science in the nineteenth century, see Pedersen 1959. A succinct survey of the history of linguistics is provided by Robins (1967).

Among Chomsky's many writings on the implications of language acquisition for the study of the mind, we would especially recommend Chomsky 1959 and Chomsky 1972; a more recent, but much more difficult work is Chomsky 1986b. There have been few recent attempts at surveying work in (human or machine) sentence processing. Fodor et al. 1974 is a comprehensive review of early psycholinguistic work within the Chomskyan paradigm, but it is now quite dated. Garrett 1990 and Fodor 1995 are more recent, but much more limited in scope. For a readable, linguistically oriented, general introduction to computational linguistics, see Jurafsky and Martin 2000.

1.8 Problems

⚠ *This symbol before a problem indicates that it should not be skipped. The problem either deals with material that is of central importance in the chapter, or it introduces something that will be discussed or used in subsequent chapters.*

⚠ **Problem 1: Judging Examples**

For each of the following examples, indicate whether it is acceptable or unacceptable. (Don't worry about what prescriptivists might say: we want native speaker intuitions of what sounds right). If it is unacceptable, give an intuitive explanation of what is wrong with it, i.e. whether it:

a. fails to conform to the rules of English grammar (for any variety of English, to the best of your knowledge),
b. is grammatically well-formed, but bizarre in meaning (if so, explain why), or
c. contains a feature of grammar that occurs only in a particular variety of English, for example, slang, or a regional dialect (your own or another); if so, identify the feature. Is it stigmatized in comparison with 'standard' English?

If you are uncertain about any judgments, feel free to consult with others. Nonnative speakers of English, in particular, are encouraged to compare their judgments with others.

(i) Kim and Sandy is looking for a new bicycle.
(ii) Have you the time?
(iii) I've never put the book.
(iv) The boat floated down the river sank.
(v) It ain't nobody goin to miss nobody.
(vi) Terry really likes they.
(vii) Chris must liking syntax.
(viii) Aren't I invited to the party?
(ix) They wondered what each other would do.
(x) There is eager to be fifty students in this class.
(xi) They persuaded me to defend themselves.
(xii) Strings have been pulled many times to get people into Harvard.
(xiii) Terry left tomorrow.
(xiv) A long list of everyone's indiscretions were published in the newspaper.
(xv) Which chemical did you mix the hydrogen peroxide and?
(xvi) There seem to be a good feeling developing among the students.

Problem 2: Reciprocals

English has a 'reciprocal' expression *each other* (think of it as a single word for present purposes), which behaves in some ways like a reflexive pronoun. For example, a direct object *each other* must refer to the subject, and a subject *each other* cannot refer to the direct object:

(i) They like each other.
(ii)*Each other like(s) them.

A. Is there some general property that all antecedents of reciprocals have that not all antecedents of reflexives have? Give both grammatical and ungrammatical examples to make your point.
B. Aside from the difference noted in part (A), do reciprocals behave like reflexives with respect to Hypothesis III? Provide evidence for your answer, including both acceptable and unacceptable examples, illustrating the full range of types of configurations we considered in motivating Hypothesis III.
C. Is the behavior of reciprocals similar to that of reflexives in imperative sentences and in sentences containing *appeal* and *appear*? Again, support your answer with both positive and negative evidence.
D. Consider the following contrast:

    They lost each other's books.
    *They lost themselves' books.

   Discuss how such examples bear on the applicability of Hypothesis III to reciprocals.

   [*Hint: before you answer the question, think about what the verbal arguments are in the above sentences.*]

Problem 3: Ambiguity

Give a brief description of each ambiguity illustrated in (12) on page 11, saying what the source of ambiguity is – that is, whether it is lexical, structural (modificational), or both.

2

Some Simple Theories of Grammar

2.1 Introduction

Among the key points in the previous chapter were the following:

- Language is rule-governed.
- The rules aren't the ones we were taught in school.
- Much of our linguistic knowledge is unconscious, so we have to get at it indirectly; one way of doing this is to consult intuitions of what sounds natural.

In this text, we have a number of objectives. First, we will work toward developing a set of rules that will correctly predict the acceptability of (a large subset of) English sentences. The ultimate goal is a grammar that can tell us for any arbitrary string of English words whether or not it is a well-formed sentence. Thus we will again and again be engaged in the exercise of formulating a grammar that generates a certain set of word strings – the sentences predicted to be grammatical according to that grammar. We will then examine particular members of that set and ask ourselves: 'Is this example acceptable?' The goal then reduces to trying to make the set of sentences generated by our grammar match the set of sentences that we intuitively judge to be acceptable.[1]

A second of our objectives is to consider how the grammar of English differs from the grammar of other languages (or how the grammar of standard American English differs from those of other varieties of English). The conception of grammar we develop will involve general principles that are just as applicable (as we will see in various exercises) to superficially different languages as they are to English. Ultimately, much of the outward differences among languages can be viewed as differences in vocabulary.

This leads directly to our final goal: to consider what our findings might tell us about human linguistic abilities in general. As we develop grammars that include principles of considerable generality, we will begin to see constructs that may have universal applicability to human language. Explicit formulation of such constructs will help us evaluate Chomsky's idea, discussed briefly in Chapter 1, that humans' innate linguistic endowment is a kind of 'Universal Grammar'.

[1] Of course there may be other interacting factors that cause grammatical sentences to sound less than fully acceptable – see Chapter 9 for further discussion. In addition, we don't all speak exactly the same variety of English, though we will assume that existing varieties are sufficiently similar for us to engage in a meaningful discussion of quite a bit of English grammar; see Chapter 15 for more discussion.

In developing the informal rules for reflexive and nonreflexive pronouns in Chapter 1, we assumed that we already knew a lot about the structure of the sentences we were looking at – that is, we talked about subjects, objects, clauses, and so forth. In fact, a fully worked out theory of reflexive and nonreflexive pronouns is going to require that many other aspects of syntactic theory get worked out first. We begin this grammar development process in the present chapter.

We will consider several candidates for theories of English grammar. We begin by quickly dismissing certain simple-minded approaches. We spend more time on a formalism known as 'context-free grammar', which serves as a starting point for most modern theories of syntax. Appendix B includes a brief overview of some of the most important schools of thought within the paradigm of generative grammar, situating the approach developed in this text with respect to some alternatives.

2.2 Two Simplistic Syntactic Theories

2.2.1 Lists as Grammars

The simplest imaginable syntactic theory asserts that a grammar consists of a list of all the well-formed sentences in the language. The most obvious problem with such a proposal is that the list would have to be too long. There is no fixed finite bound on the length of English sentences, as can be seen from the following sequence:

(1) Some sentences go on and on.
Some sentences go on and on and on.
Some sentences go on and on and on and on.
Some sentences go on and on and on and on and on.
. . .

Every example in this sequence is an acceptable English sentence. Since there is no bound on their size, it follows that the number of sentences in the list must be infinite. Hence there are infinitely many sentences of English. Since human brains are finite, they cannot store infinite lists. Consequently, there must be some more compact way of encoding the grammatical knowledge that speakers of English possess.

Moreover, there are generalizations about the structure of English that an adequate grammar should express. For example, consider a hypothetical language consisting of infinitely many sentences similar to those in (1), except that every other sentence reversed the order of the words *some* and *sentences*:[2]

(2) An Impossible Hypothetical Language:

Some sentences go on and on.
Sentences some go on and on and on.
Some sentences go on and on and on and on.
Sentences some go on and on and on and on and on.
*Sentences some go on and on.
*Some sentences go on and on and on.

[2]The asterisks in (2) are intended to indicate the ungrammaticality of the strings in the hypothetical language under discussion, not in normal English.

*Sentences some go on and on and on and on.

*Some sentences go on and on and on and on and on.

...

Of course, none of these sentences[3] where the word *sentences* precedes the word *some* is a well-formed English sentence. Moreover, no natural language exhibits patterns of that sort – in this case, having word order depend on whether the length of the sentence is divisible by 4. A syntactic theory that sheds light on human linguistic abilities ought to explain why such patterns do not occur in human languages. But a theory that said that grammars consisted only of lists of sentences could not do that. If grammars were just lists, then there would be no patterns that would be excluded – and none that would be expected, either.

This form of argument – that a certain theory of grammar fails to 'capture a linguistically significant generalization' – is very common in generative grammar. It takes for granted the idea that language is 'rule governed', that is, that language is a combinatoric system whose operations are 'out there' to be discovered by empirical investigation. If a particular characterization of the way a language works fails to distinguish in a principled way between naturally occurring types of patterns and those that do not occur then it's assumed to be the wrong characterization of the grammar of that language. Likewise, if a theory of grammar cannot describe some phenomenon without excessive redundancy and complications, we assume something is wrong with it. We will see this kind of argumentation again, in connection with proposals that are more plausible than the 'grammars-as-lists' idea. In Chapter 9, we will argue that (perhaps surprisingly), a grammar motivated largely on the basis of considerations of parsimony seems to be a good candidate for a psychological model of the knowledge of language that is employed in speaking and understanding.

2.2.2 Regular Expressions

A natural first step toward allowing grammars to capture generalizations is to classify words into what are often called 'parts of speech' or 'grammatical categories'. There are large numbers of words that behave in similar ways syntactically. For example, the words *apple, book, color*, and *dog* all can appear in roughly the same contexts, such as the following:

(3) a. That __ surprised me.
b. I noticed the __ .
c. They were interested in his __ .
d. This is my favorite __ .

Moreover, they all have plural forms that can be constructed in similar ways (orthographically, simply by adding an *-s*).

Traditionally, the vocabulary of a language is sorted into nouns, verbs, etc. based on loose semantic characterizations (e.g. 'a noun is a word that refers to a person, place, or thing'). While there is undoubtedly a grain of insight at the heart of such definitions,

[3]Note that we are already slipping into a common, but imprecise, way of talking about unacceptable strings of words as 'sentences'.

we can make use of this division into grammatical categories without committing ourselves to any semantic basis for them. For our purposes, it is sufficient that there are classes of words that may occur grammatically in the same environments. Our theory of grammar can capture their common behavior by formulating patterns or rules in terms of categories, not individual words.

Someone might, then, propose that the grammar of English is a list of patterns, stated in terms of grammatical categories, together with a lexicon – that is, a list of words and their categories. For example, the patterns could include (among many others):

(4) a. ARTICLE NOUN VERB
    b. ARTICLE NOUN VERB ARTICLE NOUN

And the lexicon could include (likewise, among many others):

(5) a. Articles: a, the
    b. Nouns: cat, dog
    c. Verbs: attacked, scratched

This mini-grammar licenses forty well-formed English sentences, and captures a few generalizations. However, a grammar that consists of a list of patterns still suffers from the first drawback of the theory of grammars as lists of sentences: it can only account for a finite number of sentences, while a natural language is an infinite set of sentences. For example, such a grammar will still be incapable of dealing with all of the sentences in the infinite sequence illustrated in (1).

We can enhance our theory of grammar so as to permit infinite numbers of sentences by introducing a device that extends its descriptive power. In particular, the problem associated with (1) can be handled using what is known as the 'Kleene star'.[4] Notated as a superscripted asterisk, the Kleene star is interpreted to mean that the expression it is attached to can be repeated any finite number of times (including zero). Thus, the examples in (1) could be abbreviated as follows:

(6) Some sentences go on and on [and on]*.

A closely related notation is a superscripted plus sign (called the Kleene plus), meaning that one or more occurrences of the expression it is attached to are permissible. Hence, another way of expressing the same pattern would be:

(7) Some sentences go on [and on]$^+$.

We shall employ these, as well as two common abbreviatory devices. The first is simply to put parentheses around material that is optional. For example, the two sentence patterns in (4) could be collapsed into: ARTICLE NOUN VERB (ARTICLE NOUN). The second abbreviatory device is a vertical bar, which is used to separate alternatives.[5] For example, if we wished to expand the mini-grammar in (4) to include sentences like *The dog looked big*, we could add the pattern ARTICLE NOUN VERB ADJECTIVE and collapse it with the previous patterns as: ARTICLE NOUN VERB (ARTICLE NOUN)|ADJECTIVE. Of

[4] Named after the mathematician Stephen Kleene.

[5] This is the notation standardly used in computer science and in the study of mathematical properties of grammatical systems. Descriptive linguists tend to use curly brackets to annotate alternatives.

course, we would also have to add the verb *looked* and the adjective *big* to the lexicon.[6]

Patterns making use of the devices just described – Kleene star, Kleene plus, parentheses for optionality, and the vertical bar for alternatives – are known as 'regular expressions'.[7] A great deal is known about what sorts of patterns can and cannot be represented with regular expressions (see, for example, Hopcroft et al. 2001, chaps. 2 and 3), and a number of scholars have argued that natural languages in fact exhibit patterns that are beyond the descriptive capacity of regular expressions (see Bar-Hillel and Shamir 1960, secs. 5 and 6). The most convincing arguments for employing a grammatical formalism richer than regular expressions, however, have to do with the need to capture generalizations.

In (4), the string ARTICLE NOUN occurs twice, once before the verb and once after it. Notice that there are other options possible in both of these positions:

(8) a. *Dogs* chase *cats.*
    b. *A large dog* chased *a small cat.*
    c. *A dog with brown spots* chased *a cat with no tail.*

Moreover, these are not the only positions in which the same strings can occur:

(9) a. Some people yell at *(the) (noisy) dogs (in my neighborhood).*
    b. Some people consider *(the) (noisy) dogs (in my neighborhood)* dangerous.

Even with the abbreviatory devices available in regular expressions, the same lengthy string of symbols – something like (ARTICLE) (ADJECTIVE) NOUN (PREPOSITION ARTICLE NOUN) – will have to appear over and over again in the patterns that constitute the grammar. Moreover, the recurring patterns are in fact considerably more complicated than those illustrated so far. Strings of other forms, such as *the noisy annoying dogs*, *the dogs that live in my neighborhood*, or *Rover, Fido, and Lassie* can all occur in just the same positions. It would clearly simplify the grammar if we could give this apparently infinite set of strings a name and say that any string from the set can appear in certain positions in a sentence.

Furthermore, as we have already seen, an adequate theory of syntax must somehow account for the fact that a given string of words can sometimes be put together in more than one way. If there is no more to grammar than lists of recurring patterns, where these are defined in terms of parts of speech, then there is no apparent way to talk about the ambiguity of sentences like those in (10).

(10) a. We enjoyed the movie with Cher.
     b. The room was filled with noisy children and animals.
     c. People with children who use drugs should be locked up.
     d. I saw the astronomer with a telescope.

[6]This extension of the grammar would license some unacceptable strings, e.g. **The cat scratched big.* Overgeneration is always a danger when extending a grammar, as we will see in subsequent chapters.

[7]This is not intended as a rigorous definition of regular expressions. A precise definition would include the requirement that the empty string is a regular expression, and would probably omit some of the devices mentioned in the text (because they can be defined in terms of others). Incidentally, readers who use computers with the UNIX operating system may be familiar with the command 'grep'. This stands for 'Global Regular Expression Printer'.

In the first sentence, it can be us or the movie that is 'with Cher'; in the second, it can be either just the children or both the children and the animals that are noisy; in the third, it can be the children or their parents who use drugs, and so forth. None of these ambiguities can be plausibly attributed to a lexical ambiguity. Rather, they seem to result from different ways of grouping the words.

In short, the fundamental defect of regular expressions as a theory of grammar is that they provide no means for representing the fact that a string of several words may constitute a unit. The same holds true of several other formalisms that are provably equivalent to regular expressions (including what is known as 'finite-state grammar').

The recurrent strings we have been seeing are usually called 'phrases' or '(syntactic) constituents'.[8] Phrases, like words, come in different types. All of the italicized phrases in (8)–(9) above obligatorily include a noun, so they are called 'Noun Phrases'. The next natural enrichment of our theory of grammar is to permit our regular expressions to include not only words and parts of speech, but also phrase types. Then we also need to provide (similarly enriched) regular expressions to provide the patterns for each type of phrase. The technical name for this theory of grammar is 'context-free phrase structure grammar' or simply 'context-free grammar', sometimes abbreviated as CFG. CFGs, which will also let us begin to talk about structural ambiguity like that illustrated in (10), form the starting point for most serious attempts to develop formal grammars for natural languages.

2.3 Context-Free Phrase Structure Grammar

The term 'grammatical category' now covers not only the parts of speech, but also types of phrase, such as noun phrase and prepositional phrase. To distinguish the two types, we will sometimes use the terms 'lexical category' (for parts of speech) and 'nonlexical category' or 'phrasal category' to mean types of phrase. For convenience, we will abbreviate them, so that 'NOUN' becomes 'N', 'NOUN PHRASE' becomes 'NP', etc.

A context-free phrase structure grammar has two parts:

- A LEXICON, consisting of a list of words, with their associated grammatical categories.[9]
- A set of RULES of the form A $\rightarrow \varphi$ where A is a nonlexical category, and 'φ' stands for a regular expression formed from lexical and/or nonlexical categories; the arrow is to be interpreted as meaning, roughly, 'can consist of'. These rules are called 'phrase structure rules'.

The left-hand side of each rule specifies a phrase type (including the sentence as a type of phrase), and the right-hand side gives a possible pattern for that type of phrase. Because

[8]There is a minor difference in the way these terms are used: linguists often use 'phrase' in contrast to 'word' to mean something longer, whereas words are always treated as a species of constituent.

[9]This conception of a lexicon leaves out some crucial information. In particular, it leaves out information about the meanings and uses of words, except what might be generally associated with the grammatical categories. While this impoverished conception is standard in the formal theory of CFG, attempts to use CFG to describe natural languages have made use of lexicons that also included semantic information. The lexicon we develop in subsequent chapters will be quite rich in structure.

phrasal categories can appear on the right-hand sides of rules, it is possible to have phrases embedded within other phrases. This permits CFGs to express regularities that seem like accidents when only regular expressions are permitted.

A CFG has a designated 'initial symbol', usually notated 'S' (for 'sentence'). Any string of words that can be derived from the initial symbol by means of a sequence of applications of the rules of the grammar is licensed (or, as linguists like to say, 'generated') by the grammar. The language a grammar generates is simply the collection of all of the sentences it generates.[10]

To illustrate how a CFG works, consider the following grammar: (We use 'D' for 'Determiner', which includes what we have up to now been calling 'articles', but will eventually also be used to cover some other things, such as *two* and *my*; 'A' stands for 'Adjective'; 'P' stands for 'Preposition'.)

(11) a. Rules:

S → NP VP
NP → (D) A* N PP*
VP → V (NP) (PP)
PP → P NP

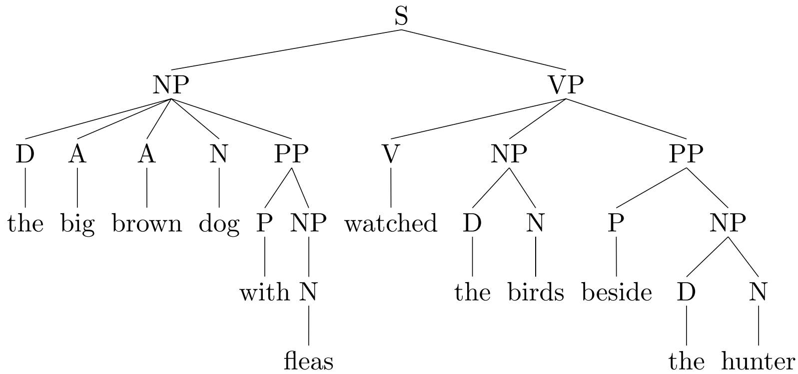
b. Lexicon:

D: the, some
A: big, brown, old
N: birds, fleas, dog, hunter
V: attack, ate, watched
P: for, beside, with

This grammar generates infinitely many English sentences. Let us look in detail at how it generates one sentence: *The big brown dog with fleas watched the birds beside the hunter.* We start with the symbol S, for 'Sentence'. This must consist of the sequence NP VP, since the first rule is the only one with S on the left-hand side. The second rule allows a wide range of possibilities for the NP, one of which is D A A N PP. This PP must consist of a P followed by an NP, by the fourth rule, and the NP so introduced may consist of just an N. The third rule allows VP to consist of V NP PP, and this NP can consist of a D followed by an N. Lastly, the final PP again consists of a P followed by an NP, and this NP also consists of a D followed by an N. Putting these steps together the S may consist of the string D A A N P N V D N P D N, which can be converted into the desired sentence by inserting appropriate words in place of their lexical categories. All of this can be summarized in the following figure (called a 'tree diagram'):

[10]Our definition of CFG differs slightly from the standard ones found in textbooks on formal language theory. Those definitions restrict the right-hand side of rules to finite strings of categories, whereas we allow any regular expression, including those containing the Kleene operators. This difference does not affect the languages that can be generated, although the trees associated with those sentences (see the next section) will be different in some cases.

(12)

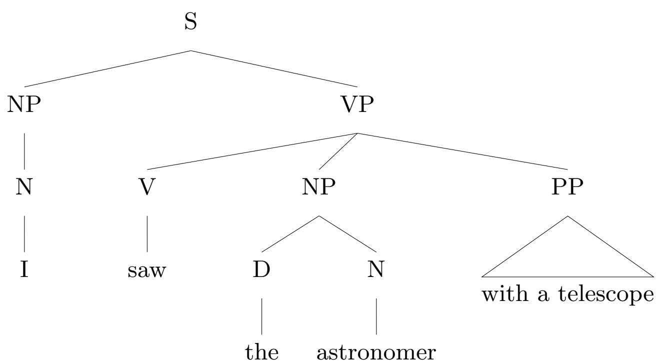


Note that certain sentences generated by this grammar can be associated with more than one tree. (Indeed, the example just given is one such sentence, but finding the other tree will be left as an exercise.) This illustrates how CFGs can overcome the second defect of regular expressions pointed out at the end of the previous section. Recall the ambiguity of (13):

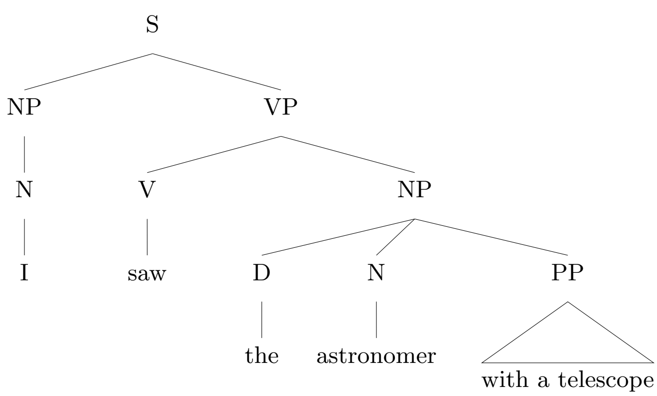
(13) I saw the astronomer with a telescope.

The distinct interpretations of this sentence ('I used the telescope to see the astronomer'; 'I saw the astronomer who had a telescope') correspond to distinct tree structures that our grammar will assign to this string of words. The first interpretation corresponds to (14a) and the latter to (14b):

(14) a.



b.



CFG thus provides us with a straightforward mechanism for expressing such ambiguities, whereas grammars that use only regular expressions don't.

The normal way of talking about words and phrases is to say that certain sequences of words (or categories) 'form a constituent'. What this means is that these strings function as units for some purpose (for example, the interpretation of modifiers) within the sentences in which they appear. So in (12), the sequence *with fleas* forms a PP constituent (as does the sequence P NP), *the big brown dog with fleas* forms an NP, and the sequence *dog with fleas* forms no constituent. Structural ambiguity arises whenever a string of words can form constituents in more than one way.

Exercise 1: Practice with CFG
Assume the CFG grammar given in (11). Draw the tree structure for the other interpretation (i.e. not the one shown in (12)) of *The big brown dog with fleas watched the birds beside the hunter.*

2.4 Applying Context-Free Grammar

In the previous section, we introduced the formalism of context-free grammar and showed how it allows us to generate infinite collections of English sentences with simple rules. We also showed how it can provide a rather natural representation of certain ambiguities we find in natural languages. But the grammar we presented was just a teaching tool, designed to illustrate certain properties of the formalism; it was not intended to be taken seriously as an attempt to analyze the structure of English. In this section, we begin by motivating some phrase structure rules for English. In the course of doing this, we develop a new test for determining which strings of words are constituents. We also introduce a new abbreviatory convention that permits us to collapse many of our phrase structure rules into rule schemas.

2.4.1 Some Phrase Structure Rules for English

For the most part, we will use the traditional parts of speech, such as noun, verb, adjective, and preposition. In some cases, we will find it useful to introduce grammatical categories that might be new to readers, and we may apply the traditional labels somewhat differently than in traditional grammar books. But the traditional classification of words into types has proved to be an extremely useful categorization over the past two millennia, and we see no reason to abandon it wholesale.

We turn now to phrases, beginning with noun phrases.

Noun Phrases

Nouns can appear in a number of positions, e.g. those occupied by the three nouns in *Dogs give people fleas.* These same positions also allow sequences of an article followed by a noun, as in *The child gave the dog a bath.* Since the place of the article can also be filled by demonstratives (e.g. *this, these*), possessives (e.g. *my, their*), or quantifiers (e.g. *each*, *some*, *many*), we use the more general term 'determiner' (abbreviated D) for this category. We can capture these facts by positing a type of phrase we'll call NP (for 'noun phrase'), and the rule NP $\rightarrow$ (D) N. As we saw earlier in this chapter, this rule will need to be elaborated later to include adjectives and other modifiers. First, however, we should consider a type of construction we have not yet discussed.

Coordination

To account for examples like *A dog, a cat, and a wombat fought*, we want a rule that allows sequences of NPs, with *and* before the last one, to appear where simple NPs can occur. A rule that does this is NP $\rightarrow$ NP$^+$ CONJ NP. (Recall that NP$^+$ means a string of one or more NPs).

Whole sentences can also be conjoined, as in *The dog barked, the donkey brayed, and the pig squealed.*[11] Again, we could posit a rule like S $\rightarrow$ S$^+$ CONJ S. But now we have two rules that look an awful lot alike. We can collapse them into one rule schema as follows, where the variable 'X' can be replaced by any grammatical category name (and 'CONJ' is the category of conjunctions like *and* and *or*, which will have to be listed in the lexicon):

(15) X $\rightarrow$ X$^+$ CONJ X.

Now we have made a claim that goes well beyond the data that motivated the rule, namely, that elements of any category can be conjoined in the same way. If this is correct, then we can use it as a test to see whether a particular string of words should be treated as a phrase. In fact, coordinate conjunction is widely used as a test for constituency – that is, as a test for which strings of words form phrases. Though it is not an infallible diagnostic, we will use it as one of our sources of evidence for constituent structure.

Verb Phrases

Consider (16):

(16) A neighbor yelled, chased the cat, and gave the dog a bone.

(16) contains the coordination of strings consisting of V, V NP, and V NP NP. According to (15), this means that all three strings are constituents of the same type. Hence, we posit a constituent which we'll call VP, described by the rule VP $\rightarrow$ V (NP) (NP). VP is introduced by the rule S $\rightarrow$ NP VP. A tree structure for the coordinate VP in (16) would be the following:

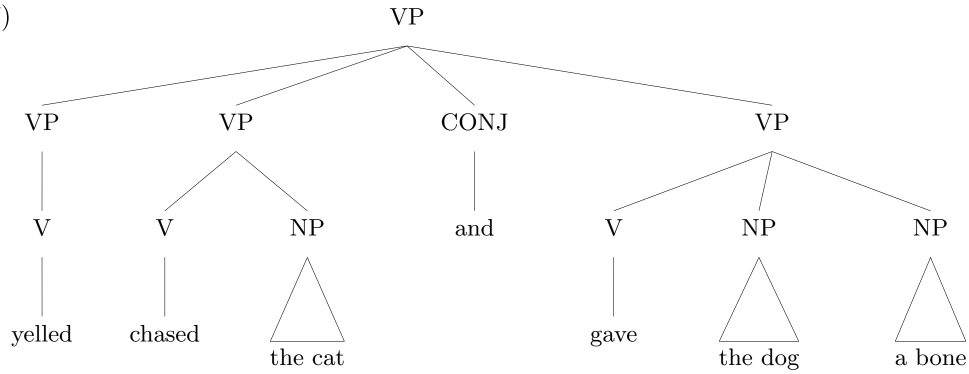
[11] There are other kinds of coordinate sentences that we are leaving aside here – in particular, elliptical sentences that involve coordination of nonconstituent sequences:

(i) Chris likes blue and Pat green.

(ii) Leslie wants to go home tomorrow, and Terry, too.

Notice that this kind of sentence, which will not be treated by the coordination rule discussed in the text, has a characteristic intonation pattern – the elements after the conjunction form separate intonational units separated by pauses.

(17)



Prepositional Phrases

Expressions like *in Rome* or *at noon* that denote places or times ('locative' and 'temporal' expressions, as linguists would say) can be added to almost any sentence, and to NPs, too. For example:

(18) a. The fool yelled at noon.
b. This disease gave Leslie a fever in Rome.
c. A tourist in Rome laughed.

These are constituents, as indicated by examples like *A tourist yelled at noon and at midnight, in Rome and in Paris.* We can get lots of them in one sentence, for example, *A tourist laughed on the street in Rome at noon on Tuesday.* These facts can be incorporated into the grammar in terms of the phrasal category PP (for 'prepositional phrase'), and the rules:

(19) a. PP $\rightarrow$ P NP
b. VP $\rightarrow$ VP PP

Since the second rule has VP on both the right and left sides of the arrow, it can apply to its own output. (Such a rule is known as a RECURSIVE rule).[12] Each time it applies, it adds a PP to the tree structure. Thus, this recursive rule permits arbitrary numbers of PPs within a VP.

As mentioned earlier, locative and temporal PPs can also occur in NPs, for example, *A protest on the street in Rome on Tuesday at noon disrupted traffic.* The most obvious analysis to consider for this would be a rule that said: NP $\rightarrow$ NP PP. However, we're going to adopt a slightly more complex analysis. We posit a new nonlexical category, which we'll call NOM (for 'nominal'), and we replace our old rule: NP $\rightarrow$ (D) N with the following:

(20) a. NP $\rightarrow$ (D) NOM
b. NOM $\rightarrow$ N
c. NOM $\rightarrow$ NOM PP

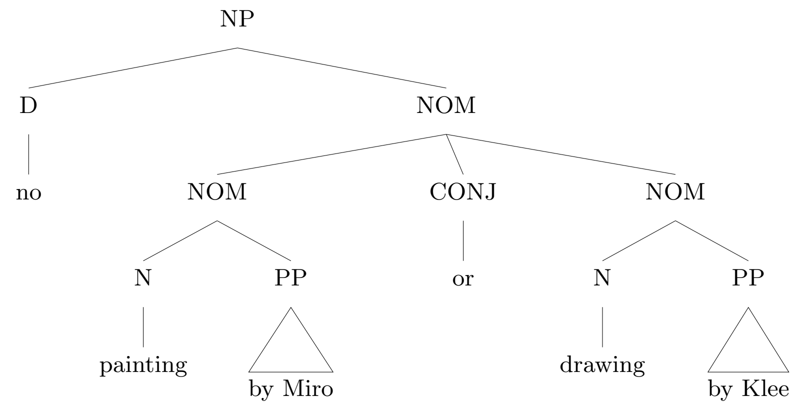
[12] More generally, we use the term RECURSION whenever rules permit a constituent to occur within a larger constituent of the same type.

The category NOM will be very useful later in the text. For now, we will justify it with the following sentences:

(21) a. The love of my life and mother of my children would never do such a thing.
b. The museum displayed no painting by Miro or drawing by Klee.

(21b) means that the museum displayed neither paintings by Miro nor drawings by Klee. That is, the determiner *no* must be understood as 'having scope' over both *painting by Miro* and *drawing by Klee* – it applies to both phrases. The most natural noun phrase structure to associate with this interpretation is:

(22)



This, in turn, is possible with our current rules if *painting by Miro or drawing by Klee* is a conjoined NOM. It would not be possible without NOM.

Similarly, for (21a), *the* has scope over both *love of my life* and *mother of my children* and hence provides motivation for an analysis involving coordination of NOM constituents.

2.4.2 Summary of Grammar Rules

Our grammar now has the following rules:

(23)
S $\rightarrow$ NP VP
NP $\rightarrow$ (D) NOM
VP $\rightarrow$ V (NP) (NP)
NOM $\rightarrow$ N
NOM $\rightarrow$ NOM PP
VP $\rightarrow$ VP PP
PP $\rightarrow$ P NP
X $\rightarrow$ X^{+} CONJ X

In motivating this grammar, we used three types of evidence for deciding how to divide sentences up into constituents:

- In ambiguous sentences, a particular division into constituents sometimes can provide an account of the ambiguity in terms of where some constituent is attached (as in (14)).

- Coordinate conjunction usually combines constituents, so strings that can serve as coordinate conjuncts are probably constituents (as we argued for VPs, PPs, and NOMs in the last few pages).
- Strings that can appear in multiple environments are typically constituents.

We actually used this last type of argument for constituent structure only once. That was when we motivated the constituent NP by observing that pretty much the same strings could appear as subject, object, or object of a preposition. In fact, variants of this type of evidence are commonly used in linguistics to motivate particular choices about phrase structure. In particular, there are certain environments that linguists use as diagnostics for constituency – that is, as a way of testing whether a given string is a constituent.

Probably the most common such diagnostic is occurrence before the subject of a sentence. In the appropriate contexts, various types of phrases are acceptable at the beginning of a sentence. This is illustrated in the following sentences, with the constituent in question italicized, and its label indicated in parentheses after the example:

(24) a. Most elections are quickly forgotten, but *the election of 2000*, everyone will remember for a long time. (NP)
b. You asked me to fix the drain, and *fix the drain*, I shall. (VP)
c. *In the morning*, they drink tea. (PP)

Another environment that is frequently used as a diagnostic for constituency is what is sometimes called the 'cleft' construction. It has the following form: *It is* (or *was*) __ *that* ... For example:

(25) a. It was *a book about syntax* that she was reading. (NP)
b. It is *study for the exam* that I urgently need to do. (VP)
c. It is *after lunch* that they always fall asleep. (PP)

Such diagnostics can be very useful in deciding how to divide up sentences into phrases. However, some caution in their use is advisable. Some diagnostics work only for some kinds of constituents. For example, while coordination provided some motivation for positing NOM as a constituent (see (21)), NOM cannot appear at the beginning of a sentence or in a cleft:

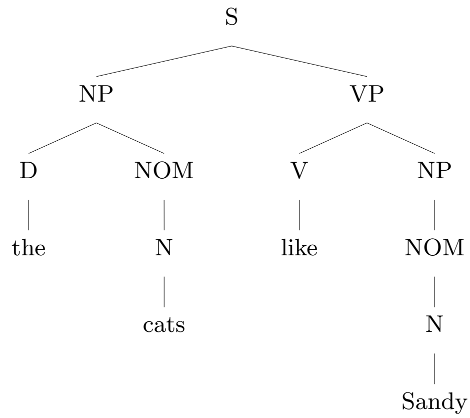
(26) a.*Many artists were represented, but painting by Klee or drawing by Miro the museum displayed no.
b.*It is painting by Klee or drawing by Miro that the museum displays no.

More generally, these tests should be regarded only as heuristics, for there may be cases where they give conflicting or questionable results. Nevertheless, they can be very useful in deciding how to analyze particular sentences, and we will make use of them in the chapters to come.

2.5 Trees Revisited

In grouping words into phrases and smaller phrases into larger ones, we are assigning internal structure to sentences. As noted earlier, this structure can be represented as a tree diagram. For example, our grammar so far generates the following tree:

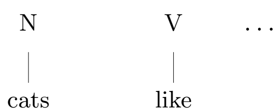
(27)



A tree is said to consist of NODES, connected by BRANCHES. A node above another on a branch is said to DOMINATE it. The nodes at the bottom of the tree – that is, those that do not dominate anything else – are referred to as TERMINAL (or LEAF) nodes. A node right above another node on a tree is said to be its MOTHER and to IMMEDIATELY DOMINATE it. A node right below another on a branch is said to be its DAUGHTER. Two daughters of the same mother node are, naturally, referred to as SISTERS.

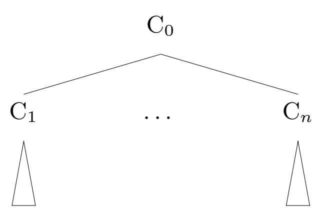
One way to think of the way in which a grammar of this kind defines (or generates) trees is as follows. First, we appeal to the lexicon (still conceived of as just a list of words paired with their grammatical categories) to tell us which lexical trees are well-formed. (By 'lexical tree', we simply mean a tree consisting of a word immediately dominated by its grammatical category.) So if *cats* is listed in the lexicon as belonging to the category N, and *like* is listed as a V, and so forth, then lexical structures like the following are well-formed:

(28)



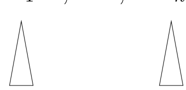
And the grammar rules are equally straightforward. They simply tell us how well-formed trees (some of which may be lexical) can be combined into bigger ones:

(29)



is a well-formed nonlexical tree if (and only if)

C_1 , . . . , C_n are well-formed trees, and



$C_0 \rightarrow C_1 \ \ldots \ C_n$ is a grammar rule.

So we can think of our grammar as generating sentences in a 'bottom-up' fashion – starting with lexical trees, and then using these to build bigger and bigger phrasal trees, until we build one whose top node is S. The set of all sentences that can be built that have S as their top node is the set of sentences the grammar generates. But note that our grammar could just as well have been used to generate sentences in a 'top-down' manner, starting with S. The set of sentences generated in this way is exactly the same. A CFG is completely neutral with respect to top-down and bottom-up perspectives on analyzing sentence structure. There is also no particular bias toward thinking of the grammar in terms of generating sentences or in terms of parsing. Instead, the grammar can be thought of as constraining the set of all possible phrase structure trees, defining a particular subset as well-formed.

Direction neutrality and process neutrality are consequences of the fact that the rules and lexical entries simply provide constraints on well-formed structure. As we will suggest in Chapter 9, these are in fact important design features of this theory (and of those we will develop that are based on it), as they facilitate the direct embedding of the abstract grammar within a model of language processing.

The lexicon and grammar rules together thus constitute a system for defining not only well-formed word strings (i.e. sentences), but also well-formed tree structures. Our statement of the relationship between the grammar rules and the well-formedness of trees is at present rather trivial, and our lexical entries still consist simply of pairings of words with parts of speech. As we modify our theory of grammar and enrich our lexicon, however, our attention will increasingly turn to a more refined characterization of which trees are well-formed.

2.6 CFG as a Theory of Natural Language Grammar

As was the case with regular expressions, the formal properties of CFG are extremely well studied (see Hopcroft et al. 2001, chaps. 4–6 for a summary). In the early 1960s, several scholars published arguments purporting to show that natural languages exhibit properties beyond the descriptive capacity of CFGs. The pioneering work in the first two decades of generative grammar was based on the assumption that these arguments were sound. Most of that work can be viewed as the development of extensions to CFG designed to deal with the richness and complexity of natural languages. Similarly, the theory we develop in this book is in essence an extended version of CFG, although our extensions are rather different in kind from some of the earlier ones.

In 1982, Geoffrey Pullum and Gerald Gazdar published a paper showing that the earlier arguments against the adequacy of CFG as a theory of natural language structure all contained empirical or mathematical flaws (or both). This led to a flurry of new work on the issue, culminating in new arguments that natural languages were not describable by CFGs. The mathematical and empirical work that resulted from this controversy substantially influenced the theory of grammar presented in this text. Many of the central papers in this debate were collected together by Savitch et al. (1987); of particular interest are Pullum and Gazdar's paper and Shieber's paper in that volume.

While the question of whether natural languages are in principle beyond the generative capacity of CFGs is of some intellectual interest, working linguists tend to be more concerned with determining what sort of formalisms can provide elegant and enlightening accounts of linguistic phenomena in practice. Hence the arguments that tend to carry the most weight are ones about what formal devices are needed to capture linguistically significant generalizations. In the next section and later chapters, we will consider some phenomena in English that suggest that the simple version of CFG introduced above needs to be extended.

Accompanying the 1980s revival of interest in the mathematical properties of natural languages, considerable attention was given to the idea that, with an appropriately designed theory of syntactic features and general principles, context-free phrase structure grammar could serve as an empirically adequate theory of natural language syntax. This proposition was explored in great detail by Gazdar et al. (1985), who developed the theory known as 'Generalized Phrase Structure Grammar' (or GPSG). Work in phrase structure grammar advanced rapidly, and GPSG quickly evolved into a new framework, now known as 'Head-driven Phrase Structure Grammar' (HPSG), whose name reflects the increased importance of information encoded in the lexical heads[13] of syntactic phrases. The theory of grammar developed in this text is most closely related to current HPSG. See Appendix B for discussion of these and other modern theories of grammar.

2.7 Problems with CFG

Two of our arguments against overly simple theories of grammar at the beginning of this chapter were that we wanted to be able to account for the infinity of language, and that we wanted to be able to account for structural ambiguity. CFG addresses these problems, but, as indicated in the previous section, simple CFGs like the ones we have seen so far are not adequate to account for the full richness of natural language syntax. This section introduces some of the problems that arise in trying to construct a CFG of English.

2.7.1 Heads

As we have seen, CFGs can provide successful analyses of quite a bit of natural language. But if our theory of natural language syntax were nothing more than CFG, our theory would fail to predict the fact that certain kinds of CF rules are much more natural than others. For example, as far as we are aware, no linguist has ever wanted to write rules like those in (30) in describing any human language:

(30) Unnatural Hypothetical Phrase Structure Rules

VP → P NP

NP → PP S

What is it that is unnatural about the rules in (30)? An intuitive answer is that the categories on the left of the rules don't seem appropriate for the sequences on the right. For example, a VP should have a verb in it. This then leads us to consider why we named NP, VP, and PP after the lexical categories N, V, and P. In each case, the phrasal category was named after a lexical category that is an obligatory part of that kind of phrase. At least in the case of NP and VP, all other parts of the phrase may sometimes be absent (e.g. *Dogs bark*).

[13] The notion of 'head' will be discussed in Section 2.7.1 below.

The lexical category that a phrasal category derives its name from is called the HEAD of the phrase. This notion of 'headedness' plays a crucial role in all human languages and this fact points out a way in which natural language grammars differ from some kinds of CFG. The formalism of CFG, in and of itself, treats category names as arbitrary: our choice of pairs like 'N' and 'NP', etc., serves only a mnemonic function in simple CFGs. But we want our theory to do more. Many phrase structures of natural languages are headed structures, a fact we will build into the architecture of our grammatical theory. To do this, we will enrich the way we represent grammatical categories, so that we can express directly what a phrase and its head have in common. This will lead eventually to a dramatic reduction in the number of grammar rules required.

The notion of headedness is a problem for CFG because it cuts across many different phrase types, suggesting that the rules are too fine-grained. The next two subsections discuss problems of the opposite type – that is, ways in which the syntax of English is sensitive to finer-grained distinctions among grammatical categories than a simple CFG can encode.

2.7.2 Subcategorization

The few grammar rules we have so far cover only a small fragment of English. What might not be so obvious, however, is that they also overgenerate – that is, they generate strings that are not well-formed English sentences. Both *denied* and *disappeared* would be listed in the lexicon as members of the category V. This classification is necessary to account for sentences like (31):

(31) a. The defendant denied the accusation.
b. The problem disappeared.

But this classification would also permit the generation of the ungrammmatical examples in (32):

(32) a.*The defendant denied.
b.*The teacher disappeared the problem.

Similarly, the verb *handed* must be followed by two NPs, but our rules allow a VP to be expanded in such a way that any V can be followed by only one NP, or no NPs at all. That is, our current grammar fails to distinguish among the following:

(33) a. The teacher handed the student a book.
b.*The teacher handed the student.
c.*The teacher handed a book.
d.*The teacher handed.

To rule out the ungrammatical examples in (33), we need to distinguish among verbs that cannot be followed by an NP, those that must be followed by one NP, and those that must be followed by two NPs. These classes are often referred to as INTRANSITIVE, TRANSITIVE, and DITRANSITIVE verbs, respectively. In short, we need to distinguish subcategories of the category V.

One possible approach to this problem is simply to conclude that the traditional category of 'verb' is too coarse-grained for generative grammar, and that it must be replaced by at least three distinct categories, which we can call IV, TV, and DTV. We can then replace our earlier phrase structure rule

VP $\rightarrow$ V (NP) (NP)

with the following three rules:

(34) a. VP $\rightarrow$ IV
b. VP $\rightarrow$ TV NP
c. VP $\rightarrow$ DTV NP NP

2.7.3 Transitivity and Agreement

Most nouns and verbs in English have both singular and plural forms. In the case of nouns, the distinction between, say, *bird* and *birds* indicates whether the word is being used to refer to just one fowl or a multiplicity of them. In the case of verbs, distinctions like the one between *sing* and *sings* indicate whether the verb's subject refers to one or many individuals. In present tense English sentences, the plurality marking on the head noun of the subject NP and that on the verb must be consistent with each other. This is referred to as SUBJECT-VERB AGREEMENT (or sometimes just 'agreement' for short). It is illustrated in (35):

(35) a. The bird sings.
b. Birds sing.
c.*The bird sing.[14]
d.*Birds sings.

Perhaps the most obvious strategy for dealing with agreement is the one considered in the previous section. That is, we could divide our grammatical categories into smaller categories, distinguishing singular and plural forms. We could then replace the relevant phrase structure rules with more specific ones. In examples like (35), we could distinguish lexical categories of N-SG and N-PL, as well as IV-SG and IV-PL. Then we could replace the rule

S $\rightarrow$ NP VP

with two rules:

S $\rightarrow$ NP-SG VP-SG

and

S $\rightarrow$ NP-PL VP-PL

But since the marking for number appears on the head noun and head verb, other rules would also have to be changed. Specifically, the rules expanding NP and VP all would have to be divided into pairs of rules expanding NP-SG, NP-PL, VP-SG, and VP-PL. Hence, we would need all of the following:

(36) a. NP-SG $\rightarrow$ (D) NOM-SG
b. NP-PL $\rightarrow$ (D) NOM-PL
c. NOM-SG $\rightarrow$ NOM-SG PP

[14] There are dialects of English in which this is grammatical, but we will be analyzing the more standard dialect in which agreement marking is obligatory.

d. NOM-PL $\rightarrow$ NOM-PL PP
e. NOM-SG $\rightarrow$ N-SG
f. NOM-PL $\rightarrow$ N-PL
g. VP-SG $\rightarrow$ IV-SG
h. VP-PL $\rightarrow$ IV-PL
i. VP-SG $\rightarrow$ VP-SG PP
j. VP-PL $\rightarrow$ VP-PL PP

This set of rules is cumbersome, and clearly misses linguistically significant generalizations. The rules in this set come in pairs, differing only in whether the category names end in '-SG' or '-PL'. Nothing in the formalism or in the theory predicts this pairing. The rules would look no less natural if, for example, the rules expanding -PL categories had their right-hand sides in the reverse order from those expanding -SG categories. But languages exhibiting this sort of variation in word order do not seem to exist.

Things get even messier when we consider transitive and ditransitive verbs. Agreement is required regardless of whether the verb is intransitive, transitive, or ditransitive. Thus, along with (35), we have (37) and (38):

(37) a. The bird devours the worm.
b. The birds devour the worm.
c.*The bird devour the worm.
d.*The birds devours the worm.

(38) a. The bird gives the worm a tug.
b. The birds give the worm a tug.
c.*The bird give the worm a tug.
d.*The birds gives the worm a tug.

If agreement is to be handled by the rules in (39):

(39) a. S $\rightarrow$ NP-SG VP-SG
b. S $\rightarrow$ NP-PL VP-PL

then we will now need to introduce lexical categories TV-SG, TV-PL, DTV-SG, and DTV-PL, along with the necessary VP-SG and VP-PL expansion rules (as well as the two rules in (39)). What are the rules for VP-SG and VP-PL when the verb is transitive or ditransitive? For simplicity, we will look only at the case of VP-SG with a transitive verb. Since the object of the verb can be either singular or plural, we need two rules:

(40) a. VP-SG $\rightarrow$ TV-SG NP-SG
b. VP-SG $\rightarrow$ TV-SG NP-PL

Similarly, we need two rules for expanding VP-PL when the verb is transitive, and four rules each for expanding VP-SG and VP-PL when the verb is ditransitive (since each object can be either singular or plural). Alternatively, we could make all objects of category NP and introduce the following two rules:

(41) a. NP $\rightarrow$ NP-SG
b. NP $\rightarrow$ NP-PL

This would keep the number of VP-SG and VP-PL rules down to three each (rather than seven each), but it introduces extra noun phrase categories. Either way, the rules are full of undesirable redundancy.

Matters would get even worse when we examine a wider range of verb types. So far, we have only considered how many NPs must follow each verb. But there are verbs that only appear in other environments; for example, some verbs require following PPs or Ss, as in (42).

(42) a. Terry wallowed in self-pity.
b.*Terry wallowed.
c.*Terry wallowed the self-pity.
d. Kerry remarked (that) it was late.
e.*Kerry remarked.
f.*Kerry remarked the time.

Exercise 2: Wallowing in Categories

A. Provide examples showing that the verbs *wallow* and *remark* exhibit the same agreement patterns as the other types of verbs we have been discussing.
B. What additional categories and rules would be required to handle these verbs?

When a broader range of data is considered, it is evident that the transitivity distinctions we have been assuming are simply special cases of a more general phenomenon. Some verbs (and, as we will see later, some other types of words as well) occur only in the environment of particular kinds of constituents. In English, these constituents characteristically occur after the verb, and syntacticians call them COMPLEMENTS. Complements will be discussed in greater detail in Chapter 4.

It should be clear by now that as additional coverage is incorporated – such as adjectives modifying nouns – the redundancies will proliferate. The problem is that we want to be able to talk about nouns and verbs as general classes, but we have now divided nouns into (at least) two categories (N-SG and N-PL) and verbs into six categories (IV-SG, IV-PL, TV-SG, TV-PL, DTV-SG, and DTV-PL). To make agreement work, this multiplication of categories has to be propagated up through at least some of the phrasal categories. The result is a very long and repetitive list of phrase structure rules.

What we need is a way to talk about subclasses of categories, without giving up the commonality of the original categories. That is, we need a formalism that permits us to refer straightforwardly to, for example, all verbs, all singular verbs, all ditransitive verbs, or all singular ditransitive verbs. In the next chapter, we introduce a device that will permit us to do this.

2.8 Transformational Grammar

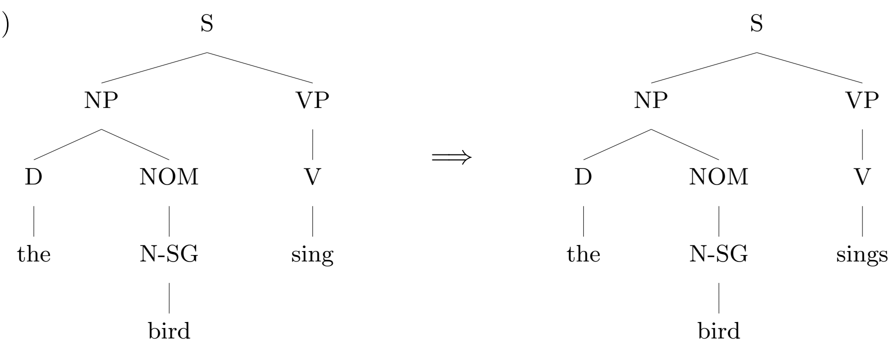
As noted in Section 2.6, much of the work in generative grammar (including this book) has involved developing extensions of Context Free Grammar to make it better adapted to the task of describing natural languages. The most celebrated proposed extension was a kind of rule called a ‘transformation’, as introduced into the field of generative grammar by Noam Chomsky.[15] Transformations are mappings from phrase structure representations to phrase structure representations (from trees to trees, in our terms) that can copy, delete, and permute parts of trees, as well as insert specified new material into them. The initial trees were to be generated by a CFG. For example, in early work on transformations, it was claimed that declarative and interrogative sentence pairs (such as *The sun is shining* and *Is the sun shining?*) were to be derived from the same underlying phrase structure by a transformation that moved certain verbs to the front of the sentence. Likewise, passive sentences (such as *The cat was chased by the dog*) were derived from the same underlying structures as their active counterparts (*The dog chased the cat*) by means of a passivization transformation. The name ‘transformational grammar’ is sometimes used for theories positing rules of this sort.[16]

In a transformational grammar, then, each sentence is associated not with a single tree structure, but with a sequence of such structures. This greatly enriches the formal options for describing particular linguistic phenomena.

For example, subject-verb agreement can be handled in transformational terms by assuming that number (that is, being singular or plural) is an intrinsic property of nouns, but not of verbs. Hence, in the initial tree structures for sentences, the verbs have no number associated with them. Subsequently, a transformation changes the form of the verb to the one that agrees with the subject NP. Such an analysis avoids the proliferation of phrase structure rules described in the preceding section, but at the cost of adding an agreement transformation.

As an illustration of how this would work, consider again the contrast in (35)[17]. Instead of creating separate singular and plural versions of NP, VP, NOM, N, and V (with the corresponding phrase structure rules in (36)), a transformational analysis could limit this bifurcation of categories to N-SG and N-PL (with the rules NOM $\rightarrow$ N-SG and NOM $\rightarrow$ N-PL). In addition, an agreement transformation (which we will not try to formalize here) would give the verb the correct form, roughly as follows:

(43)



[15]The original conception of a transformation, as developed in the early 1950s by Zellig Harris, was intended somewhat differently – as a way of regularizing the information content of texts, rather than as a system for generating sentences.

[16]See Appendix B for more discussion of varieties of transformational grammar.

[17]The analysis sketched in this paragraph is a simplified version of the one developed by Chomsky (1957). It has long since been superceded by other analyses. In presenting it here (for pedagogical purposes) we do not mean to suggest that contemporary transformationalists would advocate it.

Notice that in a theory that posits a passivization transformation (which, among other things, would move the object NP into subject position), something like the agreement transformation described in the previous paragraph would be required. To make this more concrete, consider examples like (44):

(44) a. Everyone loves puppies.
b. Puppies are loved by everyone.

Substituting the singular form of the verb in (44b) results in ill-formedness:

(45) *Puppies is loved by everyone.

In a transformational analysis, *puppies* only becomes the subject of the sentence following application of the passivization transformation. Since agreement (in English) is consistently with the subject NP, if transformations are permitted to change which NP is the subject, agreement cannot be determined until after such transformations have applied.

In general, transformational analyses involve such rule interactions. Many transformational derivations involve highly abstract underlying structures with complex sequences of transformations deriving the observable forms.

Because versions of transformational grammar have been so influential throughout the history of generative grammar, many of the phenomena to be discussed have come to be labeled with names that suggest transformational analyses (e.g. "raising", discussed in Chapter 12).

This influence is also evident in work on the psychology of language. In contemplating the mental processes underlying language use, linguists naturally make reference to their theories of language structure, and there have been repeated efforts over the years to find evidence that transformational derivations play a role in at least some aspects of language processing.

In later chapters, we will on occasion be comparing our (nontransformational) analyses with transformational alternatives. We make no pretense of doing justice to all varieties of transformational grammar in this text. Our concern is to develop a theory that can provide rigorous and insightful analyses of a wide range of the structures found in natural languages. From time to time, it will be convenient to be able to consider alternative approaches, and these will often be transformational.

2.9 What Are Grammars Theories Of?

In the opening paragraphs of Chapter 1, we said that linguists try to study language scientifically. We then went on to describe some of the grammatical phenomena that we would be investigating in this book. In this chapter, we have taken the first steps towards formulating a precise theory of grammar, and we have presented evidence for particular formulations over others.

We have not, however, said much about what a grammar is taken to be a theory of. Chapter 1 discussed the view, articulated most forcefully by Chomsky, that one reason for studying language is to gain insight into the workings of the human mind. On this view – which is shared by many but by no means all linguists – choosing one form of grammar over another constitutes a psychological hypothesis. That is, a grammar is a theory about the mental representation of linguistic knowledge.

As we noted, there are other views. Some linguists point out that communicating through language requires that different people share a common set of conventions. Any approach to language that seeks to represent only what is in the mind of an individual speaker necessarily gives short shrift to this social aspect of language.

To begin to get a handle on these issues, consider a concrete example: Pat says, “What time is it?” and Chris answers, “It’s noon”. The two utterances are physical events that are directly observable. But each of them is an instance of a sentence, and both of these sentences have been uttered many times. As syntacticians, we are interested in only some properties of these utterances; other properties, such as where they were uttered and by whom, are not relevant to our concerns. Moreover, there are many other English sentences that have never been spoken (or written), but they still have properties that our grammar should characterize. In short, the subject matter of our theory is sentences, which are abstractions, rather than observable physical events. We are interested in particular utterances only as evidence of something more abstract and general, just as a biologist is only interested in particular organisms as instances of something more abstract and general, such as a species.

A grammar of English should characterize the structure and meaning of both Pat’s utterance and Chris’s. So we need to abstract across different speakers, too. This raises some difficult issues, because no two speakers have exactly the same linguistic knowledge. In fact, linguistic differences among individuals and groups of individuals make it notoriously difficult to draw boundaries between languages. The conventional labels applied to languages (such as English, Chinese, or Arabic) are determined as much by political facts as by linguistic ones.[18] It is largely for this reason that Chomsky and many other linguists say that their object of study is the mental representations of individual speakers.

Of course, similar difficulties arise in drawing boundaries between species, but few biologists would say on those grounds that biology should only be concerned with the DNA of individual organisms. Just as biologists seek to generalize across populations of heterogeneous individuals, we want our grammar to characterize something more general than what is in one person’s mind. Occasionally, we will deal with phenomena which are not uniform across all varieties of English (see especially Chapter 15).

In short, we want our grammar to characterize the syntax of English. This involves multiple levels of abstraction from what is directly observable, as well as some attention to variation among speakers. Our object of study is not purely a matter of individual psychology, nor is it exclusively a social phenomenon. There are some aspects of language that are primarily manifestations of individual speakers’ mental representations and others that critically involve the interactions of multiple language users. Just as molecular biology and population biology both contribute to our understanding of species, linguists need not make an exclusive choice between an internal and an external perspective.

2.10 Summary

In this chapter, we began our search for an adequate model of the grammar of one natural language: English. We considered and rejected two simple approaches to grammar,

[18]Linguists sometimes joke that a ‘language’ is simply a ‘dialect’ with an army and a navy.

including a theory based on regular expressions ('finite-state grammar'). The theory of context-free grammars, by contrast, solves the obvious defects of these simple approaches and provides an appropriate starting point for the grammatical description of natural language. However, we isolated two ways in which context-free grammars are inadequate as a theory of natural language:

- CFGs are arbitrary. They fail to capture the 'headedness' that is characteristic of many types of phrase in natural language.
- CFGs are redundant. Without some way to refer to kinds of categories rather than just individual categories, there is no way to eliminate the massive redundancy that will be required in order to analyze the agreement and subcategorization patterns of natural languages.

For these reasons, we cannot accept CFG alone as a theory of grammar. As we will show in the next few chapters, however, it is possible to retain much of the character of CFG as we seek to remedy its defects.

2.11 Further Reading

The standard reference work for the basic mathematical results on formal languages (including regular expressions and context-free languages) is Hopcroft et al. 2001. Partee et al. 1990 covers much of the same material from a more linguistic perspective. Classic works arguing against the use of context-free grammars for natural languages include Chomsky 1963 and Postal 1964. Papers questioning these arguments, and other papers presenting new arguments for the same conclusion are collected in Savitch et al. 1987. For (somewhat dated) surveys of theories of grammar, see Sells 1985 and Wasow 1989. A more detailed presentation of GPSG is Gazdar et al. 1985. The history of generative grammar is presented from different perspectives by Matthews (1993), Newmeyer (1986), Harris (1993), and Huck and Goldsmith (1995).

Perhaps the best discussions of the basic phrase structures of English are to be found in good descriptive grammars, such as Quirk et al. 1972, 1985, Huddleston and Pullum 2002, or Greenbaum 1996. Important discussions of the notion of 'head' and its role in phrase structure can be found in Chomsky 1970 and Gazdar and Pullum 1981. A detailed taxonomy of the subcategories of English verbs is provided by Levin (1993).

2.12 Problems

⚠ Problem 1: More Practice with CFG

Assume the grammar rules given in (23), but with the following lexicon:

D:	a, the
V:	admired, disappeared, put, relied
N:	cat, dog, hat, man, woman, roof
P:	in, on, with
CONJ:	and, or

A. Give a well-formed English sentence that this grammar sanctions and assigns only one structure to. Draw the tree structure that the grammar assigns to it.
B. Give a well-formed English sentence that is structurally ambiguous according to this grammar. Draw two distinct tree structures for it. Discuss whether the English sentence has two distinct interpretations corresponding to the two trees.
C. Give a sentence (using only the words from this grammar) that is not covered by this grammar but which is nonetheless well-formed in English.
D. Explain what prevents the example in (C) from being covered.
E. Give a sentence sanctioned by this grammar that is not a well-formed English sentence.
F. Discuss how the grammar might be revised to correctly exclude your example in (E), without simultaneously excluding good sentences. Be explicit about how you would change the rules and/or the lexicon.
G. How many sentences does this grammar admit?
H. How many would it admit if it didn't have the last rule (the coordination schema)?

Problem 2: Structural Ambiguity
Show that the grammar in (23) can account for the ambiguity of each of the following sentences by providing at least two trees licensed for each one, and explain briefly which interpretation goes with which tree:

(i) Bo saw the group with the telescope.
(ii) Most dogs and cats with fleas live in this neighborhood.
(iii) The pictures show Superman and Lois Lane and Wonder Woman.

[*Note: We haven't provided a lexicon, so technically, (23) doesn't generate any of these. You can assume, however, that all the words in them are in the lexicon, with the obvious category assignments.*]

Problem 3: Infinity
The grammar in (23) has two mechanisms, each of which permits us to have infinitely many sentences: the Kleene operators (plus and star), and recursion (categories that can 'dominate themselves'). Construct arguments for why we need both of them. That is, why not use recursion to account for the unboundedness of coordination or use Kleene star to account for the possibility of arbitrary numbers of PPs?

[*Hint: Consider the different groupings into phrases – that is, the different tree structures – provided by the two mechanisms. Then look for English data supporting one choice of structure over another.*]

Problem 4: CFG for Japanese

Examples (i)–(x) give examples of grammatical Japanese sentences and strings made up of the same words which are not grammatical Japanese sentences.

(i) Suzuki-san-ga sono eiga-wo mita.
Suzuki-NOM that movie-ACC saw
'Suzuki saw that movie.'

(ii)*Mita Suzuki-san-ga sono eiga-wo.
Saw Suzuki-NOM that movie-ACC

(iii)*Suzuki-san-ga mita sono eiga-wo.
Suzuki-NOM saw that movie-ACC

(iv)*Suzuki-san-ga eiga-wo sono mita.
Suzuki-NOM movie-ACC that saw.

(v) Suzuki-san-ga sono omoshiroi eiga-wo mita.
Suzuki-NOM that interesting movie-ACC saw
'Suzuki saw that interesting movie.'

(vi)*Suzuki-san-ga sono eiga-wo omoshiroi mita.
Suzuki-NOM that movie-ACC interesting saw

(vii)*Suzuki-san-ga omoshiroi sono eiga-wo mita.
Suzuki-NOM interesting that movie-ACC saw

(viii) Suzuki-san-ga Toukyou e itta.
Suzuki-NOM Tokyo to went.
'Suzuki went to Tokyo.'

(ix)*Suzuki-san-ga e Toukyou itta.
Suzuki-NOM to Tokyo went.

(x)*Suzuki-san-ga itta Toukyou e.
Suzuki-NOM went Tokyo to.

A. Using the lexicon in (xi), write phrase structure rules that will generate the grammatical examples and correctly rule out the ungrammatical examples.

[*Notes: The data presented represent only a very small fragment of Japanese, and are consistent with many different CFGs. While some of those CFGs would fare better than others when further data are considered, any answer that accounts for the data presented here is acceptable. The abbreviations* 'NOM' *and* 'ACC' *in these examples stand for nominative and accusative case, which you may ignore for the purposes of this problem.*]

(xi) N: Suzuki-san-ga, eiga-wo, Toukyou
D: sono
P: e
A: omoshiroi
V: mita, itta

B. Draw the trees that your grammar assigns to (i), (v), and (viii).

Problem 5: Properties Common to Verbs

The rules in (34) embody the claim that IVs, TVs, and DTVs are entirely different categories. Hence, the rules provide no reason to expect that these categories would have more in common than any other collection of three lexical categories, say, N, P, and D. But these three types of verbs do behave alike in a number of ways. For example, they all exhibit agreement with the subject of the sentence, as discussed in Section 2.7.3. List at least three other properties that are shared by intransitive, transitive, and ditransitive verbs.

Problem 6: Pronoun Case

There are some differences between the noun phrases that can appear in different positions. In particular, pronouns in subject position have one form (referred to as NOMINATIVE, and including the pronouns *I, he, she, we*, and *they*), whereas pronouns in other positions take another form (called ACCUSATIVE, and including *me, him, her, us*, and *them*). So, for example, we say *He saw her*, not **Him saw she.*

A. How would the category of NP have to be further subdivided (that is, beyond NP-SG and NP-PL) in order to account for the difference between nominative and accusative pronouns?

B. How would the rules for S and the various kinds of VPs have to be modified in order to account for the differences between where nominative and accusative pronouns occur?

3

Analyzing Features of Grammatical Categories

3.1 Introduction

In the last chapter, we saw that there are constraints on which words can go together (what linguists call CO-OCCURRENCE RESTRICTIONS) that are not adequately described using the standard formalism of context-free grammar. Some verbs must take an object; others can never take an object; still others (e.g. *put*, *hand*) require both an object and another phrase of a particular kind. These co-occurrence restrictions, as we have seen, give rise to a great deal of redundancy in CFGs. In addition, different forms of a given verb impose different conditions on what kind of NP can precede them (i.e. on what kind of subject they co-occur with). For example, *walks* requires a third-person singular NP as its subject; *walk* requires a plural subject, or else one that is first- or second-person singular. As we saw in the last chapter, if we try to deal with this complex array of data by dividing the category V into more specific categories, each with its unique co-occurrence restrictions, we end up with a massively redundant grammar that fails to capture linguistically significant generalizations.

We also isolated a second defect of CFGs, namely that they allow rules that are arbitrary. Nothing in the theory of CFG reflects the headedness of phrases in human language – that is, the fact that phrases usually share certain key properties (nounhood, verbhood, prepositionhood, etc.) with a particular daughter within them.We must somehow modify the theory of CFG to allow us to express the property of headedness.

Our solution to the problem of redundancy is to make grammatical categories decomposable into component parts. CFG as presented so far treats each grammatical category symbol as ATOMIC – that is, without internal structure. Two categories are either identical or different; there is no mechanism for saying that two categories are alike in some ways, but different in others. However, words and phrases in natural languages typically behave alike in certain respects, but not in others. For example, the two words *deny* and *denies* are alike in requiring an NP object (both being forms of a transitive verb). But they differ in terms of the kind of subject NP they take: *denies* requires a third-person-singular subject like *Kim* or *she*, while *deny* accepts almost any NP subject except the third-person-singular kind. On the other hand, *denies* and *disappears* both take a singular subject NP, but only the former can co-occur with a following object NP. In other words, the property of taking a third-person-singular subject is independent of the property of taking a direct object NP. This is illustrated in the following table:

(1)

	3rd singular subject	plural subject
direct object NP	*denies*	*deny*
no direct object NP	*disappears*	*disappear*

The table in (1) illustrates only two of the cross-cutting properties of verbs. There are many more. For example, the properties of forming the third-person-singular form with *-s*, the past tense form with *-ed*, and the present participle with *-ing* are all orthogonal to the property of taking a direct object NP. In Chapter 8, we will see how to write rules for generating these INFLECTIONAL forms of verbs. In order to write such rules with maximal generality, we need to be able to refer to the class of all verbs, regardless of whether they take a direct object NP. More generally, an adequate theory of grammar needs to be able to categorize words into classes defined in terms of cross-cutting properties. In Chapter 2, we showed CFG to be inadequate as a theory of grammar, because it provides no means to represent cross-cutting properties. Instead, it ends up proliferating atomic categories and missing generalizations.

To accommodate these observations, we will develop the view that grammatical categories are not atomic, but rather are COMPLEXES of grammatical properties. In some ways, this innovation is similar to the periodic table of the elements in chemistry, which represents the elements as complexes of physical properties. The rows and columns of the table represent classes of elements that have properties in common, and the classes intersect: each element belongs to more than one class, and shares only some of its properties with the other elements in each of the classes it belongs to. Treating grammatical categories as complexes of grammatical properties will also pave the way for a solution to the second defect of CFGs, by allowing us to express the property of headedness.

3.2 Feature Structures

This section introduces the formal mechanism we will use for representing grammatical categories as complexes of grammatical properties. But let us first review the grammatical properties we have covered so far. We have seen that verbs differ in their transitivity. In fact, this kind of variation is not restricted to verbs. More generally, linguists talk about elements that have different combinatoric potential in terms of differing VALENCE.[1] Likewise, we talk of the NUMBER (singular or plural) of a noun, the PART OF SPEECH of a word (whether it's a noun, verb, etc.), and the FORM of a verb (e.g. whether it is a present participle, an infinitive, etc.). Previously we have been associating each word in the lexicon with a single atomic category (such as P, N-SG, etc.). Now, in order to model grammatical categories as complexes of information, we will use FEATURE STRUCTURES instead of atomic labels.

A feature structure is a way of representing grammatical information. Formally, a feature structure consists of a specification of a set of features (which we will write in upper case), each of which is paired with a particular value. Feature structures can be

[1] This term, borrowed from chemistry, refers to the capacity to combine with atoms, ions, and the like.

thought of in at least two roughly equivalent ways. For example, they may be conceived of as functions (in the mathematicians' sense of the word)[2] specifying a value for each of a set of features, or else as directed graphs where feature names label arcs that point to appropriately labeled nodes. For grammatical purposes, however, it will be most useful for us to focus on DESCRIPTIONS of feature structures, which we will write in a square bracket notation, as shown in (2):

(2) $$\begin{bmatrix} \text{FEATURE}_1 & \text{VALUE}_1 \\ \text{FEATURE}_2 & \text{VALUE}_2 \\ \quad\ldots & \\ \text{FEATURE}_n & \text{VALUE}_n \end{bmatrix}$$

For example, we might treat the category of the word *bird* in terms of a feature structure that specifies just its part of speech and number. We may assume such a category includes appropriate specifications for two appropriately named features: its part of speech (POS) is noun, and its number (NUM) is singular (sg). Under these assumptions, the lexical entry for *bird* would be a pair consisting of a form and a feature structure description, roughly as shown in (3):[3]

(3) $$\left\langle \text{bird}\ ,\ \begin{bmatrix} \text{POS} & \text{noun} \\ \text{NUM} & \text{sg} \end{bmatrix} \right\rangle$$

One of the first things we will want to do in developing a theory of grammar is to classify linguistic entities in various ways. To this end, it is particularly useful to introduce the notion of TYPE. This concept is really quite simple: if we think of a language as a system of linguistic entities (words, phrases, categories, sounds, and other more abstract entities that we will introduce as we go along), then types are just classes of those entities. We assign entities to these classes on the basis of certain properties that they share. Naturally, the properties we employ in our type classification will be those that we wish to refer to in our descriptions of the entities. Thus each grammatical type will be associated with particular features and sometimes with particular values for those features. As we develop our theory of grammatical types, we will in fact be developing a theory of what kinds of linguistic entities there are, and what kinds of generalizations hold of those entities.

Let us make this very abstract discussion more concrete by considering the use of feature structures to elucidate a simple nonlinguistic domain: universities and the people who are associated with them. We'll start from the assumption that the people and the other entities are really 'out there' in the real world. Our first step then in constructing a theory of this part of the world is to develop a model. A simple model will be a set of mathematical entities that we assume to correspond to the real ones. Our theory will be successful to the extent that we can show that the properties that our theory ascribes to our modeling entities (through stipulation or deduction from the stipulations) also hold

[2]A function in this sense is a set of ordered pairs such that no two ordered pairs in the set share the same first element. What this means for feature structures is that each feature in a feature structure must have a unique value.

[3]Throughout this book, we will describe linguistic forms in terms of standard English orthography. In fact, a lexical entry such as this should contain a phonological description that will play a role in the word's phonological realization, a topic we will not consider in detail here.

of the real world entities that they correspond to.

The domain at hand includes entities such as universities, departments and individuals (people). We might want to talk about certain properties of these entities, for example their names or telephone numbers. In order to do so, we will start to build our model by declaring the existence of a general type called *entity* and say that the features NAME and TEL(EPHONE) are appropriate features for all entities (tokens) of this type. So for each university, department, or person in this university world, we would hypothesize a distinct feature structure model that we could describe as follows:

(4) a. $\begin{bmatrix} \textit{entity} & \\ \text{NAME} & \text{Stanford University} \\ \text{TEL} & \text{650-723-2300} \end{bmatrix}$

b. $\begin{bmatrix} \textit{entity} & \\ \text{NAME} & \text{John Hennessy} \\ \text{TEL} & \text{650-723-2481} \end{bmatrix}$

c. $\begin{bmatrix} \textit{entity} & \\ \text{NAME} & \text{Stanford Linguistics} \\ \text{TEL} & \text{650-723-4284} \end{bmatrix}$

Note that we use type names (in this case *entity*), written in italics, as labels on the top line within feature structures.

Of course 'entity' is a very general classification – our theory would not have progressed far if it recognized no more specific kinds of things. So in fact, we would want our theory to include the fact that there are different subtypes of the type *entity*. Let's call these new types *university*, *department*, and *individual*. Entities belonging to each of these types have their own special properties. For example, individual people have birthdays, but universities and departments don't (or not in the same sense). Similarly, departments have chairs (or 'heads of department'), but neither universities nor individuals do. And only universities have presidents. Finally, universities and deparments, but not individuals, have founders, a fact that will motivate grouping these two types together under a common intermediate-level type which we will call *organization*. We can then accommodate all these facts by declaring each of the relevant features (BIRTHDAY, CHAIR, PRESIDENT, FOUNDERS) to be appropriate for entities of the appropriate subtype. This organization of the types of entity and the features that are appropriate for each of them results in the TYPE HIERARCHY shown in (5):

(5)
entity
[NAME, TEL]

organization [FOUNDERS] — *individual* [BIRTHDAY]

university [PRESIDENT] — *department* [CHAIR]

Each type of entity has its own constellation of features – some of them are declared appropriate for entities of the indicated type; others are sanctioned by one of the supertypes: *entity* or *organization*. This is a simple illustration of how a hierarchical classification system works. A given feature structure contains only those features that are declared appropriate by one of its types, that is, by its LEAF type[4] or one of its supertypes in a hierarchy like (5). This formal declaration is just a precise way of saying that the members of the relevant subclasses have certain properties that distinguish them from other entities in the system, as well as certain properties that they share with other entities.

Now that we've extended the model by adding types and features, the resulting descriptions that we write will be appropriately more specific, as in (6):

(6) a.
$$\begin{bmatrix} \textit{university} & \\ \text{NAME} & \text{Stanford University} \\ \text{FOUNDERS} & \left\langle \text{Leland Stanford , Jane Stanford} \right\rangle \\ \text{PRESIDENT} & \text{John Hennessy} \\ \text{TEL} & \text{650-723-2300} \end{bmatrix}$$

b.
$$\begin{bmatrix} \textit{individual} & \\ \text{NAME} & \text{John Hennessy} \\ \text{BIRTHDAY} & \text{9-22-52} \\ \text{TEL} & \text{650-723-2481} \end{bmatrix}$$

c.
$$\begin{bmatrix} \textit{department} & \\ \text{NAME} & \text{Stanford Linguistics} \\ \text{FOUNDERS} & \left\langle \text{Joseph Greenberg , Charles Ferguson} \right\rangle \\ \text{CHAIR} & \text{Eve Clark} \\ \text{TEL} & \text{650-723-4284} \end{bmatrix}$$

Note that we also need to specify what kind of value is appropriate for each feature. Here we've used angled brackets ('$\langle$' and '$\rangle$') to construct a list as the value of the feature FOUNDERS. As we will see, a feature structure also inherits any type constraints, (that is, potentially complex constraints on feature values) that are associated with its supertypes. Articulating a type hierarchy and the constraints associated with each type in the hierarchy is an important component of a theory that uses typed feature structures as its models.

Let us reconsider the feature structures in (6). These structures, as explicated above, aren't yet expressing the proper information about the objects they are trying to model. In particular, the value of features like PRESIDENT and CHAIR are atomic, i.e. the names John Hennessy and Eve Clark. But this isn't right – the president of Stanford University is the individual John Hennessy, not his name. The same goes for Eve Clark, who gives more to being chair of the Stanford Linguistics Department than just her name.

[4]The leaf types are the basic or bottom-level types in a hierarchy, i.e. the types that have no subtypes. These are also referred to in the literature (somewhat counterintuitively) as 'maximal' types.

Similarly, the value of the FOUNDERS feature should be a list of individuals, not a list of names. To reflect these observations, we now introduce complex feature structures, those whose features may have nonatomic feature structures (or lists of feature structures) as their value, where appropriate. This modification leads to the following more accurate models of Stanford and its Linguistics Department (the model of John Hennessy remains unaffected by this change):

(7) a.
$$\begin{bmatrix} \textit{university} & \\ \text{NAME} & \text{Stanford University} \\ \text{FOUNDERS} & \left\langle \begin{bmatrix} \textit{individual} & \\ \text{NAME} & \text{Leland Stanford} \end{bmatrix}, \begin{bmatrix} \textit{individual} & \\ \text{NAME} & \text{Jane Stanford} \end{bmatrix} \right\rangle \\ \text{PRESIDENT} & \begin{bmatrix} \textit{individual} & \\ \text{NAME} & \text{John Hennessy} \\ \text{BIRTHDAY} & \text{9-22-52} \\ \text{TEL} & \text{650-723-2481} \end{bmatrix} \\ \text{TEL} & \text{650-723-2300} \end{bmatrix}$$

b.
$$\begin{bmatrix} \textit{department} & \\ \text{NAME} & \text{Stanford Linguistics} \\ \text{FOUNDERS} & \left\langle \begin{bmatrix} \textit{individual} & \\ \text{NAME} & \text{Joseph Greenberg} \\ \text{BIRTHDAY} & \text{5-28-15} \end{bmatrix}, \begin{bmatrix} \textit{individual} & \\ \text{NAME} & \text{Charles Ferguson} \\ \text{BIRTHDAY} & \text{7-6-21} \end{bmatrix} \right\rangle \\ \text{CHAIR} & \begin{bmatrix} \textit{individual} & \\ \text{NAME} & \text{Eve Clark} \\ \text{BIRTHDAY} & \text{7-26-42} \\ \text{TEL} & \text{650-723-4284} \end{bmatrix} \\ \text{TEL} & \text{650-723-4284} \end{bmatrix}$$

When we model some empirical problem in this way, it is important to distinguish the modeling objects (the typed feature structures) from the statements we make about them. The objects in our model are meant to be simplified analogs of objects in the real world (if they weren't simplified, it wouldn't be a model). The statements we make about the modeling objects – our constraints – constitute our theory of the domain we are investigating. The system of types we set up of course is the first step in developing such a theory:

- It states what kinds of objects we claim exist (the types).
- It organizes the objects hierarchically into classes with shared properties (the type hierarchy).
- It states what general properties each kind of object has (the feature and feature value declarations).

We could summarize the beginnings of our theory of universities in terms of the following table (where 'IST' stands for 'immediate supertype'):[5]

(8)

TYPE	FEATURES/VALUES	IST
entity	$\begin{bmatrix}\text{NAME} & \textit{string} \\ \text{TEL} & \textit{number}\end{bmatrix}$	
organization	$\begin{bmatrix}\text{FOUNDERS} & \textit{list(individual)}\end{bmatrix}$	*entity*
university	$\begin{bmatrix}\text{PRESIDENT} & \textit{individual}\end{bmatrix}$	*organization*
department	$\begin{bmatrix}\text{CHAIR} & \textit{individual}\end{bmatrix}$	*organization*
individual	$\begin{bmatrix}\text{BIRTHDAY} & \textit{date}\end{bmatrix}$	*entity*

Against this background, it is the particular constraints we write that fill in the details. Type constraints specify properties that relevant classes of objects have (e.g. that universities have presidents who are individuals) and other constraints characterize properties of certain idiosyncratic entities that we find it necessary to recognize (e.g. that Stanford's president is John Hennessy). We then make the standard assumption that our modeling objects are in correspondence with the real world. In so doing, our constraints are making claims about reality in ways that distinguish our theory of the relevant empirical domain from many others that could be formulated.

Our (admittedly somewhat artificial) theory of Stanford University then consists of a set of constraints that reflect our claims about the way Stanford is, some of which may reflect the way all universities are. Those constraints are meant to describe (or be SATISFIED by) the objects in our model of Stanford – the feature structures assigned to appropriate types, exhibiting the relevant properties. And if we've modeled things correctly, our feature structures will reflect the reality of Stanford and we will view our theory as making correct predictions.

Theories often include constraints requiring two things to be identical. For example, suppose we wanted to state the hypothesis that the phone number of a department chair was always the same as the department's phone number. This somewhat trivial (yet precise) claim might be formulated as follows:

[5]Note that this table assumes the types *number*, *string* and *date*. These three types would also need to be incorporated into the type hierarchy.

(9)
$$department : \begin{bmatrix} \text{TEL} & \boxed{1} \\ \text{CHAIR} & \begin{bmatrix} \text{TEL} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

The colon here denotes a conditional ('if–then') relation between a type and a claim being made about the instances of that type. The boxed numerals in (9) are called 'tags'. They function like variables in algebra, logic, or programming languages. That is, they indicate that two values within a given feature structure are identical. What the constraint in (9) is saying then is that for any feature structure of type *department*, if you start at the outside and follow the feature path CHAIR|TEL, you'll arrive at the same value that you find when you start at the outside again and follow the (single-feature) path TEL.

Of course, it's easy to test the predictions of a one-sentence theory like (9). The feature structure models of type *department* that satisfy (9) have a clear and simple property and the relevant objects out in the real world are all listed in the Stanford Directory with their phone numbers. It's presumably not hard to verify whether (9) is true or not.[6] But science is full of theories whose predictions are much harder to test. Indeed, we'll see that evaluating the predictions of a theory of language based on feature structure models can sometimes be quite a subtle matter.

Interesting theories involve a number of different claims that interact. For this reason, it's essential that we have a way of combining constraints and determining which models satisfy the resulting combinations, however complex they might be. We will in fact use a simple method for combining (conjoining) constraints – one that we'll sometimes write with the symbol '&', as in (10a). Quite often, however, we will simply combine two constraints into a bigger one like (10b):[7]

(10) a. $\begin{bmatrix} \text{TEL} & \text{650-723-4284} \end{bmatrix} \;\&\; \begin{bmatrix} \text{NAME} & \text{Stanford Linguistics} \end{bmatrix}$

b. $\begin{bmatrix} \text{NAME} & \text{Stanford Linguistics} \\ \text{TEL} & \text{650-723-4284} \end{bmatrix}$

Notice how our constraints relate to their models. The first conjunct (the bracketed constraint before the '&' in (10a)) is satisfied by a set of feature structures (in our current model, it's the set that contains the feature structure we used to model the Stanford Linguistics Department and the one we used to model its chair). The second conjunct in (10a) is also satisfied by a set of feature structures, but this set has only one member: the feature structure serving as our model of the Stanford Linguistics Department. And the constraint in (10), whether we formulate it as in (10a) or as in (10b), is satisfied by the intersection of the two other sets, i.e. by the (singleton) set that contains just the feature structure we used to model the Stanford Linguistics Department.

[6] In fact, this theory of Stanford department phone numbers is easily falsified.

[7] The process of combining constraints in the fashion of (10b) is often called 'unification'. Theories of the sort we describe in this book are sometimes called 'unification-based', but this term is misleading. Unification is a method (i.e. a procedure) for solving sets of identity constraints. But it is the constraints themselves that constitute the theory, not any procedure we might use with them. Hence, we will refer to the theory of grammar we develop, and the class of related theories, as 'constraint-based', rather than 'unification-based'.

Note that the constraints in (11) are incompatible because they differ in the value they assign to the feature NAME:

(11) a. $\begin{bmatrix} \textit{university} & \\ \text{NAME} & \text{Stanford University} \end{bmatrix}$

b. $\begin{bmatrix} \textit{university} & \\ \text{NAME} & \text{Harvard University} \end{bmatrix}$

And because (11a) and (11b) are incompatible, they couldn't be used to describe the same entity.

Similarly, the constraints in (12) cannot be combined:

(12) a. $\begin{bmatrix} \textit{individual} & \\ \text{TEL} & \text{650-555-4284} \end{bmatrix}$

b. $\begin{bmatrix} \textit{department} & \\ \text{TEL} & \text{650-555-4284} \end{bmatrix}$

In this case, the problem is that (12a) and (12b) specify incompatible types, namely, *individual* and *department*. Hence (12a) and (12b) must be describing distinct entities. But the constraint in (13) is compatible with any of those in (14a)–(14c):

(13) $\begin{bmatrix} \text{TEL} & \text{888-234-5789} \end{bmatrix}$

(14) a. $\begin{bmatrix} \textit{university} \end{bmatrix}$

b. $\begin{bmatrix} \textit{individual} & \\ \text{NAME} & \text{Sailor Moon} \end{bmatrix}$

c. $\begin{bmatrix} \textit{department} & \\ \text{NAME} & \text{Metaphysics} \\ \text{CHAIR} & \text{Alexius Meinong, Jr.} \end{bmatrix}$

For example, the combination of (13) and (14b), shown in (15), is satisfied by those objects (in our model) that satisfy both (13) and (14b):

(15) $\begin{bmatrix} \textit{individual} & \\ \text{NAME} & \text{Sailor Moon} \\ \text{TEL} & \text{888-234-5789} \end{bmatrix}$

Finally, the constraints in (16) cannot be combined:

(16) a. $\begin{bmatrix} \text{BIRTHDAY} & \text{10-10-1973} \end{bmatrix}$

b. $\begin{bmatrix} \text{PRESIDENT} & \begin{bmatrix} \textit{individual} & \\ \text{NAME} & \text{Sailor Moon} \end{bmatrix} \end{bmatrix}$

In this case, the constraints cannot be combined because there is no type for which the features BIRTHDAY and PRESIDENT are appropriate. Since all of the modeling objects must belong to some type, there will be none that satisfy both (16a) and (16b).

When our feature structure constraints get a bit more complicated, we will sometimes want to indicate simultaneously the value of a particular feature and the fact that that value is identical with the value of another feature (or feature path), as shown in (17):

(17) $$\begin{bmatrix} \text{TEL} & \boxed{1}\text{650-723-4284} \\ \text{CHAIR} & [\text{TEL} \quad \boxed{1}] \end{bmatrix}$$

But it would make no difference if we wrote the phone number after the other occurrence of $\boxed{1}$ in (17):

(18) $$\begin{bmatrix} \text{TEL} & \boxed{1} \\ \text{CHAIR} & [\text{TEL} \quad \boxed{1}\text{650-723-4284}] \end{bmatrix}$$

The intended interpretation would be exactly the same. It also makes no difference what order we write the features in. For example, (17) and (18) are both equivalent to either of the following:

(19) a. $$\begin{bmatrix} \text{CHAIR} & [\text{TEL} \quad \boxed{1}\text{650-723-4284}] \\ \text{TEL} & \boxed{1} \end{bmatrix}$$

b. $$\begin{bmatrix} \text{CHAIR} & [\text{TEL} \quad \boxed{1}] \\ \text{TEL} & \boxed{1}\text{650-723-4284} \end{bmatrix}$$

Finally, it should be noticed that the choice of a particular tag is also completely arbitrary. The following constraints are also equivalent to the ones in (17)–(19):

(20) a. $$\begin{bmatrix} \text{CHAIR} & [\text{TEL} \quad \boxed{279}\text{650-723-4284}] \\ \text{TEL} & \boxed{279} \end{bmatrix}$$

b. $$\begin{bmatrix} \text{TEL} & \boxed{\aleph} \\ \text{CHAIR} & [\text{TEL} \quad \boxed{\aleph}\text{650-723-4284}] \end{bmatrix}$$

These are still simple examples. In the chapters that follow, we will have occasion to combine the various tools introduced here into fairly complex constraints.

Exercise 1: Practice with Combining Constraints

Are the following pairs of constraints compatible? If so, what does the combined constraint look like?

A. $$\begin{bmatrix} \text{TEL} & \text{650-723-4284} \end{bmatrix} \;\&\; \begin{bmatrix} \textit{department} & \\ \text{NAME} & \text{Metaphysics} \end{bmatrix}$$

B. $$\begin{bmatrix} \text{TEL} & \text{650-723-4284} \end{bmatrix} \;\&\; \begin{bmatrix} \text{TEL} & \boxed{23} \\ \text{CHAIR} & [\text{TEL} \; \boxed{23}] \end{bmatrix}$$

C. $$\begin{bmatrix} \text{PRESIDENT} & \boxed{1} \\ \text{FOUNDERS} & \langle \boxed{1} \rangle \end{bmatrix} \;\&\; \begin{bmatrix} \textit{individual} & \\ \text{NAME} & \text{John Hennessy} \end{bmatrix}$$

3.3 The Linguistic Application of Feature Structures

3.3.1 Feature Structure Categories

So how do typed feature structures help us with our linguistic concerns? Instead of saying that there is just one kind of linguistic entity, which must bear a value for every feature we recognize in our feature structures, we will often want to say that a given entity is of a certain type for which only certain features are appropriate. In fact, we will use typing in many ways: for example, to ensure that [NUM sg] (or [NUM pl]) can only be specified for certain kinds of words (for example, nouns, pronouns, and verbs), but not for prepositions or adjectives.[8] Likewise, we will eventually introduce a feature AUX to distinguish auxiliaries (helping verbs like *will* and *have*) from all other verbs, but we won't want to say that nouns are all redundantly specified as [AUX –]. Rather, the idea that we'll want our grammar to incorporate is that the feature AUX just isn't appropriate for nouns. We can use types as a basis for classifying the feature structures we introduce and the constraints we place on them. In so doing, we provide an easy way of saying that particular features only go with certain types of feature structure. This amounts to the beginnings of a linguistic ontology: the types lay out what kinds of linguistic entities exist in our theory, and the features associated with those types tell us what general properties each kind of entity exhibits.[9]

In addition, the organization of linguistic objects in terms of a type hierarchy with intermediate types (analogous to *organization* in the university example) is significant. Partial generalizations – generalizations that hold of many but not all entities – are very common in the domain of natural language. Intermediate types allow us to state those generalizations. This feature of our theory will become particularly prominent when we organize the lexical entries into a hierarchy in Chapter 8.

In this chapter, we will develop a feature-based grammar that incorporates key ideas from the CFG we used in Chapter 2. We will show how feature structures can solve some of the problems we raised in our critical discussion of that grammar. As we do so, we will gradually replace all the atomic category names used in the CFG (S, NP, V, etc.) by typed feature structures. Since the grammar presented in this chapter is modeled on the CFG of Chapter 2, it is just an intermediate step in our exposition. In Chapter 4, we will refine the Chapter 3 grammar so that in the chapters to come we can systematically expand its coverage to include a much wider set of data.

3.3.2 Words and Phrases

To start with, let us draw a very intuitive distinction between two types: *word* and *phrase*. Our grammar rules (i.e. our phrase structure rules) all specify the properties of phrases;

---

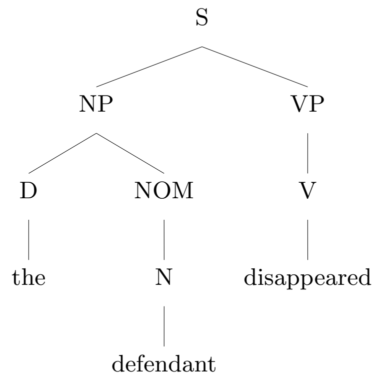
[8] Many such restrictions are language-particular. For example, adjectives are distinguished according to number (agreeing with the noun they modify) in many languages. Even prepositions exhibit agreement inflection in some languages (e.g. modern Irish) and need to be classified in similar terms.

[9] We might instead introduce some mechanism for directly stipulating dependencies between values of different features – such as a statement that the existence of a value for AUX implies that the value of POS is 'verb'. (For a theory that incorporates a mechanism like this, see Gazdar et al. 1985.) But mechanisms of this kind are unnecessary, given the availability of types in our theory.

the lexicon provides a theory of words.

Consider the CFG tree in (21):

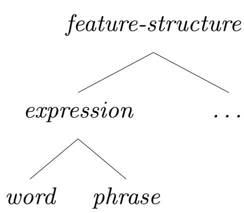
(21)



In this tree, the nodes S, NP, NOM, and VP are all *phrase*s. The nodes D, N and V are all *word*s. Both of these statements may seem unintuitive at first, because the words *word* and *phrase* are used in various ways. Sometimes a particular form, e.g. *the*, *defendant* or *disappeared*, is referred to as a word and certain sequences of forms, e.g. *the defendant* are called phrases. In the sense we intend here, however, 'word' refers to the category that the lexicon associates with a given form like *disappeared* and 'phrase' refers to the category that the grammar associates with a sequence of such forms.

Although there is an intuitive contrast between *word*s and *phrase*s, they also have some properties in common, especially in contrast to the more abstract grammatical types we will be positing below. We will therefore create our type hierarchy so that *word* and *phrase* are both subtypes of *expression*:[10]

(22)



One property that words and phrases have in common is part of speech. In the CFG of Chapter 2, this similarity was represented mnemonically (although not formally) in the atomic labels we choose for the categories: NP and N have in common that they are essentially nominal, VP and V that they are essentially verbal, etc. With feature structures, we can represent this formally. We will assume that all *expression*s specify values for a feature we will call HEAD. The value of HEAD will indicate the expression's part of speech. This feature is called HEAD because the part of speech of a phrase depends on the part of speech of one particular daughter, called the head daughter. That is, an NP structure is nominal because it has an N inside of it. That N is the head daughter of the NP structure.

[10]Note that the most general type in our theory will be called *feature-structure*. All of the types we introduce will be subtypes of *feature-structure*. If we were to fully flesh out the university example, something similar would have to be done there.

So far, then, our feature structure representation of the category NP looks like this:

(23)
$$\begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \textit{noun} \end{bmatrix}$$

and our feature structure representation of the lexical entry for a noun, say *bird*, looks like this:

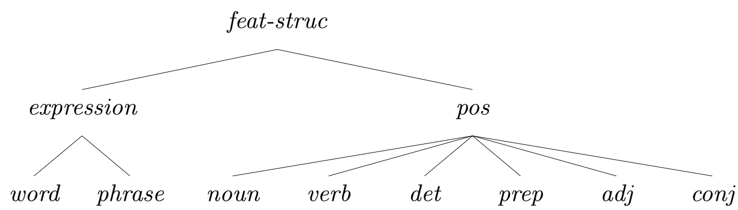
(24)
$$\left\langle \text{bird} \ , \ \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{noun} \end{bmatrix} \right\rangle$$

3.3.3 Parts of Speech

Let us reflect for a moment on parts of speech. There are certain features that are appropriate for certain parts of speech, but not others. We proposed above to distinguish helping verbs from all others in terms of the feature AUX(ILIARY), which will be appropriate only for verbs. Likewise, we will use the feature AGR(EEMENT) only for nouns, verbs, and determiners. To guarantee that only the right features go with the right parts of speech, we will introduce a set of types. Then we can declare feature appropriateness for each part of speech type, just as we did in our type hierarchy for Stanford University.

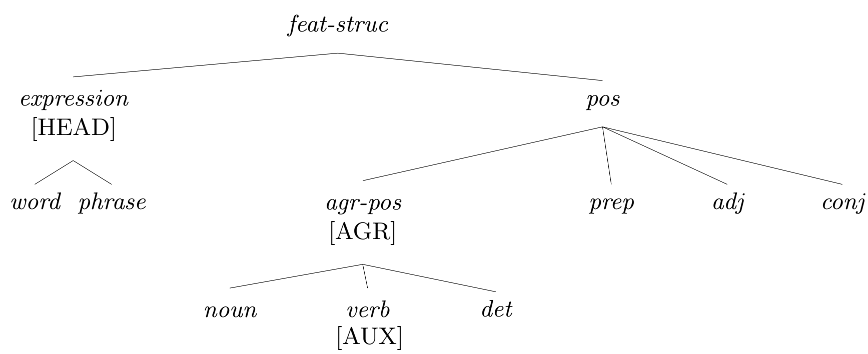
We therefore introduce the types *noun, verb, adj, prep, det*, and *conj* for the six lexical categories we have so far considered. We then make all of these subtypes of a type called *part-of-speech* (*pos*), which is itself a subtype of *feat*(*ure*)-*struc*(*ture*). The resulting type organization is as shown in (25):

(25)



But in fact, if we want to introduce features only once in a given type hierarchy, then we will have to modify this picture slightly. That's because there are three parts of speech that take the feature AGR.[11] We will thus modify the type hierarchy to give these three types a common supertype where the feature AGR is introduced, as shown in (26):

(26)



[11] There will be a few more as we expand the coverage of our grammar in later chapters.

In this way, determiners and nouns will both specify values for AGR and verbs will specify values for both AGR and AUX. Notice, however, that it is not the words themselves that specify values for these features – rather, it is the feature structures of type *noun*, *verb* or *det*. Individual words (and phrases) get associated with this information because they have a feature HEAD whose value is always a feature structure that belongs to some subtype of *pos*.

So far, we have motivated distinguishing the different subtypes of *pos* as a way of making sure that words only bear features that are appropriate for their part of speech. There is, however, another benefit. As discussed in Section 3.3.2 above, the value of the HEAD feature represents information that a phrase (more precisely, the mother nodes of a phrase structure) shares with its head daughter. (We will see how the grammar enforces this identity in Section 3.3.5 below.) The features we posit for the *pos* types (so far, AGR and AUX) also encode information that phrases share with their head daughters. This is particularly clear in the case of agreement: just as an NP is only nominal because it has an N inside of it, a singular NP is only singular because it has a singular N inside of it. By making AGR a feature of (the relevant subtypes of) *pos*, we can represent this very efficiently: we identify the HEAD value of the mother (say, NP) and that of its head daughter (N). In doing so, we identify not only the mother and head daughter's part of speech, but also any other associated information, for example, their number.[12] In refining our account of the feature structures of type *pos*, we will thus be formulating a general theory of what features the head daughter shares with its mother in a headed phrase.

3.3.4 Valence Features

The approach we are developing also provides a more satisfying analysis of our earlier categories IV, TV, and DTV. Instead of treating these as unanalyzable (i.e. as atoms), we now decompose these as feature structures. To do this, we introduce a new feature VAL (for 'valence'). The value of VAL is a feature structure (of type *val-cat*) representing the combinatoric potential of the word or phrase. The first feature we will posit under VAL is COMPS (for 'complements' – see Chapter 2, Section 2.7), which we use to indicate what the required following environment is for each type of verb: (For now, we assume that the possible values of COMPS are itr = intransitive, str = strict-transitive, and dtr = ditransitive, though we will revise this in the next chapter.)

(27)

$$\text{IV} = \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{itr} \end{bmatrix} \end{bmatrix} \qquad \text{TV} = \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{str} \end{bmatrix} \end{bmatrix}$$

$$\text{DTV} = \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{dtr} \end{bmatrix} \end{bmatrix}$$

[12] We will return to the feature AGR and describe what kinds of things it takes as its value in Section 3.3.6 below. AUX will be taken up in Chapter 13.

The three categories described in (27) all share the type *word* and the feature specification [HEAD *verb*]. This is just the combination of types and features that we would naturally identify with the category V. And by analyzing categories in terms of types and features, we can distinguish between the different valence possibilities for verbs, while still recognizing that all verbs fall under a general category. The general category V is obtained by leaving the value of the VAL feature unspecified, as in (28):

(28) $$\mathrm{V} \; = \begin{bmatrix} \textit{word} & \\ \mathrm{HEAD} & \textit{verb} \end{bmatrix}$$

The term UNDERSPECIFICATION is commonly used in linguistics to indicate a less specific linguistic description. Given our modeling assumptions, underspecification has a precise interpretation: an underspecified description (or constraint) always picks out a larger class of feature structures than a fully specified one. In general, the less information given in a description (i.e. the more underspecified it is), the more models (feature structures) there are that will satisfy that description.

In the grammar so far, the category VP differs from the category V only with respect to its type assignment.[13] So VP is recast as the following description:

(29) $$\mathrm{VP} \; = \begin{bmatrix} \textit{phrase} & \\ \mathrm{HEAD} & \textit{verb} \end{bmatrix}$$

And the class of grammatical categories that includes just verbs and verb phrases is defined precisely by the underspecification in (30):

(30) $$\begin{bmatrix} \mathrm{HEAD} & \textit{verb} \end{bmatrix}$$

Similarly, we can reanalyze the categories N and NP as follows:

(31) $$\mathrm{N} = \begin{bmatrix} \textit{word} & \\ \mathrm{HEAD} & \textit{noun} \end{bmatrix} \quad \mathrm{NP} = \begin{bmatrix} \textit{phrase} & \\ \mathrm{HEAD} & \textit{noun} \end{bmatrix}$$

Within this general approach, we can retain all our previous categories (V, S, NP, etc.) as convenient abbreviations.

Underspecification allows us to provide compact descriptions for the sets of categories that our grammar will actually need to refer to, what linguists usually call ‘natural classes’. For example, while we couldn’t even talk about IV, DTV, and TV as one class in CFG, we can now refer to them together as *word*s that are [HEAD *verb*]. We will use the symbol V as an abbreviation for this feature structure description, but it should now be regarded merely as an abbreviated description of the class of typed feature structures just described. The same is true for N, NP, VP, etc.

Observe that the feature analysis we have just sketched does not yet accommodate the category NOM. NP and NOM are both [HEAD *noun*]. And since the COMPS value is used to indicate what the following environment must be, it is not appropriate for the distinction between NP and NOM. Recall that NOM differs from NP in that it

[13] Additional differences with respect to their VAL values will be discussed shortly. A more sweeping reanalysis of the feature composition of these categories is introduced in Chapter 4 and carried on to subsequent chapters.

does not include the determiner, which is at the beginning of the phrase. In fact, it is a straightforward matter to use features to model our three-level distinction among N, NOM, and NP. NOM is the category that includes everything in the NP except the determiner, e.g. *picture of Yosemite* in *that picture of Yosemite*. We can distinguish NOM and NP using features in much the same way that we distinguished transitive and intransitive verbs – that is, by introducing a valence feature that indicates a restriction on the possible contexts in which the category in question can appear. In this case, the feature will specify whether or not a determiner is needed. We call this feature SPR (SPECIFIER). Just as we introduced 'complement' as a generalization of the notion of object, we are now introducing 'specifier' as a generalization of the notion of determiner.

For now, we will treat SPR as having two values: [SPR −] categories need a specifier on their left; [SPR +] categories do not, either because they label structures that already contain a specifier or that just don't need one. Note that like COMPS, SPR encodes an aspect of an expression's combinatoric potential. NP and NOM are thus defined as in (32):

(32)

$$\text{NP} = \begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \quad \text{NOM} = \begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix}$$

We can also use the feature SPR to distinguish between VP and S, by treating a subject NP as the VP's specifier. That is, VP and S can be distinguished as follows:

(33)

$$\text{S} = \begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \quad \text{VP} = \begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix}$$

In calling both determiners and subject NPs specifiers, we are claiming that the relationship between subject and VP is in important respects parallel to the relationship between determiner and NOM. The intuition behind this claim is that specifiers (subject NPs and determiners) serve to complete the phrases they are in. S and NP are fully formed categories, while NOM and VP are still incomplete. The idea that subjects and determiners play parallel roles seems particularly intuitive when we consider examples like (34).

(34) a. We created a monster.
  b. our creation of a monster

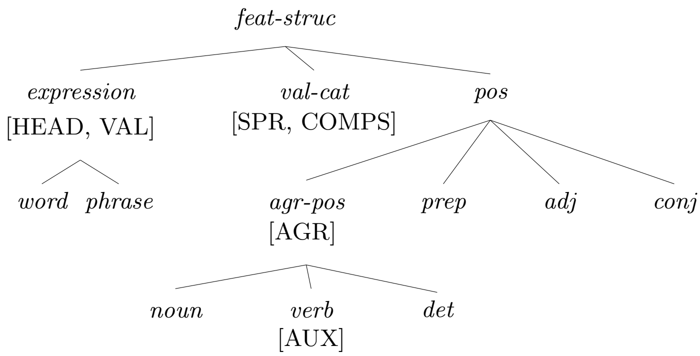
We will have more to say about the feature SPR in the next chapter.

Returning to (32), notice that we have extended the intuitive meaning of the specification [COMPS itr] so that it applies to phrases as well as to words. This is a natural extension, as phrases (whether NP, S, VP or NOM) are like strictly intransitive verbs in that they cannot combine with complements. (Recall that a phrase contains its head's complement(s), so it can't combine with any more). Notice also that under this conception, the abbreviations NP and S both include the following feature specifications:

(35)
$$\begin{bmatrix} \text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix}$$

As *word*s and *phrase*s both need to be specified for the valence features, we declare VAL to be appropriate for the type *expression*. The value of VAL is a *val-cat*, and COMPS and SPR are both features of *val-cat*.[14] Our type hierarchy now looks like this:

(36)



3.3.5 Reformulating the Grammar Rules

Turning now to the phrase structure rules considered in Chapter 2, we can reformulate our VP rules in terms of our new feature structure categories. Consider the following way of stating these rules:

(37) a.
$$\begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \boxed{1} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \rightarrow \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \boxed{1} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix}$$

b.
$$\begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \boxed{1} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \rightarrow \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \boxed{1} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{str} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \ \text{NP}$$

c.
$$\begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \boxed{1} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \rightarrow \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \boxed{1} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{dtr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \ \text{NP} \ \ \text{NP}$$

[14]In Chapter 5, we will add a further feature, MOD, to *val-cat*.

The two occurrences of $\boxed{1}$ in each of these rules tell us that the HEAD value of the mother and that of the first daughter must be identified. Since the rules in (37) were introduced as VP rules, the obvious value to assign to $\boxed{1}$ is *verb*. But, by stating the rules in this underspecified way, we can use them to cover some other structures as well. The first rule, for intransitives, can be used to introduce nouns, which can never take NP complements (in English). This is done simply by instantiating $\boxed{1}$ as *noun*, which will in turn cause the mother to be a NOM. To make this work right, we will have to specify that lexical nouns, like intransitive verbs, must be [COMPS itr]:

(38)
$$\left\langle \text{bird}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \right\rangle$$

Note that both verbs and nouns are lexically specified as [SPR −], i.e. as having not (yet) combined with a specifier.

We can now recast the CFG rules in (39):

(39) a. S → NP VP
b. NP → (D) NOM

Assuming, as we did above, that S is related to VP and V in just the same way that NP is related to NOM and N, the rules in (39) may be reformulated as (40a) and (40b), respectively:

(40) a.
$$\begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \boxed{1}\textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \rightarrow \text{NP} \begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \boxed{1} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & - \end{bmatrix} \end{bmatrix}$$

b.
$$\begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \boxed{1}\textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \rightarrow \text{D} \begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \boxed{1} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & - \end{bmatrix} \end{bmatrix}$$

In these rules, ‘NP’ and ‘D’ are abbreviations for feature structure descriptions. NP was defined in (32) above. We’ll assume that ‘D’ is interpreted as follows:

(41)
$$\text{D} = \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{det} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix}$$

Note that the feature structure rule in (40b) differs from the CFG NP rule in (39b) in that the former makes the determiner obligatory. In fact, the optionality in the CFG rule caused it to overgenerate: while some nouns (like *information* or *facts*) can appear with or without a determiner, others (like *fact*) require a determiner, and still others (like *you* or *Alex*) never take a determiner:

(42) a. I have the information.
b. I have information.
c. I was already aware of that fact.
d.*I was already aware of fact.
e. I know you.
f.*I know the you.

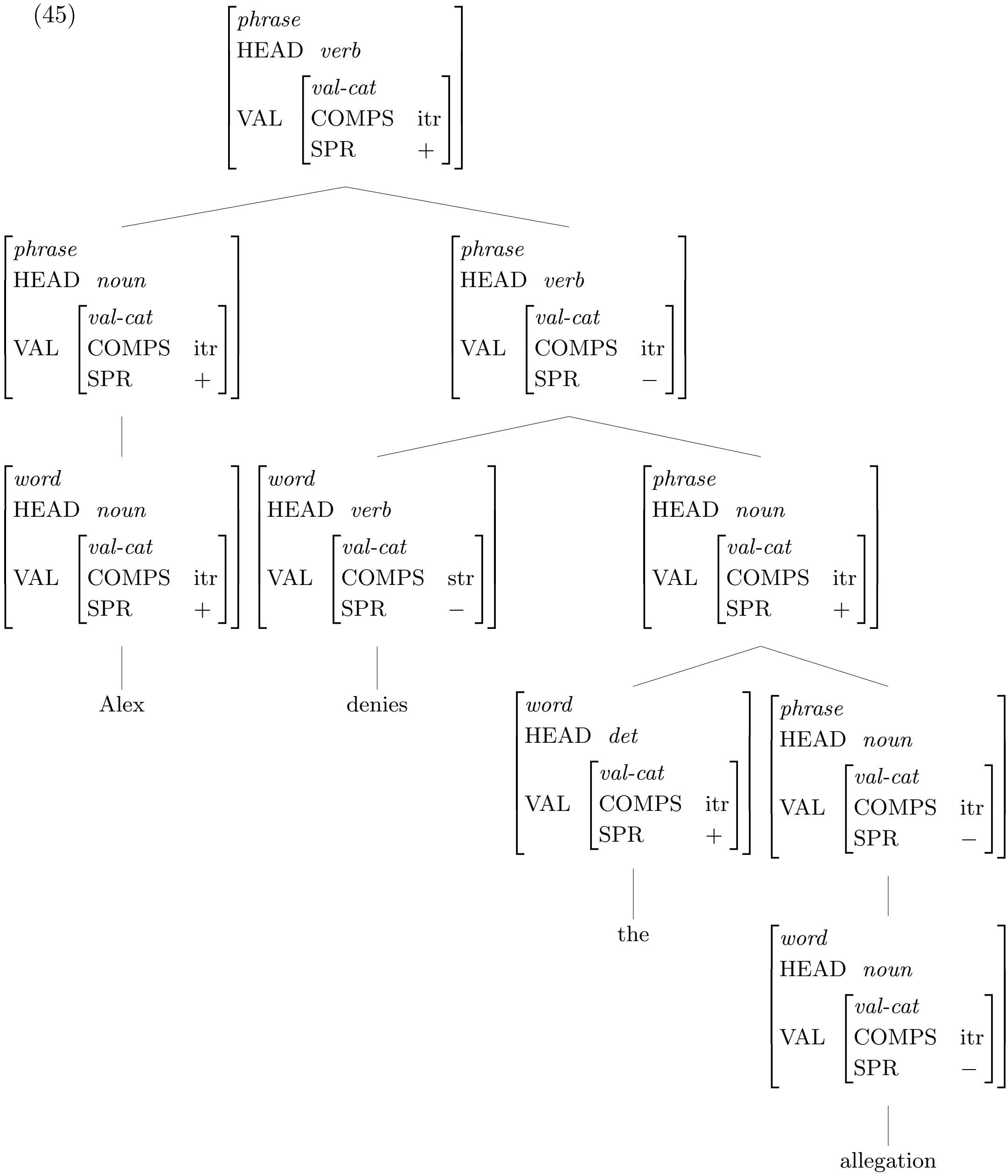
Since the CFG rule in (39b) doesn't distinguish between different kinds of Ns, it in fact licenses all of the NPs in (42). We will return to the problem of nouns whose determiners are truly optional (like *information*) in Chapter 8. The thing to note here is that the feature SPR allows us to distinguish nouns that require determiners (like *fact* or *bird*) from those that refuse determiners (like *you* or *Alex*). The former are specified as [SPR −], and build NPs with the rule in (40b). The latter are [SPR +] (see (43)), and require a new rule, given in (44):

(43)
$$\left\langle \text{Alex}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \right\rangle$$

(44)
$$\begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \boxed{1}\textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \rightarrow \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \boxed{1} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & + \end{bmatrix} \end{bmatrix}$$

Given the rules and categories just sketched, it is important to see that our grammar now licenses trees like the one shown in (45):

(45)



Exercise 2: Understanding Tree (45)

A. For each node in (45) other than the preterminal nodes, identify the rule that licensed it.

B. Find the right abbreviation (e.g. NP, S, ...) for each node in (45).

Two rules we haven't yet reconsidered are the ones that introduce PP modifiers, repeated in (46):

(46) a. VP $\rightarrow$ VP PP
b. NOM $\rightarrow$ NOM PP

Although we will have nothing to say about the internal structure of PPs in this chapter, we would like to point out the potential for underspecification to simplify these rules, as well. Once categories are modeled as feature structures, we can replace the two CFG rules in (46) with one grammar rule, which will look something like (47):

(47)
$$\begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \boxed{2} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \rightarrow \begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \boxed{2} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & - \end{bmatrix} \end{bmatrix} \text{PP}$$

Note that the head daughter of this rule is unspecified for COMPS. In fact, all of the categories of type *phrase* licensed by our grammar are [COMPS itr], so so specifying the head daughter's type as *phrase* is sufficient to get the effect of (46a,b) without adding a COMPS value.

Exercise 3: COMPS Value of Phrases
Look at the grammar summary in Section 3.6 and verify that this last claim is true.

In the next chapter, we will carry the collapsing of phrase structure rules even further. First, however, let us examine how features can be used in the analysis of agreement.

3.3.6 Representing Agreement with Features

In Section 3.3.3 above, we stated that the types *noun*, *verb* and *det* bear a feature AGR. In this section, we will consider what the value of that feature should be and how it can help us model subject-verb agreement.[15]

Agreement in English involves more than one kind of information. For subject-verb agreement, both the person and the number of the subject are relevant. Therefore, we want the value of AGR to be a feature structure that includes (at least) these two kinds of information, i.e. bears at least the features PER(SON) and NUM(BER). We will call the type of feature structure that has these features an *agr-cat* (*agreement-category*). The type *agr-cat* is a subtype of *feature-structure*.[16] The values of PER and NUM are atomic. The values of PER are drawn from the set {1st, 2nd, 3rd} and the values for NUM from the set {sg, pl}. The result is that instances of the type *agr-cat* will look like (48):

(48)
$$\begin{bmatrix} \textit{agr-cat} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{sg} \end{bmatrix}$$

[15] Determiner-noun agreement will be addressed in Problem 3 and then brought up again in Chapter 4.

[16] See the grammar summary in Section 3.6 for how this addition affects the type hierarchy.

AGR is a feature of (certain) subtypes of *pos*. This means that it is a HEAD FEATURE, i.e. one of the features that appears inside the HEAD value. Consequently, AGR-specifications get passed up from words to phrases and then to larger phrases. For example, the mother node in (49) will have the same specification for AGR as its head daughter:

(49)

$$\begin{array}{c}
\left[\begin{array}{ll}
\textit{phrase} & \\
\text{HEAD} & \left[\begin{array}{ll}\textit{noun} & \\ \text{AGR} & \left[\begin{array}{ll}\textit{agr-cat} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{sg}\end{array}\right]\end{array}\right] \\
\text{VAL} & \left[\begin{array}{ll}\textit{val-cat} & \\ \text{COMPS} & \text{itr} \\ \text{SPR} & +\end{array}\right]
\end{array}\right] \\
| \\
\left[\begin{array}{ll}
\textit{word} & \\
\text{HEAD} & \left[\begin{array}{ll}\textit{noun} & \\ \text{AGR} & \left[\begin{array}{ll}\textit{agr-cat} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{sg}\end{array}\right]\end{array}\right] \\
\text{VAL} & \left[\begin{array}{ll}\textit{val-cat} & \\ \text{COMPS} & \text{itr} \\ \text{SPR} & +\end{array}\right]
\end{array}\right] \\
| \\
\text{Alex}
\end{array}$$

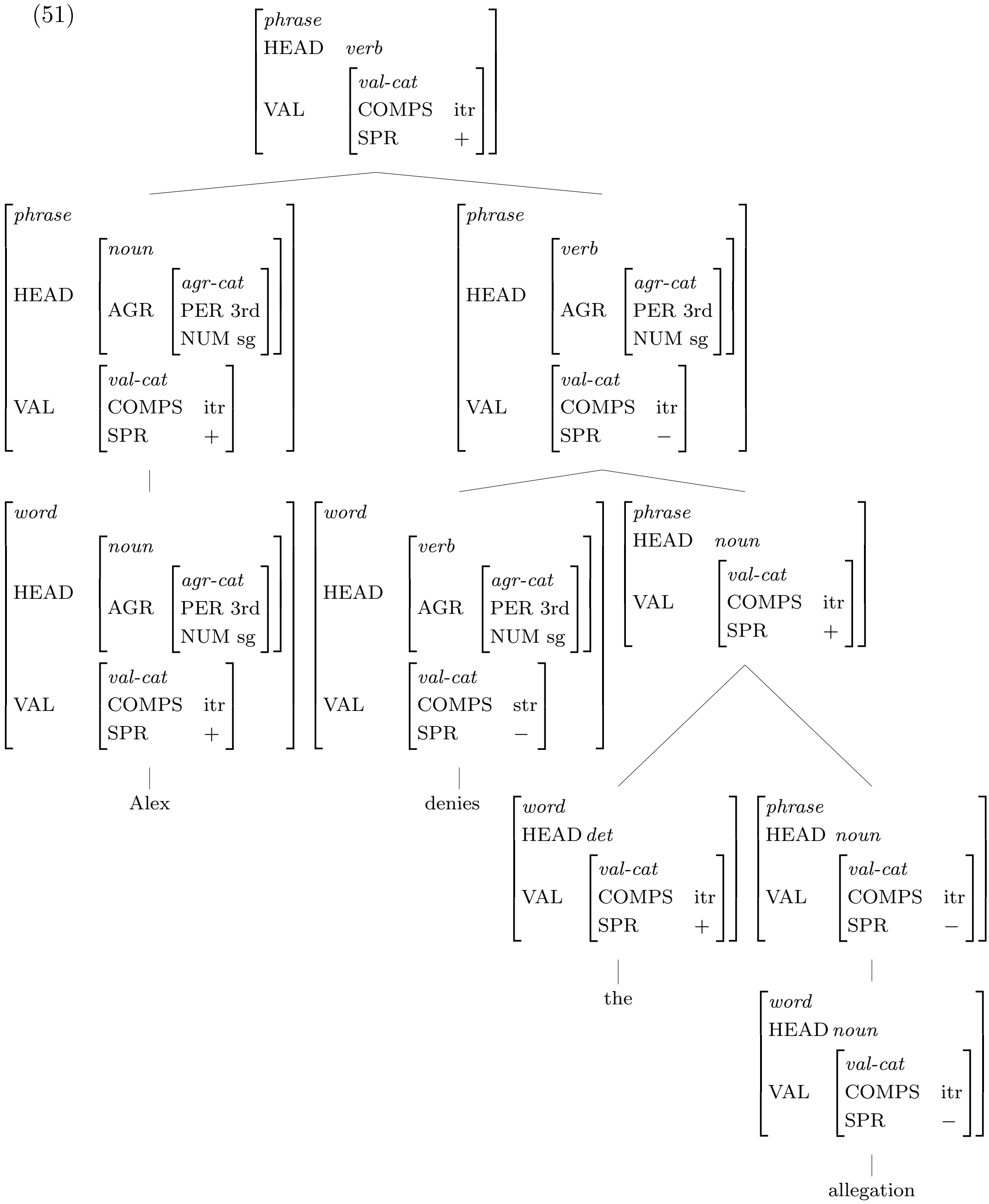
We want AGR information to be part of a phrase like this, because it is the kind of phrase that can be the subject of a simple sentence. If the verb within the VP and the noun that is the head of the subject NP both pass up their AGR specifications in this way, it is a simple matter to account for subject-verb agreement by revising our rule (40a) for combining NP and VP into an S. This revision may take the following form:

(50)

$$\left[\begin{array}{ll}
\textit{phrase} & \\
\text{HEAD} & \boxed{1}\textit{verb} \\
\text{VAL} & \left[\begin{array}{ll}\text{COMPS} & \text{itr} \\ \text{SPR} & +\end{array}\right]
\end{array}\right]
\rightarrow
\begin{array}{c}\text{NP} \\ \left[\text{HEAD} \; \left[\text{AGR} \;\; \boxed{2}\right]\right]\end{array}
\left[\begin{array}{ll}
\textit{phrase} & \\
\text{HEAD} & \boxed{1}\left[\text{AGR} \;\; \boxed{2}\right] \\
\text{VAL} & \left[\begin{array}{ll}\text{COMPS} & \text{itr} \\ \text{SPR} & -\end{array}\right]
\end{array}\right]$$

And in consequence of the revision in (50), AGR values are constrained as illustrated in (51):[17]

(51)



[17]In this tree, we omit the AGR specifications on the object NP and the root node, even though the grammar will provide them.

More generally, assuming the appropriate lexical entries, the revised analysis correctly accounts for all the contrasts in (52):

(52) a. The defendant denies the allegation.
b. *The defendant deny the allegation.
c. The defendants deny the allegation.
d. *The defendants denies the allegation.
e. The defendant walks.
f. *The defendant walk.
g. The defendants walk.
h. *The defendants walks.

Representing categories as complexes of features enables us to capture these facts without proliferating grammar rules. This is a distinct improvement over the CFG of Chapter 2.

3.3.7 The Head Feature Principle

The grammar rules proposed in the previous sections ((37a–c), (40), and (47)) have all identified the mother's HEAD value with the HEAD value of one of the daughters. The relevant HEAD-sharing daughter is always the one we have been referring to as the head daughter: the N in a NOM phrase, the NOM in an NP, the V in a VP, the VP in an S, and the VP or NOM that co-occurs with a PP modifier. But our theory does not yet include any notion of head daughter. If it did, we could factor out a general constraint about identity of HEAD values, instead of stating the same constraint in each of our five rules (with possibly more to come). The purpose of this section is to propose a general principle with this effect.

Rather than stipulating identity of features in an ad hoc manner on both sides of the rules, our analysis will recognize that in a certain kind of phrase – a HEADED PHRASE – one daughter is assigned special status as the HEAD DAUGHTER. Once this notion is incorporated into our theory (thus providing a remedy for the second defect of standard CFGs noted in the last chapter), we can factor out the identity constraint that we need for all the headed phrases, making it a general principle. We will call this generalization the Head Feature Principle (HFP).

Certain rules introduce an element that functions as the head of the phrase characterized by the rule. We will call such rules HEADED RULES. To indicate which element introduced in a headed rule is the head daughter, we will label one element on the right hand side of the rule with the letter '**H**'. So a headed rule will have the following general form:[18]

(53) $[\textit{phrase}] \rightarrow \ \ldots \ \ \mathbf{H}[\,] \ \ \ldots$

So far, we have done two things: (i) we have identified the head daughter in a headed rule and (ii) we have bundled together (within the HEAD value) all the feature specifications that the head daughter must share with its mother. With these two adjustments in place, we are now in a position to simplify the grammar of headed phrases.

[18]Note that '**H**', unlike the other shorthand symbols we use occasionally (e.g. 'V' and 'NP'), does not abbreviate a feature structure in a grammar rule. Rather, it merely indicates which feature structure in the rule corresponds to the phrase's head daughter.

First we simplify all the headed rules: they no longer mention anything about identity of HEAD values:

(54) a.
$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \rightarrow \mathbf{H}\begin{bmatrix} \textit{word} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix}$$

b.
$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \rightarrow \mathbf{H}\begin{bmatrix} \textit{word} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{str} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \ \text{NP}$$

c.
$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \rightarrow \mathbf{H}\begin{bmatrix} \textit{word} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{dtr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \ \text{NP} \ \ \text{NP}$$

d.
$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \rightarrow \overset{\text{NP}}{\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AGR} & \boxed{2} \end{bmatrix} \end{bmatrix}} \ \mathbf{H}\begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \boxed{2} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & - \end{bmatrix} \end{bmatrix}$$

e.
$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \rightarrow \text{D} \ \ \mathbf{H}\begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & - \end{bmatrix} \end{bmatrix}$$

f.
$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \rightarrow \mathbf{H}\begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & + \end{bmatrix} \end{bmatrix}$$

g.
$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \rightarrow \mathbf{H}\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix} \text{SPR} & - \end{bmatrix} \end{bmatrix} \ \ \text{PP}$$

The element labeled '**H**' in the above rules is the head daughter.

Second, we state the Head Feature Principle as a general constraint governing all trees built by headed rules.

(55) Head Feature Principle (HFP)
In any headed phrase, the HEAD value of the mother and the HEAD value of the head daughter must be identical.

The HFP makes our rules simpler by factoring out those properties common to all headed phrases, and making them conditions that will quite generally be part of the trees defined by our grammar. By formulating the HFP in terms of HEAD value identity, we allow information specified by the rule, information present on the daughter or the mother, or information required by some other constraint all to be amalgamated, as long as that information is compatible.[19]

3.4 Phrase Structure Trees

At this point, we must address the general question of how rules, lexical entries and principles like the HFP interact to define linguistic structures. Our earlier discussion of this question in Chapter 2 requires some revision, now that we have introduced feature structures and types. In the case of simple context-free grammars, descriptions and structures are in simple correspondence: in CFG, each local subtree (that is, a mother node with its daughters) corresponds in a straightforward fashion to a rule of the grammar. All of the information in that local subtree comes directly from the rule. There is no reason to draw a distinction between the linguistic objects and the grammar's descriptions of them. But now that rules, lexical entries and principles like the HFP all contribute constraints (of varying degrees of specificity) that linguistic tokens must satisfy, we must take care to specify how these constraints are amalgamated and how the grammar specifies which expressions are grammatical.

3.4.1 The Formal System: an Informal Account

The distinction between descriptions and the structures they describe is fundamental. We use feature structures in our models of linguistic entities. Consider what this meant for the feature structures we used to model universities, departments and individuals. Each feature structure model was assumed to have all the properties relevant to understanding the university system; in our example, this included (for individuals) a name, a birthday, and a telephone number. The objects we took as models were thus complete in relevant respects.[20] Contrast this with descriptions of university individuals. These come in varying degrees of completeness. A description may be partial in not specifying values for every feature, in specifying only part of the (complex) value of a feature, in failing to specify a type, or in specifying nothing at all. A complete description of some entity will presumably be satisfied by only one thing – the entity in question. An empty description is satisfied by all the entities in the modeling domain. Any nonempty partial description is satisfied by some things in the modeling domain, and not by others.

Our theory of language works the same way. We use trees to model phrases and we use feature structures to model the grammatical categories that label the nodes in those trees. These models are complete (or RESOLVED) with respect to all linguistically relevant properties.[21] On the other hand, the lexical entries, grammar rules and principles are not models but rather partial descriptions of models. They thus need not be (and in

[19] The Head Feature Principle is sometimes formulated as 'percolation' of properties of lexical heads to the phrases that they 'project'. While it is often helpful to think of information as propagating up or down through a tree, this is just a metaphor. Our formulation of the generalization avoids attributing directionality of causation in the sharing of properties between phrases and their heads.

[20] Of course, a model and the thing it is a model of differ with respect to certain irrelevant properties. Our models of university individuals should omit any irrelevant properties that all such individuals presumably have, ranging from hair color to grandmothers' middle names to disposition with respect to Indian food.

[21] 'Resolvedness' is a direct consequence of our decision to define complex feature structures as total functions over a given domain of features.

fact usually aren't) fully resolved. For example, since the English word *you* is ambiguous between singular and plural, we might want to posit a lexical entry for it like the following:

(56)
$$\left\langle \text{you}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & [\text{PER}\ \ \text{2nd}] \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \right\rangle$$

This lexical entry is not complete in that it does not provide a specification for the feature NUM.[22]

Because the lexical entry is underspecified, it licenses two distinct WORD STRUCTURES (local, non-branching subtrees whose mother is of type *word*). These are shown in (57) and (58):

(57)
$$\begin{array}{c} \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \begin{bmatrix} \textit{agr-cat} & \\ \text{PER} & \text{2nd} \\ \text{NUM} & \text{sg} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \\ | \\ \text{you} \end{array}$$

(58)
$$\begin{array}{c} \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \begin{bmatrix} \textit{agr-cat} & \\ \text{PER} & \text{2nd} \\ \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \\ | \\ \text{you} \end{array}$$

Here all the appropriate features are present (the mothers' feature structures are 'totally well-typed') and each feature has a completely resolved value.[23]

---

[22]This analysis of the ambiguity of *you* won't work in later versions of our grammar, and is presented here by way of illustration only.

[23]Again, this follows from defining feature structures in terms of total functions.

The relationship of the models to the grammar becomes more intricate when we consider not only lexical entries, but also grammar rules and the one general principle we have so far. These can all be thought of as constraints. Together, they serve to delimit the class of tree structures licensed by the grammar. For example, the grammar rule in (54b) above, repeated here as (59), is a constraint that can be satisfied by a large number of local subtrees. One such subtree is given in (60):

(59)
$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \rightarrow \mathbf{H} \begin{bmatrix} \textit{word} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{str} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \ \text{NP}$$

(60)
$$\begin{array}{c}
\begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \begin{bmatrix} \textit{agr-cat} & \\ \text{PER} & \text{2nd} \\ \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \\
\diagup \qquad\qquad \diagdown \\
\begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \begin{bmatrix} \textit{agr-cat} & \\ \text{PER} & \text{2nd} \\ \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{str} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix}
\begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \begin{bmatrix} \textit{agr-cat} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix}
\end{array}$$

How many local subtrees are there that satisfy rule (59)? The answer to this question breaks down into a number of subquestions:

(61) a. How many feature structure categories can label the mother node?
b. How many feature structures categories can label the first daughter?
c. How many feature structures categories can label the second daughter?

The number of models satisfying (59) will be obtained by multiplying the answer to (61a) times the answer to (61b) times the answer to (61c), because, in the absence of other constraints, these choices are independent of one another.

Let us consider the mother node first. Here the types of the mother's and head daughter's feature structures are fixed by the rule, as are the SPR and COMPS values, but the HEAD value is left unconstrained. In the grammar developed in this chapter, we have six parts of speech. This means that there are six options for the type of the HEAD value. If we pick *noun*, *det*, or *verb*, however, we have more options, depending on the values of AGR. Given that PER has three distinct values and NUM has two, there are six possible AGR values. Hence there are six distinct HEAD values of type *noun*, six of type *det* and six of type *verb*. Given that there is only one HEAD value of type *adj*, one of type *prep* and one of type *conj*, it follows that there are exactly 21 (= (3 × 6) + 3) possible HEAD values for the mother. Since all other feature values are fixed by the rule, there are then 21 possible feature structures that could label the mother node.

By similar reasoning, there are exactly 21 possible feature structures that could label the first (head) daughter in a local subtree satisfying rule (59). As for the second daughter, which is constrained to be an NP, there are only 6 possibilities – those determined by varying AGR values. Thus, there are 2646 (21 × 21 × 6) local subtrees satisfying rule (59), given the grammar developed in this chapter.

Note that one of these is the local subtree shown in (62), where the mother and the head daughter have divergent HEAD values:

(62) A Tree Not Licensed by the Grammar

$$\begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \begin{bmatrix} \textit{agr-cat} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \begin{bmatrix} \textit{agr-cat} & \\ \text{PER} & \text{2nd} \\ \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{str} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \quad \begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \begin{bmatrix} \textit{agr-cat} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix}$$

It is subtrees like this that are ruled out by the HFP, because the HFP requires that the HEAD value of the mother be identical to that of the head daughter. Hence, by incorporating the HFP into our theory, we vastly reduce the number of well-formed local subtrees licensed by any headed rule. The number of local subtrees satisfying both (59) and the HFP is just 126 (21 × 6). And in fact only 42 ((6 + 1) × 6) of these will ever be used in trees for complete sentences licensed by our grammar: in such trees, a word structure must be compatible with the head daughter, but only word structures for verbs or prepositions are ever specified as [COMPS str].

We complete the picture in much the same way as we did for CFGs. A phrase structure tree Φ is licensed by a grammar G if and only if:

- Φ is terminated (i.e. the nodes at the bottom of the tree are all labeled by lexical forms),
- the mother of Φ is labeled by S,[24]
- each local subtree within Φ is licensed by a grammar rule of G or a lexical entry of G, and
- each local subtree within Φ obeys all relevant principles of G.

A grammar is successful to the extent that it can be shown that the tree structures it licenses – its models – have properties that correspond to our observations about how the language really is. Recall from our discussion in Section 2.9 of Chapter 2 that what we are taking to be the reality of language involves aspects of both the mental representations of individual speakers and the social interactions among speakers. Thus, we're idealizing a fair bit when we talk about the sentences of the language being 'out there' in the world. In particular, we're abstracting away from variation across utterances and systematic variation across speakers. But we will have plenty to talk about before this idealization gets in our way, and we will have many observations and intuitions to draw from in evaluating the claims our models make about the external reality of language.

3.4.2 An Example

Consider the sentence *They swim*. Let's suppose that the lexical entries for *they* and *swim* are as shown in (63). Note that the lexical entry for the plural form *swim* is underspecified for person.

(63) a.

$$\left\langle \text{they}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \begin{bmatrix} \text{PER} & \text{3rd} \\ \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \right\rangle$$

[24]Remember that S is now an abbreviation defined in (33) above.

b.

$$\left\langle \text{swim} \; , \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \begin{bmatrix} \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \right\rangle$$

Given these two lexical entries, the following are both well-formed local subtrees, according to our theory:

(64) a.

$$\begin{array}{c} \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \begin{bmatrix} \textit{agr-cat} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \\ | \\ \text{they} \end{array}$$

b.

$$\begin{array}{c} \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \begin{bmatrix} \textit{agr-cat} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \\ | \\ \text{swim} \end{array}$$

Observe that these word structures contain only fully resolved feature structures. Furthermore, the structure in (64b) contains a specification for the feature PER that will make the relevant tree structure compatible with the structure over *they* when we combine them to build a sentence.

These lexical structures can now be embedded within larger structures sanctioned by the rules in (54f,a) and the HFP, as illustrated in (65a,b):

(65) a.

$$\begin{array}{c}
\begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \begin{bmatrix} \textit{agr-cat} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \\
\Big| \\
\begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \begin{bmatrix} \textit{agr-cat} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \\
\Bigg| \\
\text{they}
\end{array}$$

b.

$$
\begin{bmatrix}
\textit{phrase} & \\
\text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \begin{bmatrix} \textit{agr-cat} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \\
\text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix}
\end{bmatrix}
$$

$$
\Big|
$$

$$
\begin{bmatrix}
\textit{word} & \\
\text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \begin{bmatrix} \textit{agr-cat} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \\
\text{VAL} & \begin{bmatrix} \textit{val-cat} & \\ \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix}
\end{bmatrix}
$$

$$
\Big|
$$

swim

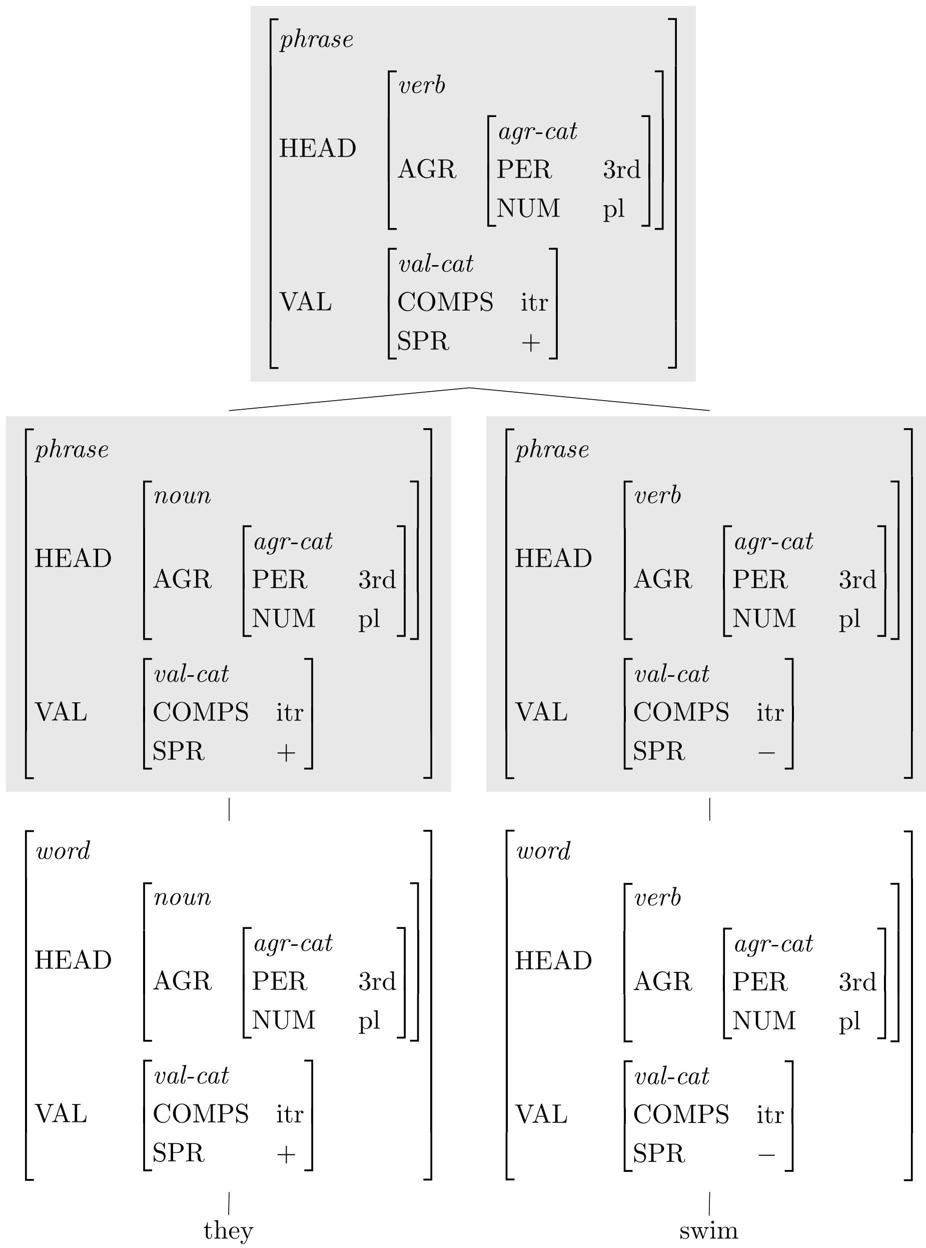
The shading in these and subsequent trees indicates the portion of the tree that is licensed by the rule in question (together with the HFP).

And finally, we can use rule (54d), repeated here as (66) to build a sentential phrase structure that combines the two previous structures. This is shown in (67):

(66)

$$
\begin{bmatrix}
\textit{phrase} & \\
\text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix}
\end{bmatrix}
\rightarrow
\overset{\text{NP}}{\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AGR} & \boxed{2} \end{bmatrix} \end{bmatrix}}
\ \mathbf{H}
\begin{bmatrix}
\textit{phrase} & \\
\text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \boxed{2} \end{bmatrix} \\
\text{VAL} & \begin{bmatrix} \text{SPR} & - \end{bmatrix}
\end{bmatrix}
$$

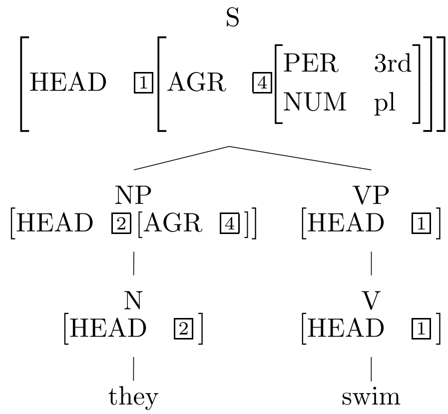
(67)



The nodes of the local subtree licensed by the rule in (66) (and the HFP) are again indicated by shading.

We will display phrase structure trees throughout this book, usually to illustrate the effect of particular constraints that are under discussion. Though the feature structures in the trees licensed by our grammar are always total functions, we will often display tree diagrams that contain defined abbreviations (e.g. NP or S) or which omit irrelevant feature specifications (or both). Similarly, we may want to illustrate particular identities within phrase structure trees that have been enforced by linguistic constraints. To this end, we will sometimes include tags (e.g. $\boxed{3}$) in our tree diagrams to indicate identities induced by linguistic constraints. To illustrate the effect of the HFP, for example, we might replace the tree diagram in (67) with one like (68):

(68)



A diagram like (68) always abbreviates a phrase structure tree whose nodes are labeled by fully determinate, resolved feature structures.

3.5 Summary

The introduction of features has given us a formal mechanism for talking about ways in which sets of words (and phrases) behave both alike and differently. By allowing embedded feature structures, underspecifying categories, and formulating general constraints stating identities that must hold in well-formed trees, we have been able to generalize our phrase structure rules and reduce their number. This in turn has led us to carefully distinguish between our grammar rules and the fully determinate ('resolved') structures that satisfy them, and further between the models licensed by our grammar and the abbreviated representations of those models such as (68) that we will often use to focus our discussions throughout the remainder of this book.

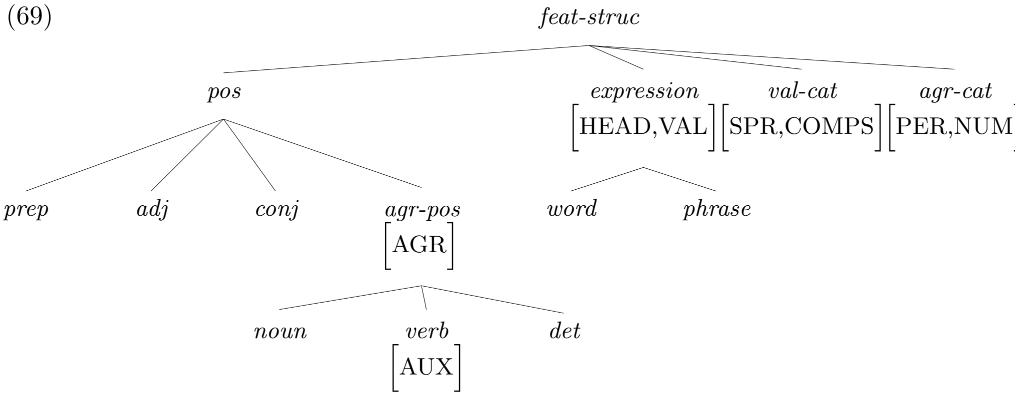
The theory we are developing is still closely related to standard CFG, yet it is somewhat more abstract. We no longer think of our phrase structure rules as specifying all the information that labels the nodes of trees. Rather, the rules, the lexicon, and some general principles – of which the HFP is the first example – all place certain constraints on trees, and any imaginable tree is well-formed so long as it conforms to these constraints. In this way, our grammar continues to be constraint-based, with the rules, lexical entries, and general principles all working together to define the well-formed structures of the language.

But the changes introduced in this chapter are not yet sufficient. They still leave us with three rules introducing complements that have too much in common and should be collapsed, and two rules introducing specifiers that similarly need to be collapsed. Moreover, as we will see in the next chapter, we have simplified the facts of agreement too much. The grammar we develop there will allow the more complex facts to be systematized, while at the same time eliminating further redundancy from the phrase structure rules of our grammar.

3.6 The Chapter 3 Grammar

3.6.1 The Type Hierarchy

(69)



3.6.2 Feature Declarations and Type Constraints

TYPE	FEATURES/CONSTRAINTS	IST
feat-struc		
pos		*feat-struc*
agr-pos	$\begin{bmatrix}\text{AGR} & \textit{agr-cat}\end{bmatrix}$	*pos*
noun		*agr-pos*
det		*agr-pos*
verb	$\begin{bmatrix}\text{AUX} & \left\{+, -\right\}\end{bmatrix}$	*agr-pos*
prep		*pos*
adj		*pos*
conj		*pos*
expression	$\begin{bmatrix}\text{HEAD} & \textit{pos} \\ \text{VAL} & \textit{val-cat}\end{bmatrix}$	*feat-struc*
word		*expression*
phrase		*expression*
val-cat	$\begin{bmatrix}\text{COMPS} & \left\{\text{itr, str, dtr}\right\} \\ \text{SPR} & \left\{+, -\right\}\end{bmatrix}$	*feat-struc*
agr-cat	$\begin{bmatrix}\text{PER} & \left\{\text{1st, 2nd, 3rd}\right\} \\ \text{NUM} & \left\{\text{sg, pl}\right\}\end{bmatrix}$	*feat-struc*

3.6.3 Abbreviations

(70)

$$\text{S} = \begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \quad \text{NP} = \begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix}$$

$$\text{VP} = \begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \quad \text{NOM} = \begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix}$$

$$\text{V} = \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{verb} \end{bmatrix} \quad \text{N} = \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{noun} \end{bmatrix}$$

$$\text{D} = \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{det} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix}$$

3.6.4 The Grammar Rules

(71) Head-Complement Rule 1:

$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \rightarrow \mathbf{H}\begin{bmatrix} \textit{word} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix}$$

(72) Head-Complement Rule 2:

$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \rightarrow \mathbf{H}\begin{bmatrix} \textit{word} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{str} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \text{ NP}$$

(73) Head-Complement Rule 3:

$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \rightarrow \mathbf{H}\begin{bmatrix} \textit{word} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{dtr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \text{ NP NP}$$

(74) Head-Specifier Rule 1:

$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \rightarrow \overset{\text{NP}}{\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \end{bmatrix}} \ \mathbf{H}\begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \boxed{1} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & - \end{bmatrix} \end{bmatrix}$$

(75) Head-Specifier Rule 2:[25]

$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix}\text{COMPS} & \text{itr}\\ \text{SPR} & +\end{bmatrix}\end{bmatrix} \rightarrow \text{D} \quad \mathbf{H}\begin{bmatrix}\textit{phrase} & \\ \text{HEAD} & \textit{noun}\\ \text{VAL} & \begin{bmatrix}\text{SPR} & -\end{bmatrix}\end{bmatrix}$$

(76) Non-Branching NP Rule:

$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix}\text{COMPS} & \text{itr}\\ \text{SPR} & +\end{bmatrix}\end{bmatrix} \rightarrow \mathbf{H}\begin{bmatrix}\textit{word} & \\ \text{HEAD} & \textit{noun}\\ \text{VAL} & \begin{bmatrix}\text{SPR} & +\end{bmatrix}\end{bmatrix}$$

(77) Head-Modifier Rule:

$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix}\text{COMPS} & \text{itr}\\ \text{SPR} & -\end{bmatrix}\end{bmatrix} \rightarrow \mathbf{H}\begin{bmatrix}\textit{phrase} & \\ \text{VAL} & \begin{bmatrix}\text{SPR} & -\end{bmatrix}\end{bmatrix} \text{PP}$$

(78) Coordination Rule:

$$\boxed{1} \rightarrow \boxed{1}^{+} \begin{bmatrix}\textit{word} & \\ \text{HEAD} & \textit{conj}\end{bmatrix} \boxed{1}$$

3.6.5 The Head Feature Principle (HFP)

In any headed phrase, the HEAD value of the mother and the HEAD value of the head daughter must be identical.

3.6.6 Sample Lexical Entries

(79)

$$\left\langle \text{walks}\ , \begin{bmatrix}\textit{word} & \\ \text{HEAD} & \begin{bmatrix}\textit{verb} & \\ \text{AGR} & \begin{bmatrix}\text{NUM} & \text{sg}\\ \text{PER} & \text{3rd}\end{bmatrix}\end{bmatrix}\\ \text{VAL} & \begin{bmatrix}\text{COMPS} & \text{itr}\\ \text{SPR} & -\end{bmatrix}\end{bmatrix}\right\rangle$$

$$\left\langle \text{walk}\ , \begin{bmatrix}\textit{word} & \\ \text{HEAD} & \begin{bmatrix}\textit{verb} & \\ \text{AGR} & \begin{bmatrix}\text{NUM} & \text{pl}\end{bmatrix}\end{bmatrix}\\ \text{VAL} & \begin{bmatrix}\text{COMPS} & \text{itr}\\ \text{SPR} & -\end{bmatrix}\end{bmatrix}\right\rangle$$

[25] See Problem 3 for more on this rule.

$$\left\langle \text{denies}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sg} \\ \text{PER} & \text{3rd} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{str} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \right\rangle$$

$$\left\langle \text{deny}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \begin{bmatrix} \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{str} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \right\rangle$$

$$\left\langle \text{defendant}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sg} \\ \text{PER} & \text{3rd} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \right\rangle$$

$$\left\langle \text{the}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{det} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \right\rangle$$

$$\left\langle \text{Chris}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \begin{bmatrix} \text{PER} & \text{3rd} \\ \text{NUM} & \text{sg} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \right\rangle$$

$$\left\langle \text{you}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \begin{bmatrix} \text{PER} & \text{2nd} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \right\rangle$$

$$\left\langle \text{and}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{conj} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \right\rangle$$

3.7 Further Reading

One of the earliest (but often ignored) demonstrations of the descriptive power of feature structures is Harman 1963. Chomsky (1965) provides one of the earliest explicit discussions of syntactic features in generative grammar. The modern tradition of using complex feature structures (that is, features with feature structures as their values) begins with Kay 1979, Bear 1981, Bresnan 1982b, and Gazdar 1981 (see also Kaplan 1975 and Gazdar et al. 1985). For an elementary discussion of the formal properties of unification and its use in grammatical description, see Shieber 1986. For differing and more detailed technical presentations of the logic of typed feature structures, see King 1989, Carpenter 1992, Richter 1999, 2000, and Penn 2000.

3.8 Problems

Problem 1: Applying the Chapter 3 Grammar

A. Formulate a lexical entry for the word *defendants*.

B. Draw a tree for the sentence *The defendants walk*. Show the values for all of the features on every node and use tags to indicate the effects of any identities that the grammar requires.

C. Explain how your lexical entry for *defendants* interacts with the Chapter 3 grammar to rule out **The defendants walks*. Your explanation should make reference to grammar rules, lexical entries and the HFP.

Problem 2: 1st Singular and 2nd Singular Forms of Verbs

The sample lexical entry for *walk* given in (79) is specified as [AGR [NUM pl]]. This accounts for (i)–(iii), but not (iv) and (v):

(i) They walk.
(ii) We walk.
(iii) You (pl) walk. (cf. You yourselves walk.)
(iv) You (sg) walk. (cf. You yourself walk.)
(v) I walk.

Formulate lexical entries for *walk* in (iv) and (v). Be sure that those lexical entries don't license (vi):

(vi)*Dana walk.

Problem 3: Determiner-Noun Agreement

The Chapter 3 grammar declares AGR to be a feature appropriate for the types *noun*, *verb*, and *det*, but so far we haven't discussed agreement involving determiners. Unlike the determiner *the*, most other English determiners do show agreement with the nouns they combine with:

(i) a bird/*a birds
(ii) this bird/*this birds
(iii) that bird/*that birds
(iv) these birds/*these bird
(v) those birds/*those bird
(vi) many birds/*many bird

A. Formulate lexical entries for *this* and *these*.
B. Modify Head-Specifier Rule 2 so that it enforces agreement between the noun and the determiner just like Head-Specifier Rule 1 enforces agreement between the NP and the VP.
C. Draw a tree for the NP *these birds*. Show the value for all features of every node and use tags to indicate the effects of any identities that the grammar (including your modified HSR2 and the Head Feature Principle) requires.

Problem 4: Coordination and Modification

The Chapter 3 Grammar includes a coordination rule that is very similar to the coordination rule from the context-free grammar in (23) in Chapter 2 (see page 32).[26] The only difference is notational: Now that we have a more general kind of notation – tags – for representing identity, we can replace the 'X's in the Chapter 2 version of the rule with tags.

The Chapter 3 Grammar also includes a Head-Modifier Rule. This rule corresponds to the two rules that introduced PPs in the Chapter 2 CFG:

(i) NOM $\rightarrow$ NOM PP
(ii) VP $\rightarrow$ VP PP

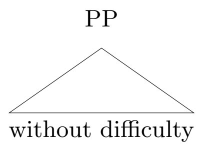
The first thing to notice about these rules is that they allow PPs to modify coordinate structures.[27] That is, the head daughter in the Head-Modifier Rule can be the entire italicized phrases in sentences like (iii) and (iv).

(iii) Alex *walks and reads books* without difficulty.
(iv) Terry likes the *poetry and music* on this program.

Of course, (iii) and (iv) are ambiguous: The PP can also be modifying just the rightmost conjunct within the coordinate structures.

[26] We will in fact revise this coordination rule in subsequent chapters.
[27] This was also true of the rules in the Chapter 2 grammar.

A. Draw the two trees for (iii) using the Chapter 3 grammar, and indicate which interpretation goes with which tree. [*Notes: You may use abbreviations for the feature structures at the nodes. Since we haven't given any sample lexical entries for prepositions, abbreviate the structure under the PP node with a triangle like this:*



The node above and may be abbreviated as CONJ.]

The Chapter 3 grammar, in its present form, doesn't allow PPs to modify Ss or NPs (which are both [SPR +]). Is this prediction correct? Consider the examples in (v) and (vi):

(v) Alex walks without difficulty.

(vi) Terry likes the music on the program.

In these examples, it is hard to tell which constituents the PPs *without difficulty* and *on the program* modify. Whether they attach low (modifying VP and NOM respectively, as currently permitted by the Chapter 3 grammar) or high (modifying S and NP, respectively, not currently permitted by the Chapter 3 grammar), we get the same string of words, and it's difficult to tell what the semantic differences between the two possible attachment sites would be. This question cannot be resolved just by considering simple examples like (v) and (vi).

B. Use coordination to resolve this question. That is, provide an argument USING EXAMPLES WITH COORDINATION to show that the prediction of the Chapter 3 grammar is incorrect: PPs must be able to modify S and NP as well as VP and NOM. [*Hint: Your argument should make reference to the different meanings associated with the different tree structures, depending on where the PP attaches.*]

Problem 5: Identifying the Head of a Phrase

The head of a phrase is the element inside the phrase whose properties determine the distribution of that phrase, i.e. the environments in which it can occur. We say that nouns head noun phrases, since (ii)-(v) can all show up in the same environments as (i): e.g. as the specifier of a verb, as a complement of a transitive verb and as the complement of prepositions like *of* or *on*.

(i) giraffes

(ii) tall giraffes

(iii) giraffes with long necks

(iv) all giraffes

(v) all tall giraffes with long necks

On the other hand (vi)–(ix) do not have the same distribution as the phrases in (i)–(v).

(vi) tall

(vii) with long necks

(viii) all

(ix) all tall

Thus it appears to be the noun in (i)–(v) that defines the distributional properties of the whole phrase, and it is the noun that we call the head.

In this problem we apply this criterion for identifying heads to a domain that is off the beaten path of grammatical analysis: English number names.[28] The goal of this problem is to identify the head in expressions like *two hundred* and *three hundred*. That is, which is the head of *two hundred*: *two* or *hundred*? In order to answer this, we are going to compare the distribution of *two hundred* with that of two minimally different phrases: *three hundred* and *two thousand*.

Now, many environments that allow *two hundred* also allow *three hundred* and *two thousand*:

(x) There were two hundred/three hundred/two thousand.

(xi) Two hundred/three hundred/two thousand penguins waddled by.

Some environments do distinguish between them, however. One such environment is the environment to the right of the word *thousand*:

(xii) four thousand two hundred

(xiii) four thousand three hundred

(xiv) *four thousand two thousand

A. Based on the data in (xii)–(xiv), which phrase has the same distribution as *two hundred*: *three hundred* or *two thousand*?

B. Does your answer to part (A) support treating *two* or *hundred* as the head of *two hundred*? Explain your answer in a sentence or two.

Similarly, we can compare the distribution of *two hundred five* to the two minimally different phrases *two hundred six* and *two thousand five*. Once again, the environment to the right of *thousand* will do:

(xv) four thousand two hundred five

(xvi) four thousand two hundred six

(xvii) *four thousand two thousand five

C. Based on the data in (xv)–(xvii), which phrase has the same distribution as *two hundred five*: *two hundred six* or *two thousand five*?

D. Does your answer to part (C) support treating *five* as the head of *two hundred five*? Briefly explain why.

---

[28] This problem is based on the analysis of English number names in Smith 1999.

4

Complex Feature Values

4.1 Introduction

By reanalyzing grammatical categories feature structures, we were able to codify the relatedness of syntactic categories and to express the property of headedness via a general principle: the Head Feature Principle. The grammar of the preceding chapter not only provides a more compact way to represent syntactic information, it also systematically encodes the fact that phrases of different types exhibit parallel structures. In particular, the rules we gave in the previous chapter suggest that lexical head daughters in English uniformly occur at the left edge of their phrases.[1] Of course, VPs and PPs are consistently head-initial. In addition, assuming our analysis of NPs includes the intermediate-level category NOM, nouns are initial in the phrases they head, as well. The Chapter 3 grammar thus expresses a correct generalization about English phrases.

One motivation for revising our current analysis, however, is that our rules are still not maximally general. We have three distinct rules introducing lexical heads, one for each of the three COMPS values. This would not necessarily be a problem, except that, as noted in Chapter 2, these three valences are far from the only possible environments lexical heads may require. Consider the examples in (1):

(1) a. Pat relies on Kim.
b.*Pat relies.
c. The child put the toy on the table.
d.*The child put the toy.
e. The teacher became angry with the students.
f.*The teacher became.
g. The jury believed the witness lied.

Examples (1a,b) show that some verbs require a following PP; (1c,d) show that some verbs must be followed by both an NP and a PP; (1e,f) show a verb that can be followed by a kind of phrase we have not yet discussed, called an adjective phrase (AP); and (1g) shows a verb that can be followed by an S. We say only that *became* CAN be followed by an AP and that *believed* CAN be followed by an S, because they can also appear in sentences like *Pat became an astronaut* and *Pat believed the story*, in which they are

[1]This is not true in some other languages, e.g. in Japanese, the lexical head daughters are phrase-final, resulting in SOV (Subject-Object-Verb) ordering, as well as noun-final NPs.

followed by NPs. In fact, it is extremely common for verbs to be able to appear in multiple environments. Similarly, (2) shows that *ate*, like many other English verbs, can be used either transitively or intransitively:

(2) The guests ate (the cheese).

Facts like these show that the number of values of COMPS must be far greater than three. Hence, the Chapter 3 grammar would have to be augmented by many more grammar rules in order to accommodate the full range of verbal subcategories. In addition, given the way COMPS values are keyed to rules, a worrisome redundancy would arise: the lexical distinctions would all be encoded twice – once in the phrase structure rules and once in the (many) new values of COMPS that would be required.

Exercise 1: More Subcategories of Verb
There are other subcategories of verb, taking different combinations of complements than those illustrated so far. Think of examples of as many as you can. In particular, look for verbs followed by each of the following sequences: NP-S, NP-AP, PP-S, and PP-PP.

Intuitively, we would like to have one rule that simply says that a phrase (a VP, in the cases above) may consist of a lexical head (a V, in these cases) followed by whatever other phrases the lexical head requires. We could then relegate to the lexicon (and only to the lexicon) the task of specifying for each word what elements must appear together with that word. In this chapter, we develop a way to do just this. It involves enriching our conception of valence features (SPR and COMPS) in a way somewhat analogous to what we did with grammatical categories in the previous chapter. The new conception of the valence features not only allows for more general rules, but also leads to a reduction of unnecessary structure in our trees and to improvements in our analysis of agreement phenomena.

4.2 Complements

4.2.1 Syntactic and Semantic Aspects of Valence

Before we begin the discussion of this analysis, let us consider briefly the status of the kinds of co-occurrence restrictions we have been talking about. It has sometimes been argued that the number and type of complements a verb takes is fully determined by its meaning. For example, the verb *disappear* is used to describe events involving a single entity (expressed by its subject); *deny*'s semantics involves events with two participants, one typically human and the other a proposition; and an event described by *hand* must include three participants: the person who does the handing, the thing handed, and the recipient of the transaction. Correspondingly, *disappear* takes no complements, only a subject; *deny* takes a subject and a complement, which may be either an NP (as in *The defendant denied the charges*) or an S (as in *The defendant denied he was guilty*); and *hand* takes a subject and two NP complements (or one NP and one PP complement).

It is undeniable that the semantics of a verb is intimately related to its valence. There is, however, a certain amount of syntactic arbitrariness to it, as well. For example, the words *eat, dine*, and *devour* all denote activities necessarily involving both a consumer of food and the food itself. Hence, if a word's valence were fully determined by its meanings, one might expect that all three would be simple transitives, requiring a subject and an NP complement (that is, a direct object). But this expectation would be wrong – *dine* is intransitive, *devour* is obligatorily transitive, and (as noted above), *eat* can be used intransitively or transitively:

(3) a. The guests devoured the meal.
b.*The guests devoured.
c.*The guests dined the meal.
d. The guests dined.
e. The guests ate the meal.
f. The guests ate.

Thus, though we recognize that there is an important link between meaning and valence, we will continue to specify valence syntactically. We will say more about the connection between meaning and valence – and more generally about the syntax-semantics interface – in later chapters.

4.2.2 The COMPS Feature

In the Chapter 3 grammar, the lexical entry for a verb like *deny* would specify that it is [COMPS str]. This ensures that it can only appear in word structures whose mother node is specified as [COMPS str], and such word structures can be used to build larger structures only by using the rule of our grammar that introduces an immediately following NP. Hence, *deny* has to be followed by an NP.[2] As noted above, the co-occurrence effects of complement selection are dealt with by positing both a new COMPS value and a new grammar rule for each co-occurrence pattern.

How can we eliminate the redundancy of such a system? An alternative approach to complement selection is to use features directly in licensing complements – that is, to have a feature whose value specifies what the complements must be. We will now make this intuitive idea explicit. First, recall that in the last chapter we allowed some features (e.g. HEAD, AGR) to take values that are feature structures themselves. If we treat COMPS as such a feature, we can allow its value to state directly what the word's complement must be. The value of COMPS for *deny* can simply be an NP, as shown in (4):

(4) $$\begin{bmatrix} \text{COMPS} & \begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \textit{noun} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix}$$

and in abbreviated form in (5):

(5) $$\begin{bmatrix} \text{COMPS} & \text{NP} \end{bmatrix}$$

Similarly, we can indicate that a verb takes another type of complement: *rely, become*, and *believe*, for example, can take COMPS values of PP, AP, and S, respectively. Optional

[2] Soon, we will consider the other possible environment for *deny*, namely the one where it is followed by a clause.

complements, such as the object of *eat* can be indicated using parentheses; that is, the lexical entry for *eat* can specify [COMPS (NP)]. Likewise, we can indicate alternative choices for complements using the vertical bar notation introduced in the discussion of regular expressions in Chapter 2. So the entry for *deny* or *believe* includes the specification: [COMPS NP | S].

Of course there is a problem with this proposal: it does not cover verbs like *hand* and *put* that require more than one complement. But it's not hard to invent a straightforward way of modifying this analysis to let it encompass multiple complements. Instead of treating the value of COMPS as a single feature structure, we will let it be a LIST of feature structures.[3] Intuitively, the list specifies a sequence of categories corresponding to the complements that the word combines with. So, for example, the COMPS values for *deny, become*, and *eat* will be lists of length one. For *hand*, the COMPS value will be a list of length two, namely ⟨ NP , NP ⟩. For verbs taking no complements, like *disappear*, the value of COMPS will be ⟨ ⟩ (a list of length zero). This will enable the rules we write to ensure that a tree containing a verb will be well-formed only if the sisters of the V-node can be identified with the categories specified on the list of the verb. For example, *rely* will only be allowed in trees where the VP dominates a V and a PP.

Now we can collapse all the different rules for expanding a phrase into a lexical head (**H**) and other material. We can just say:

(6) Head-Complement Rule

$$\begin{bmatrix} \textit{phrase} \\ \text{VAL} \quad \begin{bmatrix} \text{COMPS} \quad \langle\ \rangle \end{bmatrix} \end{bmatrix} \rightarrow \mathbf{H}\begin{bmatrix} \textit{word} \\ \text{VAL} \quad \begin{bmatrix} \text{COMPS} \quad \langle \boxed{1}, \ldots, \boxed{n} \rangle \end{bmatrix} \end{bmatrix} \quad \boxed{1} \ldots \boxed{n}$$

The tags in this rule enforce identity between the non-head daughters and the elements of the COMPS list of the head. The $\boxed{1} \ldots \boxed{n}$ notation allows this rule to account for phrases with a variable number of non-head daughters. n stands for any integer greater than or equal to 1. Thus, if a word is specified lexically as [COMPS ⟨ AP ⟩], it must co-occur with exactly one AP complement; if it is [COMPS ⟨ NP , NP ⟩], it must co-occur with exactly two NP complements, and so forth. Finally, the mother of any structure licensed by (6), which we will call a HEAD-COMPLEMENT PHRASE, must be specified as [COMPS ⟨ ⟩], because that mother must satisfy the description on the left-hand side of the rule.[4]

In short, the COMPS list of a lexical entry specifies a word's co-occurrence requirements; and the COMPS list of a phrasal node is empty. So, in particular, a V must have sisters that match all the feature structures in its COMPS value, and the VP that it heads has the empty list as its COMPS value and hence cannot combine with complements. The Head-Complement Rule, as stated, requires all complements to be realized as sisters of the lexical head.[5]

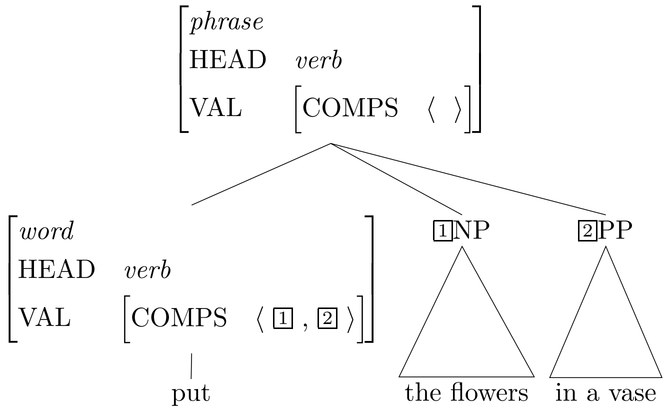
[3] Recall that we used this same technique to deal with multiple founders of organizations in our feature-structure model of universities presented at the beginning of Chapter 3.

[4] Note that by underspecifying the complements introduced by this rule – not even requiring them to be phrases, for example – we are implicitly leaving open the possibility that some complements will be nonphrasal. This will become important below and in the analysis of negation presented in Chapter 13.

[5] This flat structure appears well motivated for English, but our general theory would allow us to write a Head-Complement Rule for some other language that allows some of the complements to be introduced higher in the tree structure. For example, structures like the one in (i) would be allowed by a version of the Head-Complement Rule that required neither that the head daughter be of type *word*

If you think in terms of building the tree bottom-up, starting with the verb as head, then the verb has certain demands that have to be satisfied before a complete, or 'saturated', constituent is formed. On this conception, the complements can be thought of as being 'cancelled off' of the head daughter's COMPS list in the process of building a headed phrase. We illustrate this with the VP *put the flowers in a vase*: the verb *put* requires both a direct object NP and a PP complement, so its COMPS value is ⟨ NP , PP ⟩. The requisite NP and PP will both be sisters of the V, as in (7), as all three combine to form a VP, i.e. a verbal phrase whose complement requirements have been fulfilled:

(7)

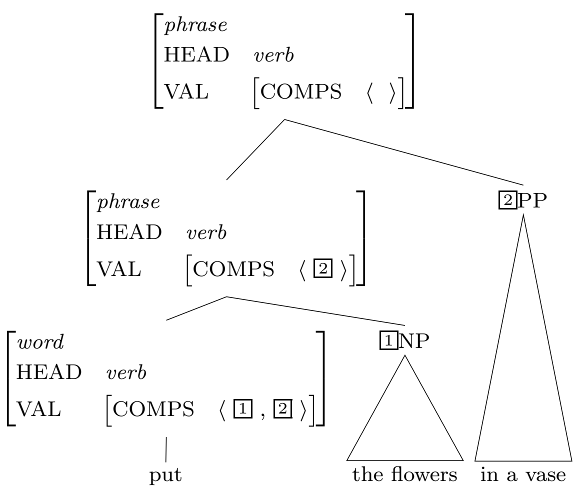


As is evident from this example, we assume that the elements in the value of COMPS occur in the same order as they appear in the sentence. We will continue to make this assumption, though ultimately a more sophisticated treatment of linear ordering of phrases in sentences may be necessary.

---

nor that the mother have an empty COMPS list:

(i) Tree Licensed by a Hypothetical Alternative Head-Complement Rule:



Such grammatical variations might be regarded as 'parameters' that are set differently in particular languages. That is, it may be that all languages manifest the Head-Complement Rule, but there are minor differences in the way languages incorporate the rule into their grammar. The order of the head and the complements is another possible parameter of variation.

4.2.3 Complements vs. Modifiers

A common source of confusion is the fact that some kinds of constituents, notably PPs, can function either as complements or as modifiers. This often raises the question of how to analyze a particular PP: should it be treated as a complement, licensed by a PP on the COMPS list of a nearby word, or should it be analyzed as a modifier, introduced by a different grammar rule? Some cases are clear. For example, we know that a PP is a complement when the choice of preposition is idiosyncratically restricted by another word, such as the verb *rely*, which requires a PP headed by *on* or *upon*:

(8) a. We relied on/upon Leslie.
  b.*We relied over/with/on top of/above Leslie.

In fact, PPs that are obligatorily selected by a head (e.g. the directional PP required by *put*) can safely be treated as complements, as we will assume that modifiers are always optional.

Conversely, there are certain kinds of PP that seem to be able to co-occur with almost any kind of verb, such as temporal or locative PPs, and these are almost always analyzed as modifiers. Another property of this kind of PP is that they can iterate: that is, where you can get one, you can get many:

(9) a. We celebrated in the streets.
  b. We celebrated in the streets in the rain on Tuesday in the morning.

The underlying intuition here is that complements refer to the essential participants in the situation that the sentence describes, whereas modifiers serve to further refine the description of that situation. This is not a precisely defined distinction, and there are problems with trying to make it into a formal criterion. Consequently, there are difficult borderline cases that syntacticians disagree about. Nevertheless, there is considerable agreement that the distinction between complements and modifiers is a real one that should be reflected in a formal theory of grammar.

4.2.4 Complements of Non-verbal Heads

Returning to our analysis of complements, notice that although we have motivated our treatment of complements entirely in terms of verbs and verb phrases, we have formulated our analysis to be more general. In particular, our grammar of head-complement structures allows adjectives, nouns, and prepositions to take complements of various types. The following examples suggest that, like verbs, these kinds of words exhibit a range of valence possibilities:

(10) Adjectives
  a. The children are happy.
  b. The children are happy with the ice cream.
  c. The children are happy that they have ice cream.
  d.*The children are happy of ice cream.
  e.*The children are fond.
  f.*The children are fond with the ice cream.
  g.*The children are fond that they have ice cream.
  h. The children are fond of ice cream.

(11) Nouns

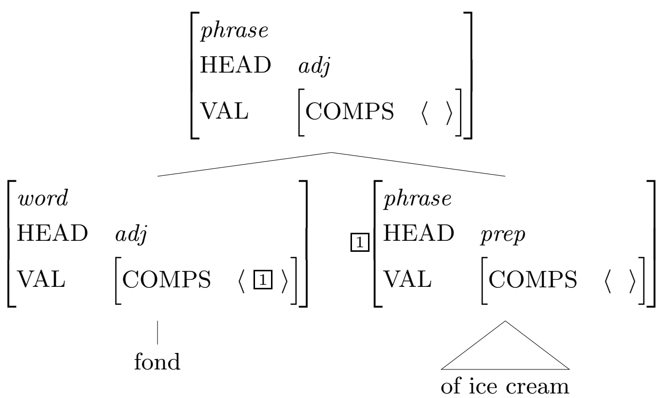
a. A magazine appeared on the newsstands.
b. A magazine about crime appeared on the newsstands.
c. *Newsweek* appeared on the newsstands.
d.**Newsweek* about crime appeared on the newsstands.
e. The report surprised many people.
f. The report that crime was declining surprised many people.
g. The book surprised many people.
h.*The book that crime was declining surprised many people.

(12) Prepositions

a. The storm arrived after the picnic.
b. The storm arrived after we ate lunch.
c. The storm arrived during the picnic.
d.*The storm arrived during we ate lunch.
e.*The storm arrived while the picnic.
f. The storm arrived while we ate lunch.

The Head-Complement Rule can license APs, PPs, and NPs in addition to VPs. As with the VPs, it will license only those complements that the head A, P or N is seeking. This is illustrated for adjectives in (13): the complement PP, tagged $\boxed{1}$, is precisely what the head adjective's COMPS list requires:

(13)



Exercise 2: COMPS Values of Non-Verbal Heads
Based on the examples above, write out the COMPS values for the lexical entries of *happy, magazine, Newsweek, report, book, after, during*, and *while.*

4.3 Specifiers

Co-occurrence restrictions are not limited to complements. As we have noted in earlier chapters, certain verb forms appear with only certain types of subjects. In particular, in the present tense, English subjects and verbs must agree in number. Likewise, as we saw in Problem 3 of Chapter 3, certain determiners co-occur only with nouns of a particular number:

(14) a. This dog barked.
b.*This dogs barked.
c.*These dog barked.
d. These dogs barked.

Moreover, some determiners co-occur only with 'mass' nouns (e.g. *furniture, footwear, information*), and others only with 'count' nouns (e.g. *chair, shoe, fact*), as illustrated in (15):

(15) a. Much furniture was broken.
b.*A furniture was broken.
c.*Much chair was broken.
d. A chair was broken.

We can handle such co-occurrence restrictions in much the same way that we dealt with the requirements that heads impose on their complements. To do so, we will reinterpret the feature SPR in the same way we reinterpreted the feature COMPS. Later in this chapter (see Sections 4.6.1 and 4.6.2), we'll see how we can use these features to handle facts like those in (14)–(15).

Recall that in Chapter 3, we used the term specifier to refer to both subjects and determiners. We will now propose to collapse our two earlier head-specifier rules into one grammar rule that will be used to build both Ss and NPs. In the Chapter 3 grammar, the feature SPR takes atomic values (+ or −) and records whether or not the phrase contains a specifier.[6] On analogy with the feature COMPS, the feature SPR will now take a list as its value. The lexical entry for a verb (such as *sleep*, *deny*, or *hand*) will include the following specification:

(16) $\begin{bmatrix}\text{SPR} & \langle\ \text{NP}\ \rangle\end{bmatrix}$

Likewise, the lexical entry for a noun like *book*, *meal*, or *gift* will include the following specification:

(17) $\begin{bmatrix}\text{SPR} & \langle\ [\text{HEAD}\ \ \textit{det}]\ \rangle\end{bmatrix}$

The decision to treat the value of SPR as a list may strike some readers as odd, since sentences only have a single subject and NPs never have more than one determiner. But notice that it allows the feature SPR to continue to serve roughly the function it served in the Chapter 3 grammar, namely recording whether the specifier requirement of a phrase is satisfied. Indeed, making SPR list-valued provides a uniform way of formulating the

[6]More precisely, whether or not a given phrase has satisfied any needs it might have to combine with a specifier. Recall that proper nouns are also [SPR +] in the Chapter 3 grammar.

idea that a particular valence requirement is unfulfilled (the valence feature – COMPS or SPR – has a nonempty value) or else is fulfilled (the value of the valence feature is the empty list).

We can now redefine the category NOM in terms of the following feature structure descriptions:[7]

(18)
$$\text{NOM} = \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle \text{ X } \rangle \end{bmatrix} \end{bmatrix}$$

And once again there is a family resemblance between our interpretation of NOM and the description abbreviated by VP, which is now as shown in (19):

(19)
$$\text{VP} = \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle \text{ X } \rangle \end{bmatrix} \end{bmatrix}$$

Both (18) and (19) have empty COMPS lists and a single element in their SPR lists. Both are intermediate between categories with nonempty COMPS lists and saturated expressions – that is, expressions whose COMPS and SPR lists are both empty.

Similarly, we can introduce a verbal category that is analogous in all relevant respects to the saturated category NP. This verbal category is the feature structure analog of the familiar category S:

(20)
$$\text{NP} = \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \qquad \text{S} = \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix}$$

Note crucially that our abbreviations for NOM, VP, NP and S no longer mention the type *phrase*. Since these are the constructs we will use to formulate rules and lexical entries in this chapter (and the rest of the book), we are in effect shifting to a perspective where phrasality has a much smaller role to play in syntax. The binary distinction between words and phrases is largely replaced by a more nuanced notion of 'degree of saturation' of an expression – that is the degree to which the elements specified in the head's valence features are present in the expression. As we will see in a moment, there is a payoff from this perspective in terms of simpler phrase structure trees.

Because NP and S now have a parallel formulation in terms of feature structures and parallel constituent structures, we may collapse our old rules for expanding these categories (given in (21)) into a single rule, shown in (22):

[7] The specification [SPR ⟨ X ⟩] represents a SPR list with exactly one element on it. The 'X' is used to represent a completely underspecified feature structure. In the case of a NOM, this element will always be [HEAD *det*], but it would be redundant to state this in the definition of the abbreviation.

(21) Head-Specifier Rules from the Chapter Three Grammar:

a.
$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \rightarrow \overset{\text{NP}}{\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \end{bmatrix}} \quad \mathbf{H}\begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \boxed{1} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & - \end{bmatrix} \end{bmatrix}$$

b.
$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \text{itr} \\ \text{SPR} & + \end{bmatrix} \end{bmatrix} \rightarrow \text{D} \quad \mathbf{H}\begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & - \end{bmatrix} \end{bmatrix}$$

(22) Head-Specifier Rule (Version I)

$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \rightarrow \boxed{2} \quad \mathbf{H}\begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle \boxed{2} \rangle \end{bmatrix} \end{bmatrix}$$

The tag $\boxed{2}$ in this rule identifies the SPR requirement of the head daughter with the non-head daughter. If the head daughter is 'seeking' an NP specifier (i.e. is specified as [SPR ⟨ NP ⟩]), then the non-head daughter will be an NP. If the head daughter is 'seeking' a determiner specifier, then the non-head daughter will be [HEAD det]. Phrases licensed by (22) will be known as HEAD-SPECIFIER PHRASEs.

We said earlier that the lexical entries for nouns and verbs indicate what kind of specifier they require. However, the head-daughter of a head-specifier phrase need not be a word. For example, in the sentence *Kim likes books*, the head daughter of the head-specifier phrase will be the phrase *likes books*. Recall that the head-complement rules in the Chapter 3 grammar all required that the mother and the head daughter be specified as [SPR −]. In our current grammar, however, we need to ensure that the particular kind of specifier selected by the head daughter in a head-complement phrase is also selected by the head-complement phrase itself (so that a VP combines only with an NP and a NOM combines only with a determiner). We must somehow guarantee that the SPR value of a head-complement phrase is the same as the SPR value of its head daughter.[8] We might thus add a stipulation to this effect, as shown in (23):[9]

(23) Head-Complement Rule (Temporary Revision)

$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \boxed{A} \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \rightarrow \mathbf{H}\begin{bmatrix} \textit{word} & \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \boxed{A} \\ \text{COMPS} & \langle \boxed{1}, \ldots, \boxed{n} \rangle \end{bmatrix} \end{bmatrix} \boxed{1} \ldots \boxed{n}$$

[8] At first glance, one might be tempted to accomplish this by making SPR a head feature, but in that case the statement of the HFP would have to be complicated, to allow rule (22) to introduce a discrepancy between the HEAD value of a mother and its head daughter.

[9] This version of the Head-Complement Rule should be considered a temporary revision, as we will soon find a more general way to incorporate this constraint into the grammar.

(Note that here we are using the tag $\boxed{A}$ to designate neither an atomic value nor a feature structure, but rather a list of feature structures.[10])

4.4 Applying the Rules

Now that we have working versions of both the Head-Specifier and Head-Complement Rules, let's use them to construct a tree for a simple example. These rules build the tree in (26) for the sentence in (24) from the lexical entries in (25):[11]

(24) Alex likes the opera.

(25) a.

$$\left\langle \text{likes}\ , \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \text{NP}\ \rangle \\ \text{COMPS} & \langle\ \text{NP}\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

b.

$$\left\langle \text{Alex}\ , \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

c.

$$\left\langle \text{the}\ , \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{det} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

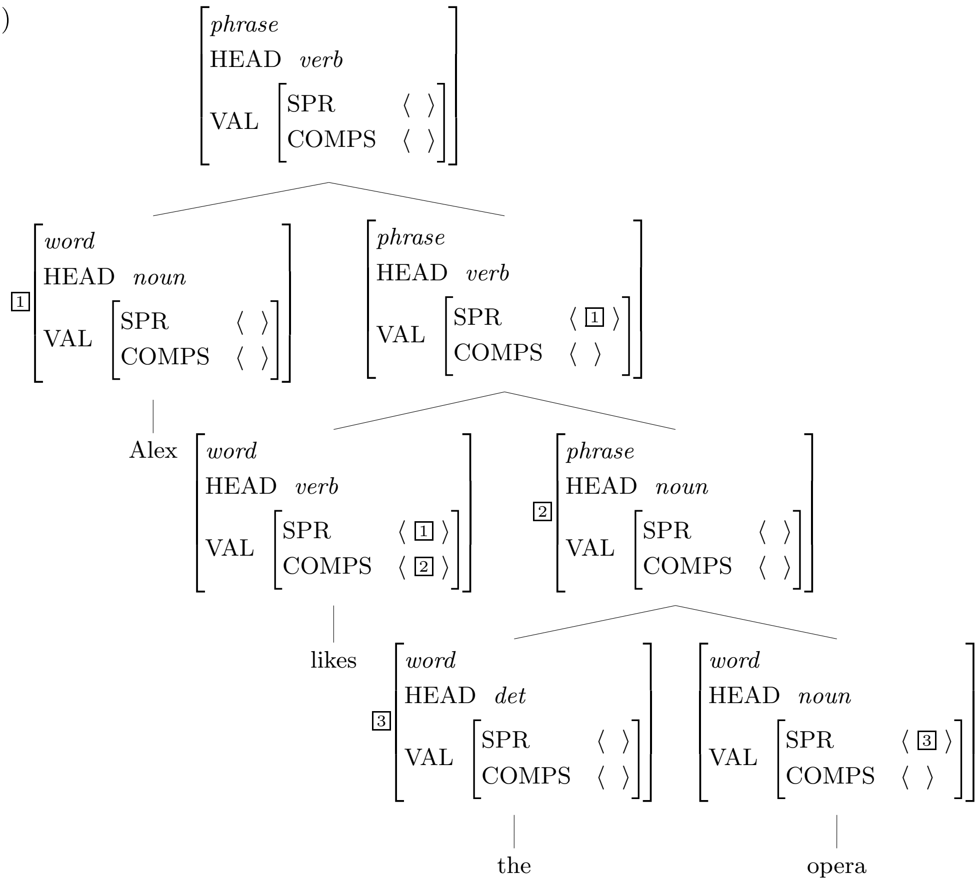
d.

$$\left\langle \text{opera}\ , \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \text{D}\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

[10] We will henceforth adopt the convention of using numbers to tag feature structures or atomic values and letters to tag lists of feature structures.

[11] For the purposes of this example, we are ignoring the problem of subject-verb agreement. It will be taken up below in Section 4.6.1.

(26)



There are several things to notice about this tree:

First, compared to the trees generated by the Chapter 3 grammar, it has a simpler constituent structure. In particular, it has no non-branching nodes (except those immediately dominating the actual words). The Head-Specifier Rule requires that its head daughter be [COMPS ⟨ ⟩], but there are two ways that this could come about. The head daughter could be a word that is [COMPS ⟨ ⟩] to start with, like *opera*; or it could be a phrase licensed by the Head-Complement Rule, like *likes the opera*. This phrase is [COMPS ⟨ ⟩] according to the definition of the Head-Complement Rule. In brief, the head daughter of the Head-Specifier Rule can be either a *word* or a *phrase*, as long as it is [COMPS ⟨ ⟩].

Similarly, the verb *likes* requires an NP complement and an NP specifier. Of course, the symbol NP (and similarly D) is just an abbreviation for a feature structure description, namely that shown in (20). Once again, we see that the type (*word* or *phrase*) of the expression isn't specified, only the HEAD, SPR and COMPS values. Thus any nominal expression that is saturated (i.e. has no unfulfilled valence features) can serve as the specifier or complement of *likes*, regardless of whether it's saturated because it started out that way (like *Alex*) or because it 'has already found' the specifier it selected lexically (as in *the opera*).

This is an advantage of the Chapter 4 grammar over the Chapter 3 grammar: the non-branching nodes in the trees licensed by the Chapter 3 grammar constitute unmotivated extra structure. As noted above, this structural simplification is a direct consequence of our decision to continue specifying things in terms of NP, NOM, S and VP, while changing the interpretation of these symbols. However, we will continue to use the symbols N and V as abbreviations for the following feature structure descriptions:

(27)
$$\mathrm{N} \;=\; \begin{bmatrix} \textit{word} & \\ \mathrm{HEAD} & \textit{noun} \end{bmatrix} \qquad \mathrm{V} \;=\; \begin{bmatrix} \textit{word} & \\ \mathrm{HEAD} & \textit{verb} \end{bmatrix}$$

This means that in some cases, two abbreviations may apply to the same node. For instance, the node above *Alex* in (26) may be abbreviated as either NP or N. Similarly, the node above *opera* may be abbreviated as either NOM or N. This ambiguity is not problematic, as the abbreviations have no theoretical status in our grammar: they are merely there for expository convenience.

Another important thing to notice is that the rules are written so that head-complement phrases are embedded within head-specifier phrases, and not vice versa. The key constraint here is the specification on the Head-Complement Rule that the head daughter must be of type *word*. Since the mother of the Head-Specifier Rule is of type *phrase*, a head-specifier phrase can never serve as the head daughter of a head-complement phrase.

A final thing to notice about the tree is that in any given phrase, one item is the head and it selects for its sisters. That is, *Alex* is the specifier of *likes the opera* (and also of *likes*), and *likes* is not the specifier or complement of anything.

Exercise 3: Which Rules Where?
Which subtrees of (26) are licensed by the Head-Complement Rule and which are licensed by the Head-Specifier Rule?

4.5 The Valence Principle

Recall that in order to get the SPR selection information from a lexical head like *likes* or *story* to the (phrasal) VP or NOM that it heads, we had to add a stipulation to the Head-Complement Rule. More stipulations are needed if we consider additional rules. In particular, recall the rule for introducing PP modifiers, discussed in the previous chapter. Because no complements or specifiers are introduced by this rule, we do not want any cancellation from either of the head daughter's valence features to take place. Hence, we would need to complicate the rule so as to transmit values for both valence features up from the head daughter to the mother, as shown in (28):

(28) Head-Modifier Rule (Version I)

$$\begin{bmatrix} \textit{phrase} & \\ \mathrm{VAL} & \begin{bmatrix} \mathrm{SPR} & \boxed{\mathrm{A}} \\ \mathrm{COMPS} & \boxed{\mathrm{B}} \end{bmatrix} \end{bmatrix} \;\rightarrow\; \mathbf{H}\begin{bmatrix} \mathrm{VAL} & \begin{bmatrix} \mathrm{SPR} & \boxed{\mathrm{A}} \\ \mathrm{COMPS} & \boxed{\mathrm{B}}\langle\ \rangle \end{bmatrix} \end{bmatrix} \quad \mathrm{PP}$$

Without some such requirement, the combination of a modifier and a VP wouldn't be constrained to be a VP rather than, say, an S. Similarly, a modifier could combine with an S to build a VP. It is time to contemplate a more general theory of how the valence features behave in headed phrases.

The intuitive idea behind the features SPR and COMPS is quite straightforward: certain lexical entries specify what they can co-occur with by listing the particular kinds of dependents they select. We formulated general rules stating that all the head's COMPS members are 'discharged' in a head-complement phrase and that the item in the SPR value is discharged in a head-specifier phrase. But to make these rules work, we had to add constraints preserving valence specifications in all other instances: the mother in the Head-Specifier Rule preserves the head's COMPS value (the empty list); the mother in the Head-Complement Rule preserves the head's SPR value, and the mother in the Head-Modifier Rule must preserve both the COMPS value and the SPR value of the head. The generalization that can be factored out of our rules is expressed as the following principle which, like the HFP, constrains the set of trees that are licensed by our grammar rules:

(29) The Valence Principle
Unless the rule says otherwise, the mother's values for the VAL features (SPR and COMPS) are identical to those of the head daughter.

By 'unless the rule says otherwise', we mean simply that the Valence Principle is enforced unless a particular grammar rule specifies both the mother's and the head daughter's value for some valence feature.

The effect of the Valence Principle is that: (1) the appropriate elements mentioned in particular rules are canceled from the relevant valence specifications of the head daughter in head-complement or head-specifier phrases, and (2) all other valence specifications are simply passed up from head daughter to mother. Once we factor these constraints out of our headed rules and put them into a single principle, it again becomes possible to simplify our grammar rules. This is illustrated in (30):

(30) a. Head-Specifier Rule (Near-Final Version)

$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \rangle\end{bmatrix}\end{bmatrix} \rightarrow \boxed{1} \quad \mathbf{H}\begin{bmatrix}\text{VAL} & \begin{bmatrix}\text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix}\end{bmatrix}$$

b. Head-Complement Rule (Final Version)

$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix}\text{COMPS} & \langle\ \rangle\end{bmatrix}\end{bmatrix} \rightarrow \mathbf{H}\begin{bmatrix}\textit{word} & \\ \text{VAL} & \begin{bmatrix}\text{COMPS} & \langle \boxed{1}, \ldots, \boxed{n} \rangle\end{bmatrix}\end{bmatrix} \ \boxed{1} \ldots \boxed{n}$$

c. Head-Modifier Rule (Version II)

$$\begin{bmatrix} \textit{phrase} \end{bmatrix} \rightarrow \mathbf{H}\begin{bmatrix}\text{VAL} & \begin{bmatrix}\text{COMPS} & \langle\ \rangle\end{bmatrix}\end{bmatrix} \ \text{PP}$$

While the simplicity of the rules as formulated in (30) is striking, our work is not yet done. We will make further modifications to the Head-Modifier Rule in the next chapter and again in Chapter 14. The Head-Specifier Rule will receive some minor revision in Chapter 14 as well. While the Head-Complement Rule is now in its final form, we will be introducing further principles that the rules interact with in later chapters.

4.6 Agreement Revisited

Let us now return to the problem of agreement. Our earlier analysis assigned the feature AGR to both nouns and verbs, and one of our grammar rules stipulated that the AGR values of VPs and their subjects had to match. In addition, as we saw in Problem 3 of Chapter 3, determiner-noun agreement is quite similar and could be treated by a similar stipulation on a different grammar rule. These two rules are now collapsed into our Head-Specifier Rule and so we could consider maintaining essentially the same rule-based analysis of agreement in this chapter's grammar.

However, there is a problem with this approach. There are other constructions, illustrated in (31), that we will also want to analyze as head-specifier phrases:

(31) a. They want/preferred [them *arrested*].
b. We want/preferred [them *on* our team].
c. With [them *on* our team], we'll be sure to win.
d. With [my parents as *supportive* as they are], I'll be in fine shape.

Clauses like the bracketed expressions in (31a,b) are referred to as SMALL CLAUSES; the constructions illustrated in (31c,d) are often called ABSOLUTE constructions. The problem here is that the italicized prepositions and adjectives that head these head-specifier phrases are not compatible with the feature AGR, which is defined only for the parts of speech *det*, *noun*, and *verb*. Nor would there be any independent reason to let English prepositions and adjectives bear AGR specifications, as they have no inflectional forms and participate in no agreement relations. Hence, if we are to unify the account of these head-specifier phrases, we cannot place any general constraint on them which makes reference to AGR.

There is another approach to agreement that avoids this difficulty. Suppose we posit a lexical constraint on verbs and common nouns that requires their AGR value and the AGR value of the specifier they select to be identical. This constraint could be formulated as in (32):

(32) Specifier-Head Agreement Constraint (SHAC)
Verbs and common nouns must be specified as:

$$\begin{bmatrix} \text{HEAD} & \begin{bmatrix}\text{AGR} & \boxed{1}\end{bmatrix} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \left\langle \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{AGR} & \boxed{1}\end{bmatrix}\end{bmatrix} \right\rangle\end{bmatrix}\end{bmatrix}$$

This formulation does not specify precisely what the SHAC's formal status in the grammar is. This will be rectified in Chapter 8. We introduce it here so that we can move subject-verb agreement and determiner-noun agreement out of the grammar rules and into the lexicon, without having to stipulate the agreement separately in the lexical entry of every verb and common noun. The formalization in Chapter 8 has the desired effect of avoiding the unwanted redundancy by locating specifier-head agreement in one place in the grammar.

4.6.1 Subject-Verb Agreement

This proposal can accommodate the facts of subject-verb agreement without difficulty. A verb like *walks* has a lexical entry like the one shown in (33):

(33) walks:

$$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \boxed{1}\begin{bmatrix} \text{PER} & \text{3rd} \\ \text{NUM} & \text{sg} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \begin{array}{c} \text{NP} \\ \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \end{bmatrix} \end{array} \right\rangle \end{bmatrix} \end{bmatrix}$$

Given entries like (33), the Head-Specifier Rule in (30a) above will induce agreement, simply by identifying the head daughter's SPR value with the specifier daughter. An NP like (34) is a compatible specifier for (33), but an NP like (35) is not:

(34) Kim:

$$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \begin{bmatrix} \text{PER} & \text{3rd} \\ \text{NUM} & \text{sg} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix}$$

(35) we:

$$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \begin{bmatrix} \text{PER} & \text{1st} \\ \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix}$$

This lexicalized approach to subject-verb agreement will account for the familiar contrasts like (36):

(36) a. Kim walks.
b.*We walks.

As before, the HFP will transmit agreement constraints down to the head noun of a subject NP, accounting for the pattern illustrated in (37):

(37) a. The child walks.
b.*The children walks.

At the same time, since the Head-Specifier Rule now makes no mention of AGR, it may also be used to construct small clauses (as in (31a, b)) and absolute constructions (as in (31c, d)), whose head daughters can be APs or PPs that are incompatible with AGR.[12]

[12] The details of the grammar of small clauses and absolute constructions, however, are beyond the scope of this textbook.

Let us now examine subject-verb agreement more closely. First, recall that English agreement depends on person, as well as number. We have analyzed person in terms of varying specifications for the feature PER. [PER 1st] is our notation for first person, that is, the pronouns *I* and *we*. [PER 2nd] denotes second person, which in English is always *you*. [PER 3rd] covers all nonpronominal NPs, as well as *he*, *she*, *it*, and *they*. Most present tense English verbs have one form when their subjects are third-person singular (namely a form ending in *-s*) and another form covering all other persons and numbers. The only verb whose present tense system makes finer distinctions than this is *be*, which has a special first-person singular form, *am*, a third-person singular form, *is*, and an additional form *are* (appropriate wherever *am* and *is* are not).

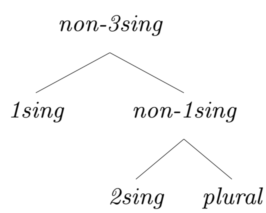
The generalization we would like to capture is this: although there are six different combinations of person and number in English, the vast majority of English verbs group these six possibilities into two sets – third person singular and other. This distinction can be incorporated into our grammar via the type hierarchy. Suppose we introduce two types called *3sing* and *non-3sing*, both immediate subtypes of the type *agr-cat*.

Instances of the type *3sing* obey the constraint shown in (38):

(38) $$\textit{3sing} : \begin{bmatrix} \text{PER} & \text{3rd} \\ \text{NUM} & \text{sg} \end{bmatrix}$$

The subtypes of *non-3sing* will be constrained to have other combinations of PER and NUM values. One possible organization of these subtypes (and the one we will adopt) is shown in (39):

(39)



The types *1sing*, *2sing*, and *plural* bear the constraints shown in (40):

(40) $$\textit{1sing} : \begin{bmatrix} \text{PER} & \text{1st} \\ \text{NUM} & \text{sg} \end{bmatrix}$$

$$\textit{2sing} : \begin{bmatrix} \text{PER} & \text{2nd} \\ \text{NUM} & \text{sg} \end{bmatrix}$$

$$\textit{plural} : \begin{bmatrix} \text{NUM} & \text{pl} \end{bmatrix}$$

The types *3sing* and *non-3sing* are motivated by the co-occurrence of verbs and nouns, however, there is actually independent evidence for the type distinction. Recall that one function of the type hierarchy is to allow us to state which features are approriate for each type of linguistic object. While PER and NUM are appropriate for both *3sing* and *non-3sing* (and will therefore be declared on the supertype *agr-cat*), the feature GEND(ER) is only appropriate to *3sing*: GEND (with values *masc*, *fem*, and *neut*) will serve to differentiate among *he, she,* and *it*, *him, her,* and *it*, and *himself, herself*, and *itself.* There is no motivation in English for assigning GEND to anything other than words that are third-person and singular.

With the addition of GEND, the full set of possible AGR values is as shown in (41):

(41) Possible AGR Values

$$\begin{bmatrix} \textit{1sing} & \\ \text{PER} & \text{1st} \\ \text{NUM} & \text{sg} \end{bmatrix} \quad \begin{bmatrix} \textit{2sing} & \\ \text{PER} & \text{2nd} \\ \text{NUM} & \text{sg} \end{bmatrix}$$

$$\begin{bmatrix} \textit{plural} & \\ \text{PER} & \text{1st} \\ \text{NUM} & \text{pl} \end{bmatrix} \quad \begin{bmatrix} \textit{plural} & \\ \text{PER} & \text{2nd} \\ \text{NUM} & \text{pl} \end{bmatrix} \quad \begin{bmatrix} \textit{plural} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{pl} \end{bmatrix}$$

$$\begin{bmatrix} \textit{3sing} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{sg} \\ \text{GEND} & \text{fem} \end{bmatrix} \quad \begin{bmatrix} \textit{3sing} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{sg} \\ \text{GEND} & \text{masc} \end{bmatrix} \quad \begin{bmatrix} \textit{3sing} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{sg} \\ \text{GEND} & \text{neut} \end{bmatrix}$$

Observe the absence of GEND on the *non-3sing* types.

This treatment of the AGR values of nouns and NPs leads to a (minor) simplification in the lexical entries for nouns and verbs. The third-person singular proper noun *Kim* and the present-tense verb form *walks* will now have lexical entries like the following:

(42) a.

$$\left\langle \text{Kim}\ , \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \textit{3sing} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

b.

$$\left\langle \text{walks}\ , \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \textit{3sing} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \text{NP}\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Lexical entries like (42b) are further subject to the SHAC, as described above.

On the other hand, we can use a single lexical entry for all the other present tense uses of a given verb. It is often assumed that it is necessary to posit separate lexical entries for present tense verb forms that take plural subjects and those that take singular, non-third-person subjects, as sketched in (43a,b):

(43) a.

$$\left\langle \text{walk}\ , \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \begin{bmatrix} \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \text{NP}\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

b.

$$\left\langle \text{walk}\ , \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \begin{bmatrix} \text{PER} & \text{1st} \mid \text{2nd} \\ \text{NUM} & \text{sg} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \text{NP}\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

But such an analysis would fail to explain the fact that the former type of verb would always be identical in form to the latter: again, a suspicious loss of generalization in the lexicon.

Once we bifurcate the types of AGR values, as described above, this problem disappears. We need only a single kind of verb subsuming both (43a) and (43b), one that includes the following lexical information:

(44) $$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AGR} & \textit{non-3sing} \end{bmatrix} \end{bmatrix}$$

Because of the SHAC, verbs so specified project VPs that take subjects whose head nouns must bear *non-3sing* AGR values, and these, as described above, must either be first-person singular, second-person singular, or plural.

The disjunctions needed for describing classes of verbs are thus given by the type hierarchy, not by writing arbitrarily disjunctive lexical entries. In fact, one of the goals of a grammar that uses types is to predict in this manner which disjunctions play a significant role in the grammatical analysis of a given language (or of language in general).

Exercise 4: The AGR Values of *am* and *are*
What would be the AGR values in the lexical entries for *am* and *are*?

4.6.2 Determiner-Noun Agreement

We have just seen how our new analysis of specifiers, taken together with the Specifier-Head Agreement Constraint and the Head Feature Principle, provides an account of the fact that a third-person singular verb form (e.g. *walks*) takes a subject NP headed by a third-person singular noun. But, as we have already seen, the specifiers of the phrases projected from these nouns also agree in number. Recall from Problem 3 of Chapter 3 that English has determiners like *this* and *a*, which only appear with singular nouns, plural determiners like *these* and *few*, which only appear with plural nouns, and other determiners like *the*, which go either way:

(45) a. This dog barked.
b.*This dogs barked.
c. A dog barked.
d.*A dogs barked.

(46) a.*These dog barked.
b. These dogs barked.

c.*Few dog barked.

d. Few dogs barked.

(47) a. The dog barked.

b. The dogs barked.

There is systematic number agreement between heads and specifiers within the NP.

We will assume that common nouns are lexically specified as shown in (48):

(48) $\begin{bmatrix}\text{SPR} & \langle\ [\text{HEAD}\ \ \textit{det}]\ \rangle\end{bmatrix}$

Hence, by the SHAC, whatever constraints we place on the AGR value of common nouns will also apply to the determiners they co-occur with. Determiner-noun agreement, like subject-verb agreement, is a lexical fact about nouns. This account makes crucial use of our hypothesis (discussed in detail in Chapter 3) that determiners and nouns both bear AGR specifications, as illustrated in (49):[13]

(49) person, boat, a, this: $\begin{bmatrix}\text{AGR} & \textit{3sing}\end{bmatrix}$

people, boats, few, these: $\begin{bmatrix}\text{AGR} & \begin{bmatrix}\text{PER} & \text{3rd}\\ \text{NUM} & \text{pl}\end{bmatrix}\end{bmatrix}$

the: $\begin{bmatrix}\text{AGR} & \begin{bmatrix}\text{PER} & \text{3rd}\end{bmatrix}\end{bmatrix}$

These lexical specifications, taken together with the SHAC and the HFP, provide a complete account of the agreement data in (45)–(47) above.

4.6.3 Count and Mass Revisited (COUNT)

In Section 4.4 above, we also observed that some determiners are restricted to occur only with 'mass' nouns (e.g. *furniture*), and others only with 'count' nouns (e.g. *chair*):

(50) a. Much furniture was broken.

b.*A furniture was broken.

c.*Much chair was broken.

d. A chair was broken.

The co-occurrence restriction illustrated in (50) – that is, the count noun/mass noun distinction – might, of course, be solely a semantic matter. In order to give it a semantic analysis, we would need to find a solid semantic criterion that would relate the meaning of any given noun to its classification according to the distributional facts. Indeed, many mass nouns (such as *air*, *water*, *sand*, and *information*) do seem to have a lot in common semantically. However, the distributional class of mass nouns also contains words like *furniture* and *succotash*.[14] These words tend to resist semantic characterizations that

[13] Since we identify the whole AGR values, we are actually analyzing determiners and nouns as agreeing in both person and number. This analysis makes different predictions from an analysis that just identified the NUM values. It might for example allow a proper treatment of NPs like *you philosophers* or *us linguists*, assuming that pronouns lead a second life as determiners.

[14] a dish of cooked lima beans and corn

work for the other members of the class. For example, no matter how you divide up a quantity of water, the smaller portions are still water. The same is more or less true for *air*, *sand*, and *information*, but not true for *furniture* and *succotash*. Any semantic analysis that doesn't extend to all members of the distributional class 'mass nouns' will need to be supplemented with a purely syntactic analysis of the (semantically) oddball cases.

In the absence of a complete semantic analysis, we will analyze the data in (50) syntactically by introducing a feature COUNT. Certain determiners (e.g. *a* and *few*) will be lexically specified as [COUNT +] and others (e.g. *much*) will be lexically treated as [COUNT −], on the basis of which nouns they co-occur with. Still other determiners, such as *the*, will be lexically unmarked for this feature, because they co-occur with both kinds of nouns. The SPR value of a count noun like *chair* would then be ⟨ D[COUNT +] ⟩, forcing such nouns to co-occur with a count determiner. And the SPR value of a mass noun like *furniture* would be ⟨ D[COUNT −] ⟩.[15]

Notice that, in contrast to AGR, COUNT is a feature only of determiners. What we might informally refer to as a 'count noun' (like *dog*) is actually one whose SPR value contains a [COUNT +] determiner. This information is not passed up to the NP node that dominates the noun. Since a verb's SPR value specifies what kind of NP it takes as its subject, only information that appears on the NP node can be selected. Consequently, our analysis predicts that no English verb requires a count (or mass) subject (or object). To the best of our knowledge, this prediction is correct.

4.6.4 Summary

In this section, we have considered two kinds of agreement: subject-verb agreement and determiner-noun agreement. In both cases, we have analyzed the agreement in terms of the SPR requirement of the head (verb or noun). Once we take into account the effects of the SHAC, our analysis includes the following lexical entries:

(51) a.

$$\left\langle \text{dog}\ , \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \boxed{1} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{det} & \\ \text{AGR} & \boxed{1}\textit{3sing} \\ \text{COUNT} & + \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

[15]We postpone discussion of the optionality of determiners until Chapter 8.

b.

$$\left\langle \text{walks}\ ,\ \left[\begin{array}{ll} \textit{word} & \\ \text{HEAD} & \left[\begin{array}{ll} \textit{verb} & \\ \text{AGR} & \boxed{1} \end{array}\right] \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \left\langle \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll} \textit{noun} & \\ \text{AGR} & \boxed{1}\textit{3sing} \end{array}\right] \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{array}\right] \end{array}\right] \right\rangle \\ \text{COMPS} & \langle\ \rangle \end{array}\right] \end{array}\right] \right\rangle$$

c.

$$\left\langle \text{the}\ ,\ \left[\begin{array}{ll} \textit{word} & \\ \text{HEAD} & \textit{det} \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{array}\right] \end{array}\right] \right\rangle$$

We have designed the architecture of our feature structures and the way they interact with our general principles to have specific empirical consequences. The parallel distribution of the feature AGR in the noun and verb feature structures above reflects the fact that both verbs and nouns agree with their specifiers. In the sentence *The dog walks*, the AGR value on the noun *dog* will pass up to the NP that it heads, and that NP then has to satisfy the specifier requirement of the verb *walks*. Nouns play a dual role in agreement: as the head of the specifier in subject-verb agreement, and as the head with which the specifier must agree in determiner-noun agreement.[16]

The picture we now have of head-specifier structures is summarized in (52).

[16]Notice that verbs also pass up their AGR specification to the VP and S phrases they project. Hence, our analysis predicts that this information about the subject NP of a sentence is locally accessible at those higher levels of structure and could be selected for or agreed with higher in the tree. This view might well be supported by the existence of verb agreement in 'tag questions':

(i) He is leaving, isn't he?

(ii) *He is leaving, isn't she?

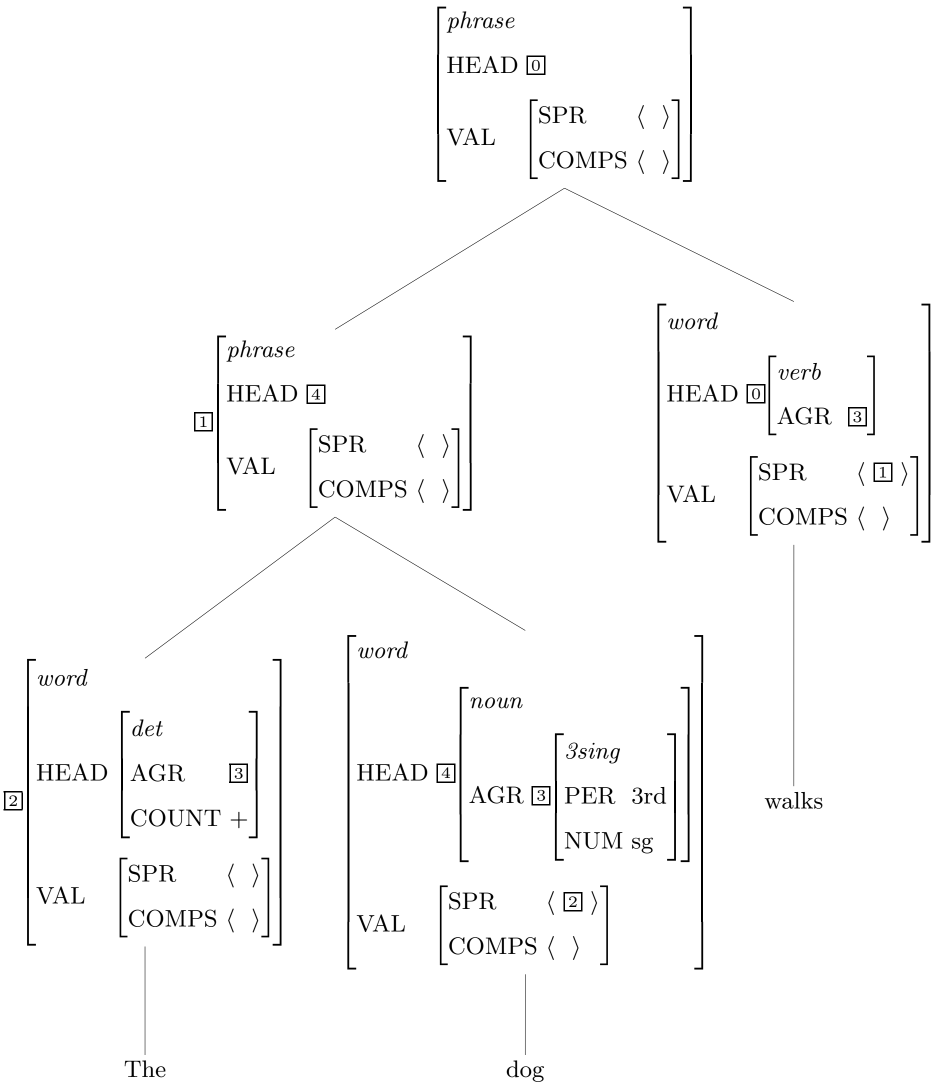
(iii) *He is leaving, aren't they?

(iv) They are leaving, aren't they?

(v) *They are leaving, isn't she?

Once again, such issues are beyond the scope of this textbook. For more on tag questions, see Bender and Flickinger 1999.

(52)



There are several things to notice about this tree:

- The HEAD value of the noun *dog* ($\boxed{4}$) and that of the phrase above it are identical in virtue of the HFP.
- Similarly, the HFP guarantees that the HEAD value of the verb *walks* ($\boxed{0}$) and that of the phrase above it are identical.
- The SHAC guarantees that the AGR value of the verb ($\boxed{3}$) is identical to that of the NP it selects as a specifier ($\boxed{1}$).
- The SHAC also guarantees that the AGR value of the noun ($\boxed{3}$) is identical to that of the determiner it selects as a specifier ($\boxed{2}$).
- Since the AGR of the noun specification is within the noun's HEAD value $\boxed{4}$, it follows from the interaction of the SHAC and the HFP that the AGR values of the NP, N, and D in (52) are all identical.

- This means in turn that whenever a verb selects a certain kind of subject NP (an [AGR *3sing*] NP in the case of the verb *walks* in (52)), that selection will restrict what kind of noun and (indirectly, through the noun's own selectional restrictions) what kind of determiner can occur within the subject NP, as desired.

4.7 Coordination and Agreement

The coordination rule from the Chapter 3 grammar, repeated here as (53), identifies the entire *expression* of the mother with the *expression*s of the conjunct daughters:

(53) Coordination Rule (Chapter 3 version):

$$\boxed{1} \rightarrow \boxed{1}^{+} \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{conj} \end{bmatrix} \boxed{1}$$

Together with our analysis of agreement, this rule makes some incorrect predictions. For example, it wrongly predicts that the examples in (54) should be ungrammatical, since the conjunct daughters have differing AGR values:

(54) a. I walk and Dana runs.
 b. Two cats and one dog live there.

Exercise 5: AGR in Coordination
Using abbreviations like NP, S and VP, draw the tree the grammar should assign to (54a). What are the AGR values of the S nodes dominating *I walk* and *Dana runs*? Where do they come from?

These data show that requiring complete identity of feature values between the conjuncts is too strong. In fact, the problem of determining exactly which information must be shared by the conjuncts and the mother in coordinate structures is a very tricky one. For now, we will revise the Coordination Rule as in (55), but we will return to this rule again in Chapters 5, 8 and 14:

(55) Coordination Rule (Chapter 4 version):

$$\begin{bmatrix} \text{VAL } \boxed{1} \end{bmatrix} \rightarrow \begin{bmatrix} \text{VAL } \boxed{1} \end{bmatrix}^{+} \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{conj} \end{bmatrix} \begin{bmatrix} \text{VAL } \boxed{1} \end{bmatrix}$$

The Coordination Rule in (55) states that any number of constituents with the same VAL value can be coordinated to form a constituent whose mother has the same VAL value. Since AGR is in HEAD (not VAL), the rule in (55) will license the sentences in (54).

However, this rule goes a bit too far in the other direction, and now overgenerates. For example, it allows NPs and Ss to coordinate with each other:

(56)*The dog slept and the cat.

On the other hand, the overgeneration is not as bad as it might seem at first glance. In particular, for non-saturated constituents (i.e. those with non-empty SPR or COMPS values), the requirement that the SPR and COMPS values be identified goes a long way towards ensuring that the conjuncts have the same part of speech as well. For example, a NOM like *cat* can't be coordinated with a VP like *slept* because they have different SPR values. In Chapter 8 we will see how to constrain conjuncts to have the same part of speech without requiring identity of the whole HEAD value.

Identifying VAL values (and therefore SPR values) also makes a very nice prediction about VP versus S coordination. While Ss with different AGR values can be coordinated as in (54a), VPs with different AGR values cannot, as shown in (57):

(57)*Kim walks and run.

Another way to phrase this is that VPs with differing SPR requirements can't be coordinated, and that is exactly how we capture this fact. Problem 9 addresses the issue of AGR values in coordinated NPs.

4.8 Case Marking

Yet another kind of selectional dependency found in many languages is the phenomenon of CASE MARKING. Case marking is a kind of variation in the form of Ns or NPs, depending on their syntactic environment. (This was addressed briefly in Problem 6 of Chapter 2.)

While many languages have case systems that involve all kinds of nouns, English has a very impoverished case system, where only pronouns show case distinctions:

(58) a. We like them.
b. They like us.
c.*We like they.
d.*Us like them.
e. Kim likes dogs.
f. Dogs like Kim.

In these examples, the forms *we* and *they* are in the NOMINATIVE case (sometimes called the SUBJECTIVE case), and the forms *us* and *them* are in the ACCUSATIVE case (sometimes called the OBJECTIVE case). Other languages have a larger selection of cases.

In Chapter 2, Problem 6 asked you to write phrase structure rules that would account for the different case markings associated with different positions in English. This kind of analysis of case marking no longer makes much sense, because we have replaced the very specific phrase structure rules of earlier chapters with more general rule schemas. With the theoretical machinery developed in this chapter, we handle case entirely in the lexicon, without changing our grammar rules. That is, the style of analysis we developed for agreement will work equally well for case marking. All we'll need is a new feature CASE that takes the atomic values 'nom' and 'acc' (and others for languages with more case distinctions). Problems 5–8 concern applying the machinery to case systems in English, Icelandic, and the Australian language Wambaya, and address issues such as what kind of feature structure CASE is a feature of.

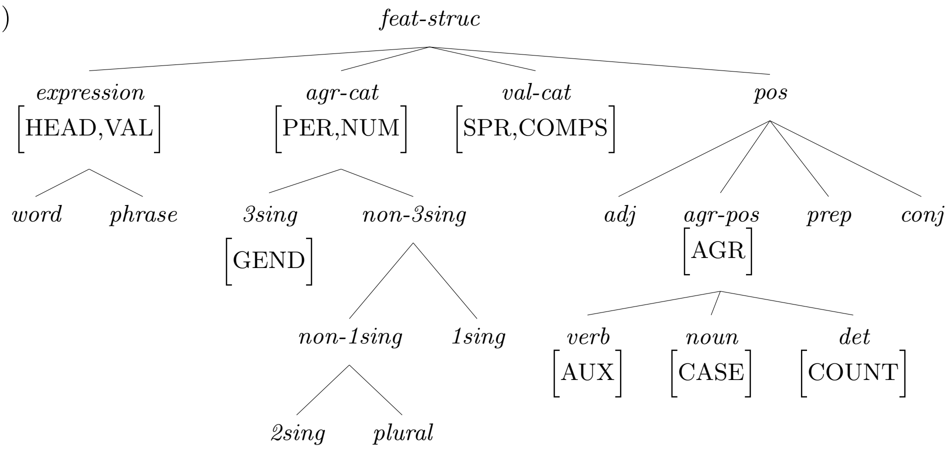
4.9 Summary

In the previous chapter, we had already seen that cross-categorial generalizations about phrase structure can be expressed in terms of schematic phrase structure rules and categories specified in terms of feature structures. In this chapter, the real power of feature structure grammars has begun to emerge. We have begun the process of providing a unified account of the generalizations about complementation and specifier selection, in terms of the list-valued features COMPS and SPR. These features, together with the Valence Principle, have enabled us to eliminate further redundancy from our grammar rules. In fact, our grammar has now been reduced to four very general rules. In this chapter, we've also seen that key generalizations about agreement can be expressed in terms of this highly compact rule system, once we rely on categories modeled as feature structures and a single Specifier-Head Agreement Constraint. Problems 5 through 8 concern extending this style of analysis to case marking phenomena.

4.10 The Chapter 4 Grammar

4.10.1 The Type Hierarchy

(59)



4.10.2 Feature Declarations and Type Constraints

TYPE	FEATURES/CONSTRAINTS	IST
feat-struc		
expression	$\begin{bmatrix}\text{HEAD} & \textit{pos}\\ \text{VAL} & \textit{val-cat}\end{bmatrix}$	*feat-struc*
word		*expression*
phrase		*expression*
val-cat	$\begin{bmatrix}\text{SPR} & \textit{list(expression)}^{17}\\ \text{COMPS} & \textit{list(expression)}\end{bmatrix}$	*feat-struc*
pos		*feat-struc*
agr-pos	$\begin{bmatrix}\text{AGR} & \textit{agr-cat}\end{bmatrix}$	*pos*
verb	$\begin{bmatrix}\text{AUX} & \{+, -\}\end{bmatrix}$	*agr-pos*
noun	$\begin{bmatrix}\text{CASE} & \{\text{nom, acc}\}\end{bmatrix}$	*agr-pos*
det	$\begin{bmatrix}\text{COUNT} & \{+, -\}\end{bmatrix}$	*agr-pos*
adj,prep,conj		*pos*
agr-cat	$\begin{bmatrix}\text{PER} & \{\text{1st, 2nd, 3rd}\}\\ \text{NUM} & \{\text{sg, pl}\}\end{bmatrix}$	*feat-struc*
3sing	$\begin{bmatrix}\text{PER} & \text{3rd}\\ \text{NUM} & \text{sg}\\ \text{GEND} & \{\text{fem, masc, neut}\}\end{bmatrix}$	*agr-cat*
non-3sing		*agr-cat*
1sing	$\begin{bmatrix}\text{PER} & \text{1st}\\ \text{NUM} & \text{sg}\end{bmatrix}$	*non-3sing*
non-1sing		*non-3sing*
2sing	$\begin{bmatrix}\text{PER} & \text{2nd}\\ \text{NUM} & \text{sg}\end{bmatrix}$	*non-1sing*
plural	$\begin{bmatrix}\text{NUM} & \text{pl}\end{bmatrix}$	*non-1sing*

[17]The formal status of list types like this one is explicated in the Appendix to Chapter 6.

4.10.3 Abbreviations

(60)

$$\text{S} = \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \qquad \text{NP} = \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix}$$

$$\text{VP} = \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle \text{X} \rangle \end{bmatrix} \end{bmatrix} \qquad \text{NOM} = \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle \text{X} \rangle \end{bmatrix} \end{bmatrix}$$

$$\text{V} = \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{verb} \end{bmatrix} \qquad \text{N} = \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{noun} \end{bmatrix}$$

$$\text{D} = \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{det} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix}$$

4.10.4 The Grammar Rules

(61) Head-Specifier Rule

$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \rightarrow \boxed{1} \quad \mathbf{H}\begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix}$$

(62) Head-Complement Rule

$$\begin{bmatrix} \textit{phrase} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \rightarrow \mathbf{H}\begin{bmatrix} \textit{word} & \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle \boxed{1}, \ldots, \boxed{n} \rangle \end{bmatrix} \end{bmatrix} \boxed{1} \ldots \boxed{n}$$

(63) Head-Modifier Rule

$$\begin{bmatrix} \textit{phrase} \end{bmatrix} \rightarrow \mathbf{H}\begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \text{PP}$$

(64)

Coordination Rule

$$\begin{bmatrix} \text{VAL} \ \boxed{1} \end{bmatrix} \rightarrow \begin{bmatrix} \text{VAL} \ \boxed{1} \end{bmatrix}^{+} \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{conj} \end{bmatrix} \begin{bmatrix} \text{VAL} \ \boxed{1} \end{bmatrix}$$

4.10.5 The Principles

(65) Head Feature Principle (HFP)

In any headed phrase, the HEAD value of the mother and the HEAD value of the head daughter must be identical.

(66) Valence Principle

Unless the rule says otherwise, the mother's values for the VAL features (SPR and COMPS) are identical to those of the head daughter.

(67) Specifier-Head Agreement Constraint (SHAC)[18]

Verbs and common nouns must be specified as:

$$\begin{bmatrix} \text{HEAD} & [\text{AGR} \ \boxed{1}] \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\, [\text{AGR} \ \boxed{1}] \,\rangle\end{bmatrix} \end{bmatrix}$$

4.10.6 Sample Lexical Entries

(68)

$$\left\langle \text{I}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix}\textit{noun} & \\ \text{AGR} & \textit{1sing}\end{bmatrix} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix} \end{bmatrix} \right\rangle$$

(69)

$$\left\langle \text{dog}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix}\textit{noun} & \\ \text{AGR} & \textit{3sing}\end{bmatrix} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \left\langle \begin{matrix}\text{D}\\ [\text{COUNT} \ \ +]\end{matrix} \right\rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix} \end{bmatrix} \right\rangle$$

(70)

$$\left\langle \text{furniture}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix}\textit{noun} & \\ \text{AGR} & \textit{3sing}\end{bmatrix} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \left\langle \begin{matrix}\text{D}\\ [\text{COUNT} \ \ -]\end{matrix} \right\rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix} \end{bmatrix} \right\rangle$$

(71)

$$\left\langle \text{a}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix}\textit{det} & \\ \text{AGR} & \textit{3sing} \\ \text{COUNT} & +\end{bmatrix} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix} \end{bmatrix} \right\rangle$$

[18] The SHAC is a principle for now, but once we have a more developed theory of lexical types in Chapter 8, it will be expressed as a constraint on the type *inflecting-lexeme*.

(72)

$$\left\langle \text{much}\ , \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix} \textit{det} & \\ \text{AGR} & \textit{3sing} \\ \text{COUNT} & - \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(73)

$$\left\langle \text{barks}\ , \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \textit{3sing} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \text{ NP } \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(74)

$$\left\langle \text{like}\ , \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \textit{non-3sing} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \text{ NP } \rangle \\ \text{COMPS} & \langle \text{ NP } \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

4.11 Further Reading

The idea of schematizing phrase structure rules across parts of speech was introduced into generative grammar by Chomsky (1970). For a variety of perspectives on grammatical agreement, see Barlow and Ferguson 1988. A helpful discussion of Icelandic case (see Problem 7) is provided by Andrews (1982). For discussion and an analysis of NP coordination, see Dalrymple and Kaplan 2000 and Sag 2003.

4.12 Problems

Problem 1: Valence Variations

In this problem, you will be asked to write lexical entries (including HEAD, SPR, and COMPS values). You may use NP, VP, etc. as abbreviations for the feature structures on COMPS lists.

As you do this problem, keep the following points in mind: (1) In this chapter we've changed COMPS to be a list-valued feature, and (2) heads select for their specifier and complements (if they have any); the elements on the SPR and COMPS lists do not simultaneously select for the head.

[*Hint: For the purposes of this problem, assume that adjectives and prepositions all have empty SPR lists.*]

A. Write lexical entries for the words *here* and *there* as they are used in (i).

   (i) Kim put the book here/there.

   [*Hint: Compare (i) to (7) on page 97.*]

B. Write a lexical entry for the adjective *fond*. Your lexical entry should account for the contrast between (10e,g) and (10h).

C. Assume that motion verbs like *jump*, *move*, etc. take an optional PP complement, that is, that these verbs have the following specification in their lexical entries:

   $\left[\text{COMPS} \quad \langle \text{ (PP) } \rangle\right]$

   Given that, use the following examples to write the lexical entries for the prepositions *out*, *from* and *of*:

   (i) Kim jumped out of the bushes.
   (ii) Bo jumped out from the bushes.
   (iii) Lee moved from under the bushes.
   (iv) Leslie jumped out from under the bushes.
   (v) Dana jumped from the bushes.
   (vi) Chris ran out the door.
   (vii)*Kim jumped out of from the bushes.
   (viii) Kim jumped out.
   (ix)*Kim jumped from.

D. Based on the following data, write the lexical entries for the words *grew* (in the 'become' sense, not the 'cultivate' sense), *seemed*, *happy*, and *close*.

   (i) They seemed happy (to me).
   (ii) Lee seemed an excellent choice (to me).
   (iii)*They seemed (to me).
   (iv) They grew happy.
   (v)*They grew a monster (to me).
   (vi)*They grew happy to me.
   (vii) They grew close to me.
   (viii) They seemed close to me to Sandy.

   [*Note: APs have an internal structure analogous to that of VPs. Though no adjectives select NP complements (in English), there are some adjectives that select PP complements (e.g. to me), and some that do not.*]

E. Using the lexical entries you wrote for part (D), draw a tree (showing the values of HEAD, SPR, and COMPS at each node, using tags as appropriate) for *They seemed close to me to Sandy.*

Problem 2: Spanish NPs I

In English, gender distinctions are only shown on pronouns, and the vast majority of common nouns are [GENDER neuter] (that is, if they serve as the antecedent of a pronoun, that pronoun will be *it*). The gender system in Spanish differs from English in two respects. First, gender distinctions are shown on determiners and adjectives as well as on pronouns. Second, all common nouns are assigned either masculine or feminine gender (there is no neuter). This problem concerns agreement in Spanish, including gender agreement.

Consider the following data from Spanish:

(i) a. La jirafa corrió.
The.FEM.SG giraffe ran.3SG
'The giraffe ran.'
b.*Las/El/Los jirafa corrió.

(ii) a. Las jirafas corrieron.
The.FEM.PL giraffes ran.3PL
'The giraffes ran.'
b.*La/El/Los jirafas corrieron.

(iii) a. El pingüino corrió.
The.MASC.SG penguin RAN.3SG
'The penguin ran.'
b.*La/Las/Los pingüino corrió.

(iv) a. Los pingüinos corrieron.
The.MASC.PL penguins RAN.3PL
'The penguins ran.'
b.*La/Las/El pingüinos corrieron.

A. Do the Spanish nouns shown obey the SHAC? Why or why not?
B. For English, we argued that the feature GEND(ER) is only appropriate for agreement categories (*agr-cat*s) that are *3sing* (i.e. PER 3rd, NUM sg). Is this true for Spanish as well? Why or why not?
C. Write lexical entries for *la*, *los*, and *pingüino*.

Problem 3: COUNT and NUM

Section 4.6.2 provides analyses of the co-occurrence restrictions between nouns and determiners that have to do with the count/mass distinction and with number agreement. An alternative analysis would eliminate the feature COUNT and assign three values to the feature NUM: sg, pl, and mass. That is, mass nouns like *furniture* would be given the value [NUM mass]. Use the following data to provide an argument favoring the analysis given in the text over this alternative:

(i) We don't have much $\left\{\begin{matrix}\text{rice}\\\text{oats}\end{matrix}\right\}$.

(ii) *We don't have many $\left\{\begin{matrix}\text{rice}\\\text{oats}\end{matrix}\right\}$.

(iii) The rice is in the bowl.

(iv)*The rice are in the bowl.

(v) The oats are in the bowl.

(vi)*The oats is in the bowl.

[*Note: You may speak a variety of English that accepts many oats as a well-formed NP. There are some other nouns that are like oats in the relevant respects in at least some dialects, including grits (as a kind of cereal), mashed potatoes, and (somewhat distastefully, but grammatically more clearly) feces. If you can find a noun that patterns as we claim oats does in examples (i)–(vi), work the problem using that noun. If your dialect has no such nouns, then work the problem for the dialect described here, putting aside your own judgments.*]

Problem 4: Complements and Specifiers in Pipil

Consider the following data from Pipil (Uto-Aztecan, El Salvador).[19]

(i) Miki-k ne masaat.
die.PAST the deer
'The deer died.'

(ii) Mukwep-ki ne tengerechul.
return.PAST the lizard
'The lizard returned.'

(iii) Yaah-ki kadentroh ne taakatsin.
go.PAST inside the little-man
'The little man went inside.'

(iv) Muchih-ki alegrár ne piltsintsín.
do.PAST rejoicing the little-boy
'The little boy rejoiced.' (Literally, 'The little boy did rejoicing.')

(v) Kichih-ke-t ne tiit ne pipiltsitsín.
make.PAST.PLURAL the fire the little-boys
'The little boys made the fire.'

A. Assume Pipil has a VP constituent—that is, a constituent that groups together the verb and its complements but excludes the specifier. Based on the VPs in (iii)–(v) write a Head-Complement Rule for this language.

B. Does this language have one Head-Specifier Rule or two? Explain your answer making reference to the data given above, and show the rule(s) you posit. [*Note: Your analysis need only account for the data given in (i)–(v). Don't worry about phrase types that aren't illustrated.*]

Problem 5: Assessing the Facts of English Case

As noted in Chapter 2, NPs appear in a variety of positions in English, including subject of a sentence, direct object of a verb, second object of a ditransitive verb like *give*, and object of a preposition. For each of these NP positions, determine which case the pronouns in that position must have. Give grammatical and ungrammatical examples of pronouns in the various positions to support your claims.

[*Note: Not all English pronouns show case distinctions, so be sure that the pronouns you use to answer this question are the kind that do.*]

[19]We would like to thank Bill Weigel for his help in constructing this problem. The data are from Campbell 1985, 102–103. He gives more detailed glosses for many of the words in these sentences.

Problem 6: A Lexical Analysis

Section 4.8 hinted that case marking can be handled in the same way that we handle agreement, i.e. without any changes to the grammar rules. Show how this can be done. Your answer should include a prose description of how the analysis works and lexical entries for *they*, *us*, *likes* and *with*.

[*Hint: Assume that there is a feature CASE with the values 'acc' and 'nom', and assume that English pronouns have CASE values specified in their lexical entries.*]

Problem 7: Case Marking in Icelandic

Background: Icelandic is closely related to English, but it has a much more elaborate and interesting case system. For one thing, it has four cases: nominative, accusative, genitive, and dative. Second, case is marked not just on pronouns, but also on nouns. A third difference is illustrated in the following examples:[20]

(i) Drengurinn kyssti stúlkuna.
the-boy.NOM kissed the-girl.ACC
'The boy kissed the girl.'

(ii) Drengina vantar mat.
the-boys.ACC lacks food.ACC
'The boys lack food.'

(iii) Verkjanna gætir ekki.
the-pains.GEN is-noticeable not
'The pains are not noticeable.'

(iv) Barninu batnaði veikin.
the-child.DAT recovered-from the-disease.NOM
'The child recovered from the disease.'

The case markings indicated in these examples are obligatory. Thus, for example, the following is ungrammatical because the subject should be accusative:

(v) *Drengurinn vantar mat.
the-boy.NOM lacks food.ACC

Your task: Explain how the examples in (i)–(iv) bear on the analysis of case marking in Icelandic. In particular, explain how they provide direct empirical evidence for treating case marking as a lexical phenomenon, rather than one associated with particular phrase structure positions. Be sure to sketch the lexical entry for at least one of these verbs.

[20] In the glosses, NOM stands for 'nominative', ACC for 'accusative', GEN for 'genitive', and DAT for 'dative'. Although it may not be obvious from these examples, there is in fact ample evidence (which we cannot present here) that the initial NPs in these examples are the subjects of the verbs that follow them.

The word-by-word glosses in (ii) and (iii) translate the verbs with third-person singular forms, but the translations below them use plural verbs that agree with the subjects. This is because verbs only agree with nominative subjects, taking a default third-person singular inflection with non-nominative subjects. This fact is not relevant to the central point of the problem.

Problem 8: Agreement and Case Marking in Wambaya

In Wambaya, a language of Northern Australia, nouns are divided into four genders: masculine (M), feminine (F), vegetable (V), and neuter (N). They are also inflected for case, such as ergative (E) and accusative (A). Consider the following Wambaya sentences, paying attention only to the agreement between the determiners and the nouns (you do not have to worry about accounting for, or understanding, the internal structure of these words or anything else in the sentence).[21]

(i) Ngankiyaga bungmanyani ngiya-ngajbi yaniyaga darranggu.
that.F.E woman.F.E she-saw that.N.A tree.N.A
'That woman saw that tree.'

(ii) Ngankiyaga bungmanyani ngiya-ngajbi mamiyaga jigama.
that.F.E woman.F.E she-saw that.V.A yam.V.A
'That woman saw that yam.'

(iii) Ngankiyaga bungmanyani ngiya-ngajbi iniyaga bungmaji.
that.F.E woman.F.E she-saw that.M.A man.M.A
'That woman saw that man.'

(iv) Ninkiyaga bungmanyini gina-ngajbi naniyaga bungmanya.
that.M.E man.M.E he-saw that.F.A woman.F.A
'That man saw that woman.'

(v) Ninkiyaga bungmanyini gina-ngajbi yaniyaga darranggu.
that.M.E man.M.E he-saw that.N.A tree.N.A
'That man saw that tree.'

(vi) Ninkiyaga bungmanyini gina-ngajbi mamiyaga jigama.
that.M.E man.M.E he-saw that.V.A yam.V.A
'That man saw that yam.'

Ergative is the standard name for the case of the subject of a transitive verb in languages like Wambaya, where intransitive and transitive subjects show different morphological patterns. Nothing crucial in this problem hinges on the distinction between nominative and ergative case. Note that the agreement patterns in (i)–(vi) are the only ones possible; for example, changing *mamiyaga* to *yaniyaga* in (vi) would be ungrammatical. Note also that the verbs are selecting for the case of the subject and object NPs, so, for example, *gina-ngajbi* must take an ergative subject and accusative object.

A. Verbs in Wambaya select subject and object NPs of a particular case and that case is morphologically expressed on the head nouns of the NPs. This means that we must get the information about which case the verb requires down from the NP to the N (or, alternatively, get the information about which case the N is in up from

[21] In fact, the Wambaya data presented here are simplified in various ways: only one of the numerous word-order patterns is illustrated and the auxiliary plus verb sequences (e.g. *ngiya-ngajbi*) are here presented as a single word, when in fact the auxiliary is an independent verb in 'second' position. We are grateful to Rachel Nordlinger, who constructed this problem, in addition to conducting the field work upon which it is based.

the N to the NP). Assuming that the relevant rules and principles from the Chapter 4 grammar of English apply in Wambaya, we could get this result automatically if we put the feature CASE in the right place in the feature structure (i.e. made it a feature of the right type of feature structure). Where should we put the feature CASE?

B. Given your answer to part (A), would our analysis of determiner-noun agreement in English work for Wambaya determiner-noun agreement? Explain your answer, giving lexical entries for *bungmanyani*, *ngankiyaga*, *bungmaji*, and *iniyaga*.

Problem 9: Agreement in NP Coordination

NP coordination exhibits some special properties. These properties are often taken as motivation for positing a second coordination rule just for NP coordination. However, there remains disagreement about the exact details of such a rule; in fact, this is an active area of current research. The purpose of this problem is to explore some of the special properties of NP coordination, and in particular, NP coordination with *and*.

We will focus on the agreement properties of coordinated NPs. The first thing to note is that the Coordination Rule doesn't specify any information about the value of the mother. This is clearly underconstrained. Consider first the feature NUM:

(i) $\text{Kim} \left\{ \begin{array}{c} \text{walks} \\ \text{*walk} \end{array} \right\}.$

(ii) $\text{Sandy} \left\{ \begin{array}{c} \text{walks} \\ \text{*walk} \end{array} \right\}.$

(iii) $\text{Kim and Sandy} \left\{ \begin{array}{c} \text{*walks} \\ \text{walk} \end{array} \right\}.$

(iv) $\text{One dog and two cats} \left\{ \begin{array}{c} \text{*lives} \\ \text{live} \end{array} \right\} \text{here.}$

A. What conclusion can you draw from the data in (i)–(iv) about the NUM value of coordinate NPs?

Now consider the question of what the PER value of coordinate NPs is. Choice of verb form does not usually help very much in determining the person of the subject, because those whose AGR value is *non-3sing* are compatible with a subject of any person (except those whose AGR is *3sing*).

However, there is another way to detect the person of the subject NP. If the VP contains a direct object reflexive pronoun, then (as we saw in Chapter 1) the reflexive must agree in person and number with the subject. This co-occurrence pattern is shown by the following examples.

(iv) $\left\{\begin{array}{l}\text{You}\\ \text{*I}\\ \text{*She}\\ \text{*They}\\ \text{*We}\end{array}\right\}$ distinguished yourself. (2nd person singular)

(v) $\left\{\begin{array}{l}\text{She}\\ \text{*You}\\ \text{*I}\\ \text{*They}\\ \text{*We}\end{array}\right\}$ distinguished herself. (3rd person singular)

(vi) $\left\{\begin{array}{l}\text{We}\\ \text{*You}\\ \text{*I}\\ \text{*They}\\ \text{*She}\end{array}\right\}$ distinguished ourselves. (1st person plural)

(vii) $\left\{\begin{array}{l}\text{They}\\ \text{*We}\\ \text{*You}\\ \text{*I}\\ \text{*She}\end{array}\right\}$ distinguished themselves. (3rd person plural)

In light of this patterning, we can now consider the person of coordinate NPs by examining examples like the following:

(viii) You and she distinguished $\left\{\begin{array}{l}\text{yourselves}\\ \text{*themselves}\\ \text{*ourselves}\end{array}\right\}$.

(ix) You and I distinguished $\left\{\begin{array}{l}\text{*yourselves}\\ \text{*themselves}\\ \text{ourselves}\end{array}\right\}$.

B. Construct further examples of sentences with coordinate subjects (stick to the conjunction *and*) that could help you discover what the person value of the coordinate NP is for every combination of PER value on the conjuncts. State the principles for determining the PER value of a coordinate NP in as general terms as you can.

Problem 10: Case and Coordination

There is considerable variation among English speakers about case marking in coordinate NPs. Consult your own intuitions (or those of a friend, if you are not a native English speaker) to determine what rule you use to assign case to pronouns in coordinate structures.

- Start by carefully constructing the right examples that will bear on this issue (the pronouns have to show a case distinction, for example, and there are different syntactic environments to consider).
- In examining the relevant data, be sure you consider both acceptable and unacceptable examples in support of your rule.
- State the rule informally – that is, give a succinct statement, in English, of a generalization that covers case in coordinate NPs in your dialect.

5

Semantics

5.1 Introduction

Our first example of syntactic argumentation in Chapter 1 was the distribution of reflexive and nonreflexive pronouns. In Chapter 7 we will return to this topic and show how it can be analyzed in the grammar we are developing. Before we can do so, however, we need to consider the nature of reference and coreference – topics that are fundamentally semantic in nature (i.e. that have to do in large part with meaning). And before we can do that, we need to discuss meaning more generally, sketching how to represent meaning in our grammar.

Reflexive pronouns provide perhaps the clearest case in which a semantic factor – coreference, in this case – plays an essential role in the grammatical distribution of particular words. But there are many other syntactic phenomena that are closely linked to meaning. Consider, for example, subject-verb agreement, which we have discussed extensively in the past two chapters. The NUM value of a noun is often predictable from its referent. Singular nouns generally refer to individual objects, and plural nouns normally refer to collections of objects. Mass nouns (which are mostly singular) usually refer to substances – that is, entities that are not naturally packaged into discrete objects. Of course, nature doesn't fully determine how the world should be divided up conceptually into objects, collections, and substances, so there may be differences between languages, or even between individuals, as to how things are referred to. Hence the German word *Hose* means essentially the same thing as English *pants* or *trousers*, but the German is singular while the English is plural. Likewise, the French use the plural noun *cheveux* to refer to the same stuff that we call *hair*. And individual English speakers differ as to whether they can use *lettuce* as a count noun. Although the correspondences are usually imperfect, syntactic properties (including such basic ones as the part-of-speech distinctions) are often closely linked to semantic characteristics. Trying to do syntax without acknowledging the associated semantic regularities would lead to missing many fundamental generalizations about linguistic structure.

The study of meaning is at least as old as the study of grammar, and there is little hope of doing justice to problems of semantics in a textbook whose primary concern is grammatical structure. However, if the grammars we develop are going to play any role in modeling real language use, then grammar minimally has to include some information about the meaning of individual words and a treatment of how these combine with each other – that is, an account of how meanings of phrases and sentences are built up from the meanings of their parts. Let us begin by contemplating the nature of sentence meaning.

5.2 Semantics and Pragmatics

Meaning is inextricably bound up with actions – people use language intentionally to do many kinds of things. Some sentences are conventionally used to query; others to make simple assertions; still others are conventionally used to issue commands. Even a piece of a sentence, say an NP like *the student sitting behind Leslie*, can be used in isolation to perform the communicative act of referring to an individual.

The kind of meaning that a sentence can be used to convey depends crucially on its syntactic form. For example, a simple 'inverted' sentence like (1), with an auxiliary verb before the subject NP, is typically used to make a query:

(1) Is Sandy tall?

And the query posed by uttering (1) is closely related to the assertion made by an utterance of the noninverted sentence in (2):

(2) Sandy is tall.

In fact, uttering (2) is a perfectly good way of answering (1).

These observations about communication, or language use, have led researchers to the view that the conventional meanings of different kinds of sentences are different kinds of abstract objects. A declarative sentence like (2), for example, is usually associated with something called a PROPOSITION. A proposition is the kind of thing you can assert, deny, or believe. It is also something (the only kind of thing) that can be true or false. An interrogative sentence like (1) is associated with a semantic object called a QUESTION. Questions are the kind of thing that can be asked and answered. Similarly, we'll call the semantic object associated with an imperative sentence a DIRECTIVE. This is the kind of object that can be issued (by simply uttering an imperative sentence, for example), and fulfilled (by causing the conditions associated with the sentence to be met). Semantics is the study of abstract constructs like propositions, questions and directives, which are assumed to play a key role in a larger theory of communication.[1]

Semantic analysis provides just one part of the account of what people convey when they communicate using language, though. In this text, we make the standard assumption that communication has two components: linguistic meaning (as characterized by semantic analysis) and reasoning about communicative goals. When a linguistic expression is uttered, its linguistic meaning makes a significant contribution to, but does not fully determine, the communicative function of the utterance.

Consider, for example, an utterance of (3):

(3) Do you have a quarter?

As noted above, we take the linguistic meaning of this sentence to be a particular question. Once the identity of the hearer is determined in the relevant context of utterance, a

[1] When speaking informally, we will sometimes talk of a given sentence as conveying a given message (proposition, question, or directive). What we really mean is that our semantic analysis associates a particular message with a given sentence and that the communicative potential of that sentence (what it can be used to convey) is determined in large part by that message.

question of this form has a determinate answer: yes or no. However, an utterance of (3) might serve to communicate much more than such a simple factual inquiry. In particular, in addition to posing a financial query to a given hearer, an utterance of (3) is likely to convey a further message – that the speaker was making the following request of the hearer:

(4) Please give me a quarter!

The question asked by an utterance of (3) is generally referred to as its literal or conventional meaning. A request like (4) is communicated by inference. Asking a certain question (the literal meaning of the interrogative sentence in (3)) in a certain kind of context can lead a hearer to reason that the deeper communicative goal of the speaker was to make a particular request, i.e. the one conveyed by (4). In a different context, i.e. a parent asking (3) of a child standing in a line of children waiting to pay a twenty-five cent admission fee for an amusement park ride, would not lead the hearer to infer (4), but rather to check to make sure that (s)he had the required admission fee. We will leave the account of such embellished communication (even the routine 'reading between the lines' that occurs more or less effortlessly in cases like this) to a more fully developed theory of language use, that is, to a theory of linguistic pragmatics. The inference from query to request is pragmatic in nature.

By contrast, the fact that a sentence like (3) must express a question as its literal meaning is semantic in nature. Semantics is the study of linguistic meaning, that is, the contribution to communication that derives directly from the conventions of the language. Pragmatics is a more general study, of how linguistic meaning interacts with situational factors and the plans and goals of conversational participants to achieve more subtle, often elaborate communicative effects.

The semantic analysis that a grammar provides serves as input for a theory of pragmatics or language use. Such a theory sets as its goal to explain what actually gets communicated via pragmatic inferences derived from the linguistic meaning of an utterance. For example, pragmatic theory might include a principle like (5):[2]

(5) Quantity Principle (simplified)

If X is weaker than Y, then asserting X implies the denial of Y.

This principle leads to pragmatic inference via 'proofs' of the following kind (justifications for steps of the proof are given in parentheses):

(6)
- A says to B: *Two things bother Pat.*
- A uttered something whose linguistic meaning is: 'At least two things bother Pat'. (semantic analysis)[3]

[2]The principle in (5), due to Grice (1989), relies on the undefined term 'weaker'. In some cases (such as the example that follows), it is intuitively obvious what 'weaker' means. But a full-fledged pragmatic theory that included (5) would have to provide a precise definition of this term.

[3]Note that the meaning of the word *two* is no stronger than the 'at least two' meaning, otherwise the following would be contradictory:

(i) [Kim: Do you have two dollars?]

Sandy: Yes, I have two dollars. In fact, I have five dollars.

- ‘At least two things bother Pat’. is weaker than ‘At least three things bother Pat’. (This is true in the context; possibly true more generally)
- B assumes that A also meant to communicate: ‘It’s not the case that there are three things that bother Pat’. (Quantity Principle)

Note that exactly the same pragmatic inference would arise from an utterance by A of any semantically equivalent sentence, such as *There are two things that bother Pat* or *Pat is bothered by two things*. This is because pragmatic theory works from the linguistic meaning of an utterance (as characterized by our semantic analysis) and hence is indifferent to the form by which such meanings are expressed.[4]

There is much more that could be said about the fascinating topic of pragmatic inference. Here, our only goal has been to show that the semantic analysis that must be included in any adequate grammar plays an essential role, albeit an indirect one, in explaining the communicative function of language in context.[5]

5.3 Linguistic Meaning

5.3.1 Compositionality

In order to even begin to deal with semantic issues like

- Which proposition is conveyed by a given declarative sentence?
- Which question is conveyed by a given interrogative sentence?

we first have to clarify what smaller semantic units propositions and questions are constructed from. Moreover, we will need to formulate constraints that specify how the meaning of a given sentence is determined by the meanings of its parts and the way that they are combined.

When we ask a question, make an assertion, or even issue a command, we are also making reference to something that is often called a SITUATION or EVENT.[6] If you utter

[4]This is not quite true. Sometimes the manner in which something is said (the form of an utterance) can make some pragmatic contribution to an utterance. Grice’s theory also included a ‘Maxim of Manner’, which was intended to account for such cases, e.g. (i):

(i) X produced a series of sounds that corresponded closely with the score of ‘Home sweet home’.

Here, A conveys that there was something deficient in X’s rendition of the song. A does this by intentionally avoiding the more concise sentence: *X sang ‘Home sweet home’.*

[5]There is more to meaning than the literal meanings and pragmatic inferences that we have discussed in this section. In particular, there are contrasts in form that correspond to differences in when it is appropriate to use a sentence. One such contrast involves ‘honorific’ forms in Japanese and other languages. The difference between (i) and (ii), is that (i) is familiar and (ii) is formal, so that (i) would be used when talking to a friend or subordinate and (ii) would be used when talking to a stranger or someone higher in a social hiearchy:

(i) Hon-wo yonda.
Book-ACC read.PAST.FAMILIAR
‘I read a book.’

(ii) Hon-wo yomimashita.
Book-ACC READ.PAST.FORMAL
‘I read a book.’

[6]Although the term ‘event’ is often used in a general sense in semantic discussions, this terminology can be misleading, especially in connection with circumstances like the following, where nothing very event-like is happening:

a declarative sentence like *Kim is running*, for example, you are claiming that there is some running situation in the world that involves something (usually a person) named Kim. The proposition that you assert is either true or false depending on a number of things, for example, whether this situation is a running event (maybe Kim is moving too slowly for it to really qualify as running), or whether the runner is someone named 'Kim' (maybe the person you have in mind is really named 'Nim'), whether the running situation is really happening now (maybe Kim has already run the race but your watch stopped several hours ago). If any of these 'maybes' turns out to be the case, then the proposition you have asserted is false – the situation you are describing as specified by the linguistic meaning of the sentence is not part of the real world.

An important part of the business of semantics is specifying truth conditions such as these, that is, specifying restrictions which must be satisfied by particular situations in order for assertions about them to be true. Consider what this means in the case of *Kim is running*. This sentence is associated with a proposition that has the following truth conditions:[7]

(7) a. there is a situation s
 b. s is a running situation
 c. the runner is some individual i
 d. i is named Kim
 e. s is temporally located around the time of utterance

If there is some situation s and some individual i such that all the conditions in (7) are satisfied, then the proposition expressed by *Kim is running* is true. If not, then that proposition is false.

Truth conditions are determined in large part by linguistic meaning, that is, the meaning associated with a sentence by the semantic component of the grammar. If our grammar consisted merely of a list of sentences, we could list the meanings of those sentences alongside their forms. However, as we saw in Chapter 2, lists do not provide plausible theories of the grammars of natural languages. Instead, we've developed a theory of grammar that allows us to systematically build up phrases and sentences from an inventory of words and phrase structure rules. Therefore we will need a semantic component to our grammar that systematically builds the meanings of sentences out of the meanings of words and the way they are put together (i.e. the phrase structure rules). In order to do this, we will need (i) some way of characterizing the linguistic meanings of words and (ii) a set of constraints that allows us to correctly specify the

(i) Bo knows baseball.
(ii) Dana is aggressive.
(iii) Sydney resembles Terry.
(iv) Chris is tall.
(v) 37 is a prime number.

It seems much more intuitive to discuss such sentences in terms of 'situations'; hence we have adopted this as our official terminology for the semantics of sentences.

[7]The exact meaning of the progressive (*be...-ing*) construction is a fascinating semantic topic with a considerable literature that we cannot do justice to here. We have adopted clause (7e) as a convenient first approximation of the truth conditional contribution of the present progressive in English.

linguistic meanings of phrase structures in terms of the meanings of their parts (their subconstituents).

In terms of the example *Kim is running*, we will need a way to ensure that the various pieces of this sentence – the noun *Kim*, the verb *is*, and the verb *running* – each make their appropriate contribution to the set of constraints summarized in (7), that the result is a proposition (not a question or a directive), and that the pieces of meaning get combined in the appropriate way (for example, that the same individual i has the properties of being named Kim and being the runner). In addition, our account must assign a meaning to *Sandy is running* that differs from that assigned to *Kim is running* only in the name of the individual i. Likewise, our account must analyze the sentence *Is Kim running?* as a question, and furthermore a question about whether or not there is a situation s and an individual i such that all the conditions in (7) are satisfied.

5.3.2 Semantic Features

The semantic objects of our grammar will be classified in terms of four SEMANTIC MODES – that is, the four basic kinds of meanings that are enumerated and illustrated in (8):

(8)

SEMANTIC MODE	KIND OF PHRASE	EXAMPLE
proposition	noninverted sentence	*Kim is happy.*
question	inverted sentence	*Is Kim happy?*
directive	imperative sentence	*Be happy!*
reference	NP	*Kim*

As we saw above, there are a number of differences among the various semantic modes. Despite these differences, the modes have something in common. Every kind of linguistic expression we have considered, irrespective of its semantic mode, refers to something that must satisfy an indicated list of restrictions for the expression to be correctly applicable. To express this generalization, we will model all expressions in terms of a single type of semantic object (a *sem-cat* or *semantic-category*) which bears three features: MODE, INDEX, and RESTR. The value of MODE provides the semantic mode of the object. The value of INDEX is an index corresponding to the situation or individual referred to. The value of RESTR (short for 'restriction') is a list of conditions that the situation or individual has to satisfy in order for the expression to be applicable to it. Semantic structures then will look like (9):

(9)
$$\begin{bmatrix} \textit{sem-cat} & \\ \text{MODE} & \left\{\text{prop, ques, dir, ref, none}\right\} \\ \text{INDEX} & \left\{i,\ j,\ k,\ \ldots,\ s_1,\ s_2,\ \ldots\right\} \\ \text{RESTR} & \langle\ \ldots\rangle \end{bmatrix}$$

There are a couple of things to note about the values of these features. The first is that, although we represent the value of RESTR as a list, the order of the elements on that list will not be semantically significant. The second is that the feature INDEX differs from other features we have encountered, in that it can take an unlimited number of different values. This is because there is no limit (in principle) to the number of different individuals or situations which can be referred to in a single sentence. Consequently, we must have (in principle, at least) an infinite number of indices available to serve as values of the feature INDEX. These values of INDEX will conventionally be written with lower-case letters; instead of tagging two occurrences of the same INDEX value, we will simply write the same lower-case letter in both places.

Propositions are analyzed in terms of feature structures like the one in (10) (where 'prop' is short for 'proposition').

(10) $\begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s \\ \text{RESTR} & \langle \ ... \ \rangle \end{bmatrix}$

A proposition like (10) will be true just in case there is some actual situation s (and there exist appropriate other individuals corresponding to whatever indices are present in (10)) such that the constraints specified in the RESTR value of (10) are all satisfied. These restrictions, the nature of which will be explained in Section 5.3.3, must include all those that are relevant to the meaning of the sentence, for example, all the constraints just mentioned in conjunction with the truth or falsity of *Kim is running*. Our grammatical analysis needs to ensure that we end up with exactly the right constraints in the RESTR list of a sentence's semantics, so that we associate exactly the right meaning with any sentence sanctioned by our grammar.

A question like *Is Kim running?* is assigned a semantics just like the one assigned to *Kim is running*, except that the MODE value must be 'question' ('ques' for short), rather than 'prop':

(11) $\begin{bmatrix} \text{MODE} & \text{ques} \\ \text{INDEX} & s \\ \text{RESTR} & \langle \ ... \ \rangle \end{bmatrix}$

In this case, the value of RESTR is again interpreted as the set of conditions placed on the situation s, but if someone poses a question, they are merely inquiring as to whether s satisfies those conditions.

Directives ('dir' for short) are represented as in (12):

(12) $\begin{bmatrix} \text{MODE} & \text{dir} \\ \text{INDEX} & s \\ \text{RESTR} & \langle \ ... \ \rangle \end{bmatrix}$

What the RESTR list does in the case of a directive is to specify what conditions have to be satisfied in order for a directive to be fulfilled.

A reference ('ref' for short) is similar to the kinds of meanings just illustrated, except that it can be used to pick out all kinds of entities – not just situations. So the semantics we assign to a referring NP has the following form:[8]

[8]There are any number of intriguing referential puzzles that are the subject of ongoing inquiry by semanticists. For example, what does an NP like *a page* refer to in the sentence: *A page is missing from this book*? And what does *the unicorn that Chris is looking for* refer to in the sentence: *The unicorn that Chris is looking for doesn't exist*?

(13) $\begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \langle \ldots \rangle \end{bmatrix}$

In this case, the RESTR list contains the conditions that the entity must meet in order for it to be legitimately referred to by the expression.

Note that we write INDICES in terms of the letters i, j, k, etc. when we are specifying the semantics of nominal expressions. The INDEX values written as s, s_1, s_2, etc. always refer to situations.

The differing values of MODE that we have just seen serve to differentiate between the kinds of meaning that are associated with various syntactic categories (like declarative, interrogative or imperative sentences or noun phrases). Many words and phrases that cannot be used by themselves to express a proposition, ask a question, refer to an individual, etc. (e.g. determiners and conjunctions) will be treated here in terms of the specification [MODE none].

5.3.3 Predications

We now turn to the question of what kind of entities make up the value of the RESTR list. Semantic restrictions associated with expressions come in many varieties, which concern what properties some individual has, who did what to whom in some situation, when, where, or why some situation occurred, and so forth. That is, semantically relevant restrictions specify which properties must hold of individuals and situations, and which relations must hold among them, in order for an expression to be applicable.

To represent this sort of information, we must introduce into our semantics some way of specifying relations among entities quite generally. We do this by introducing a type of feature structure called *predication*. The features of a *predication* specify (i) what kind of relation is involved and (ii) who or what is participating in the relation. Examples of feature structures of type *predication* are given in (14):[9]

(14) a. $\begin{bmatrix} \textit{predication} & \\ \text{RELN} & \textbf{love} \\ \text{SIT(UATION)} & s \\ \text{LOVER} & i \\ \text{LOVED} & j \end{bmatrix}$ b. $\begin{bmatrix} \textit{predication} & \\ \text{RELN} & \textbf{walk} \\ \text{SIT} & s \\ \text{WALKER} & i \end{bmatrix}$

[9]The kind of event-based semantic analysis we employ was pioneered by the philosopher Donald Davidson in a number of papers. (See, for example, Davidson 1980.) Our simplified representations differ from other work in this tradition where all talk of existence is represented via explicit existential quantification, i.e. in terms of representations like (i):

(i) there is an event s and an individual i such that: s is a running event, the runner of s is i, i is named Kim, and s is temporally located around the time of utterance

We will treat all such existential quantification as implicit in our semantic descriptions.

c. $\begin{bmatrix} \textit{predication} & \\ \text{RELN} & \textbf{give} \\ \text{SIT} & s \\ \text{GIVER} & i \\ \text{RECIPIENT} & j \\ \text{GIFT} & k \end{bmatrix}$ d. $\begin{bmatrix} \textit{predication} & \\ \text{RELN} & \textbf{book} \\ \text{SIT} & s \\ \text{INST} & k \end{bmatrix}$

e. $\begin{bmatrix} \textit{predication} & \\ \text{RELN} & \textbf{happy} \\ \text{SIT} & s \\ \text{INST} & i \end{bmatrix}$ f. $\begin{bmatrix} \textit{predication} & \\ \text{RELN} & \textbf{under} \\ \text{SIT} & s \\ \text{LOWER} & i \\ \text{HIGHER} & j \end{bmatrix}$

The predications in (14) are meant to correspond to conditions such as: 's is a situation wherein i loves j', 's is a situation wherein i walks', 's is a situation wherein i gives k to j', 's is a situation wherein k is an instance of bookhood (i.e. where k is a book)', 's is a situation wherein i is happy', and 's is a situation wherein i is under j', respectively. We will henceforth make frequent use of predications like these, without taking the time to present a proper theory of relations, predications, and the features that go with them. Note that the restriction associated with many nouns and adjectives (*book*, *happy*, etc.) includes a predication of only one (nonsituation) argument. In such cases – for example, (14d,e) – we use the feature INST(ANCE).

As indicated in (14), we are assuming that all predications are in principle 'situated', i.e. that they make reference to some particular situation (the index that is the value of the feature SIT inside each predication). This provides a semantic flexibility that allows us to analyze sentences like (15):

(15) The senator visited a classmate a week before being sworn in.

That is, one way to understand this (perhaps the most natural way) is in terms of the proposition that some person i who is now a senator was part of a visiting situation where the person who got visited – j – was once part of a certain academic situation that also included the senator. The three situations are all distinct: the situation where i instantiates senatorhood comes after the visiting situation and both these situations could come long after the situation where i and j were classmates. Yet the proposition expressed by (15) is making reference to all three situations at once, and the situational predications we have assumed give us a way to model this.[10] Though this use of multiple situations in the semantics of a single proposition is fascinating and may well be essential for semantic analysis to be successful,[11] secondary situations bring unwanted complexity

[10]Of course, sometimes we refer to someone as a senator even after they have left office. This could be analyzed as making reference to a past situation in which the individual referred to instantiated senatorhood.

[11]There is, of course, an issue as to how far to take the situation-based kind of analysis. General statements like *All cows eat grass* or *Two plus two is four* seem not to make reference to any particular situations.

and hence will be suppressed in subsequent discussion, unless they bear directly on a particular discussion. In general, we will only display the SIT feature on predications contributed by the head of a given phrase or when its value is identified with the value of some other feature.

Almost all words specify restrictions that involve predications of one kind or another, including verbs, adjectives, adverbs, prepositions, and nouns. In order for phrases containing such words to inherit these restrictions, there must be constraints that (minimally) guarantee that the RESTR values of a phrase's daughters are part of that phrase's RESTR value. Only in this way will we end up with a sentence whose meaning is a proposition (or question or directive) whose RESTR value includes all the necessary restrictions on the relevant event participants.

For example, we will want our grammar to ensure that a simple sentence like (16) is associated with a proposition like the one described in (17):

(16) Chris saved Pat.

(17)
$$\begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{save} \\ \text{SIT} & s \\ \text{SAVER} & i \\ \text{SAVED} & j \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{Chris} \\ \text{NAMED} & i \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{Pat} \\ \text{NAMED} & j \end{bmatrix} \right\rangle \end{bmatrix}$$

The restriction that s is a saving situation comes from the lexical entry for the verb *save*, the constraint that i – the saver – must be named Chris comes from the proper noun *Chris*, and the constraint that j – the saved (person) – must be named Pat comes from the lexical entry for the proper noun *Pat*. By associating (16) with the feature structure in (17), our semantic analysis says that the linguistic meaning of (16) is the proposition that will be true just in case there is an actual situation that involves the saving of someone named Pat by someone named Chris. But in order to produce the right set of restrictions in the sentence's semantic description, the restrictions of the parts of the sentence have to be amalgamated into a single list of restrictions. Note in addition that the main situation of the sentence is derived from that introduced by the verb. It is true in general that the semantics of a phrase will crucially involve the semantics of its head daughter. We will capture these semantic relationships between the parts of the sentence with two general principles, introduced in Section 5.5 below. First, however, we must consider how semantic structures fit into the tree structures our grammar licenses.

5.4 How Semantics Fits In

In earlier chapters, we considered only the syntactic properties of linguistic expressions. To accommodate the basic analysis of linguistic meaning just introduced, we need some way of introducing semantic structures into the feature structures we use to analyze words and phrases. We do this by adding two new features – SYN(TAX) and SEM(ANTICS) – and adding a level of embedding within our feature structures, as illustrated in (18):

(18)
$$
\begin{bmatrix}
\textit{expression} & \\
\text{SYN} & \begin{bmatrix} \textit{syn-cat} & \\ \text{HEAD} & \begin{bmatrix} ... \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & ... \\ \text{COMPS} & ... \end{bmatrix} \end{bmatrix} \\
\text{SEM} & \begin{bmatrix} \textit{sem-cat} & \\ \text{MODE} & ... \\ \text{INDEX} & ... \\ \text{RESTR} & \langle\ ...\ \rangle \end{bmatrix}
\end{bmatrix}
$$

There is now a syntactic side and a semantic side to all feature structures like (18), i.e. to all feature structures of type *expression*. Note that we have created another type – *syntactic-category* (*syn-cat*) – which is parallel to *sem-cat*, and which classifies the values of the feature SYN, just as *sem-cat* classifies the values of the feature SEM. Although we will add a few more features as we progress, this is in essence the feature geometry that we will adopt in the remainder of the book.

This changes the way lexical entries look, of course; their new feature geometry is illustrated in (19), though some details are not yet included:[12]

(19) a.
$$
\left\langle \text{dog}\ ,
\begin{bmatrix}
\text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \textit{3sing} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ [\text{HEAD}\ \textit{det}]\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\
\text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{dog} \\ \text{INST} & i \end{bmatrix} \right\rangle \end{bmatrix}
\end{bmatrix}
\right\rangle
$$

[12]It should be noted that our semantic analysis of proper nouns (one of many that have been proposed over the centuries) treats them as simple referring expressions whose referent must be appropriately named. In a more precise account, we might add the further condition that the speaker must intend to refer to the referent. Under this analysis, the proposition expressed by a sentence like *Kim walks* would be regarded as true just in case there is a walking event involving a certain individual that the speaker intends to refer to who is named 'Kim'.

b.

$$\left\langle \text{Kim}\ ,\ \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \textit{3sing} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{Kim} \\ \text{NAMED} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

c.

$$\left\langle \text{love}\ ,\ \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \text{NP}\ \rangle \\ \text{COMPS} & \langle\ \text{NP[acc]}\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{love} \\ \text{SIT} & s \\ \text{LOVER} & i \\ \text{LOVED} & j \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

These entries also illustrate the function of the INDEX feature in fitting together the different pieces of the semantics. Notice that the INDEX value of *love* is identified with the SIT argument of the loving predication in its RESTR list. Similarly, the INDEX value of *dog* is the same as the INST value in the predication introduced by *dog*, and that the INDEX value of *Kim* is the same as the NAMED value in the predication introduced by *Kim*. By identifying these values, we enable the NPs to 'expose' those indices to other words that might select the NPs as arguments. Those words, in turn, can associate those indices with the appropriate role arguments within their predications (i.e. features like WALKER, LOVED, etc.). This is illustrated in (20) for the verb *love*:

(20)

$$\left\langle \text{love}\ , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \begin{array}{c}\text{NP}\\ \begin{bmatrix}\text{INDEX } i\end{bmatrix}\end{array}\right\rangle \\ \text{COMPS} & \left\langle \begin{array}{c}\text{NP}\\ \begin{bmatrix}\text{CASE acc}\\ \text{INDEX } j\end{bmatrix}\end{array}\right\rangle \end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop}\\ \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix}\text{RELN} & \textbf{love}\\ \text{SIT} & s\\ \text{LOVER} & i\\ \text{LOVED} & j\end{bmatrix}\right\rangle\end{bmatrix}\end{bmatrix}\right\rangle$$

In this way, as the verb combines with a particular NP object, the index of that NP is identified with the value of the feature LOVED in the verb's semantics. Likewise, since the verb's specifier requirement is identified with the VP's specifier requirement (by the Valence Principle), when the VP combines with a particular NP subject, the index of that NP will be identified with the value of the feature LOVER in the verb's semantics. All that is left is to ensure that the predications introduced by each word are collected together to give the RESTR list of the whole sentence, and to ensure that the INDEX and MODE values of phrases are appropriately constrained. These are the topics of the next section.

Note that the addition of semantic information to our grammar has changed the way we use abbreviations in two ways. First, the labels NP, S, V, etc. now abbreviate feature structures that include both semantic and syntactic information, i.e. *expression*s which bear the features SYN and SEM. Second, we will add a notation to our system of abbreviations to allow us to refer to the INDEX value of an abbreviated expression: NP_i will be used as a shorthand for an NP whose SEM value's INDEX is i. We occasionally use this same subscript notation with other categories, too, e.g. PP_i. (The abbreviations are summarized in the grammar summary in Section 5.10.)

5.5 The Semantic Principles

We are now not only able to analyze the form of sentences of considerable complexity using our grammar, but in addition we can analyze the meanings of complex sentences by adding semantic constraints on the structures defined by our rules. The most general of these semantic constraints is given in (21):

(21) Semantic Compositionality Principle

In any well-formed phrase structure, the mother's RESTR value is the sum of the RESTR values of the daughters.

In other words, all restrictions from all the daughters in a phrase are collected into the RESTR value of the mother. The term ‘sum’ has a straightforward meaning here: the sum of the RESTR values of the daughters is the list whose members are those values, taken in order.[13] We will use the symbol ‘$\oplus$’ to designate the sum operator.[14]

In addition to the Semantic Compositionality Principle, we introduce the following constraint on the MODE and INDEX values of headed phrases:

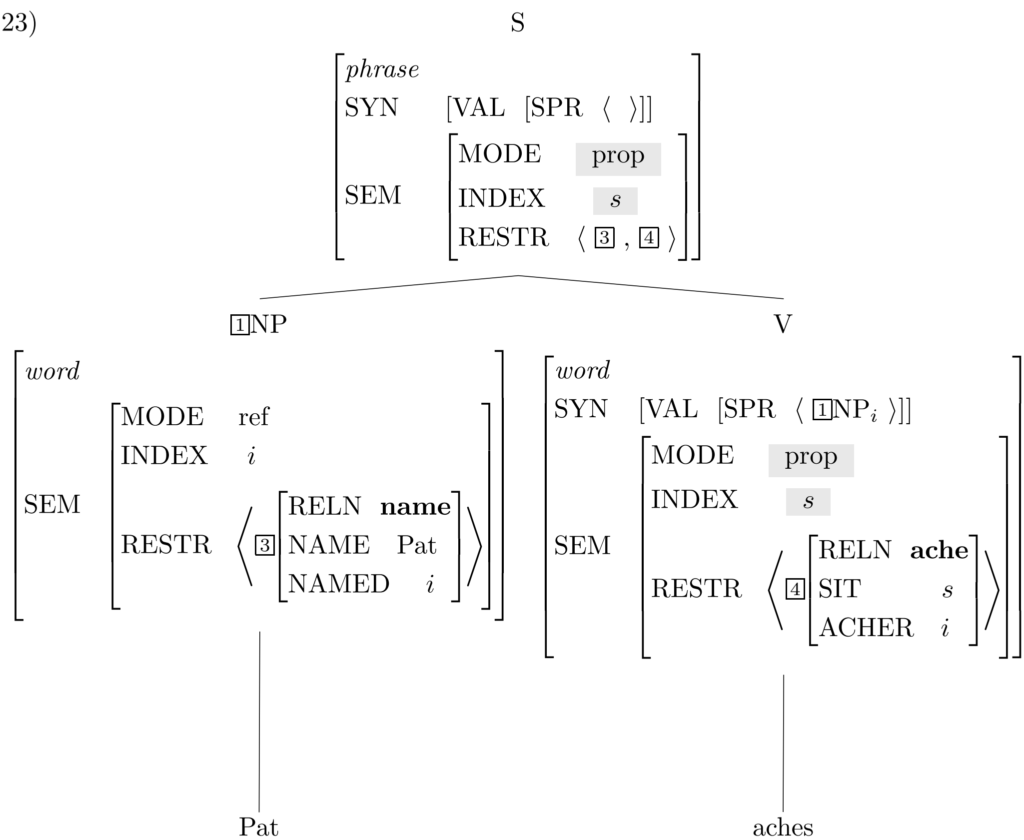
(22) Semantic Inheritance Principle

In any headed phrase, the mother’s MODE and INDEX values are identical to those of the head daughter.

The Semantic Inheritance Principle guarantees that the semantic MODE and INDEX of a phrase are identified with those of the head daughter, giving the semantics, like the syntax, a ‘head-driven’ character.

The effect of these two semantic principles is illustrated in the simple example, (23):

(23)



[13] That is, the sum of lists ⟨ A ⟩, ⟨ B, C ⟩, and ⟨ D ⟩ is the list ⟨ A, B, C, D ⟩.

[14] Notice that, unlike the familiar arithmetic sum operator, $\oplus$ is not commutative: ⟨ A ⟩ $\oplus$ ⟨ B ⟩ = ⟨ A, B ⟩, but ⟨ B ⟩ $\oplus$ ⟨ A ⟩ = ⟨ B, A ⟩. And ⟨ A, B ⟩ $\neq$ ⟨ B, A ⟩, because the order of the elements matters. Although, as noted above, the order of elements in RESTR lists has no semantic significance, we will later use $\oplus$ to construct lists in which the ordering does matter (specifically, the ARG-ST lists introduced in Chapter 7 as part of our account of reflexive binding).

The effect of both semantic principles can be clearly observed in the S node at the top of this tree. The MODE is 'prop', inherited from its head daughter, the V node *aches*, by the Semantic Inheritance Principle. Similarly (as indicated by shading in (23)), the INDEX value *s* comes from the verb. The RESTR value of the S node, [RESTR $\langle$ $\boxed{3}$, $\boxed{4}$ $\rangle$], is the sum of the RESTR values of the NP and VP nodes, as specified by the Semantic Compositionality Principle.

In this way, our analysis provides a general account of how meanings are constructed. The Semantic Compositionality Principle and the Semantic Inheritance Principle together embody a simple yet powerful theory of the relation between the structures of our grammar and the meanings they convey.

5.6 Modification

The principles in Section 5.5 account for the semantics of head-complement and head-specifier phrases. We still need to consider the Coordination Rule (which, as a non-headed rule, isn't subject to the Semantic Inheritance Principle) and the Head-Modifier Rule, which hadn't yet reached its final form in the Chapter 4 grammar. This section addresses the Head-Modifier Rule. The Coordination Rule will be the subject of the next section.

The Head-Modifier Rule of the Chapter 4 grammar looked like this:

(24) Head-Modifier Rule (Chapter 4 version)

$$\begin{bmatrix} \textit{phrase} \end{bmatrix} \rightarrow \mathbf{H}\begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \ \text{PP}$$

The only kind of modifier this rule accounts for is, of course, PPs. We'd like to extend it to adjectives and adverbs as well. Adverbs and adjectives, however, present a complication. Compared to PPs, they are relatively fussy about what they will modify. Adverbs modify verbs and not nouns (as illustrated in (25)) and adjectives modify nouns, but not verbs, (as illustrated in (26)).

(25) a. A rat died yesterday.
b.*A rat yesterday died.

(26) a. The person responsible confessed.
b.*The person confessed responsible.

In order to capture these facts, we introduce a feature called MOD which will allow modifiers to specify what kind of expressions they can modify. The value of MOD will be a (possibly empty) list of *expression*s. For elements that can be modifiers, this list contains just one *expression*. For elements that can't be modifiers, the list is empty. This allows us to make it a lexical property of adjectives that they are [MOD $\langle$ NOM $\rangle$] (or [MOD $\langle$ NP $\rangle$]) and a lexical property of adverbs that they are [MOD $\langle$ VP $\rangle$] (or [MOD $\langle$ S $\rangle$]).

MOD will be a VAL feature, like SPR and COMPS. The intuitive connection between these three features is that they all specify what the head can combine with, although the means of combination is somewhat different for MOD as opposed to SPR and COMPS. Like SPR and COMPS, MOD is passed up from the head daughter to the mother via the Valence Principle, as adjusted in (27):

(27) The Valence Principle
Unless the rule says otherwise, the mother's values for the VAL features (SPR, COMPS, MOD) are identical to those of the head daughter.

Unlike with SPR and COMPS, no rule will contradict the Valence Principle with respect to the value of MOD. This means that the MOD value of the mother will always be the same as the MOD value of the head daughter. This is desirable, as the kind of expression a phrasal modifier (such as *responsible for the mess* or *on the table*) can modify is determined by the head of the modifier (in this case, the adjective *responsible* or the preposition *on*).

Furthermore, MOD, like SPR and COMPS, must be shared between conjuncts in a coordinate structure. If it weren't, we would mistakenly license ungrammatical strings such as those in (28):

(28) a.*The cat slept soundly and furry.
b.*The soundly and furry cat slept.

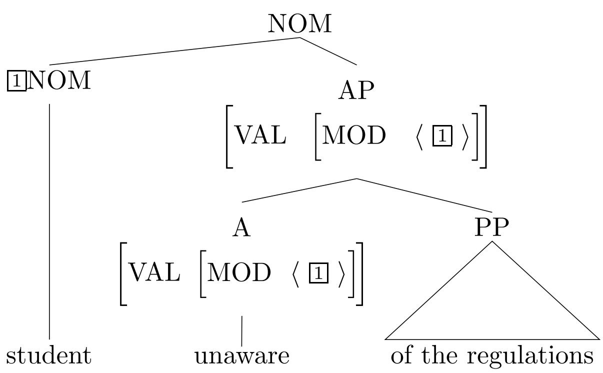
Since the Coordination Rule identifies the VAL values of the conjuncts, making MOD a VAL feature immediately captures these facts.

With modifiers now specifying what they can modify, the Head-Modifier Rule can be reformulated as in (29):[15]

(29) Head-Modifier Rule (Near-Final Version)

$$[\textit{phrase}] \rightarrow \mathbf{H}\boxed{1}\begin{bmatrix}\text{VAL} & \begin{bmatrix}\text{COMPS} & \langle\ \rangle\end{bmatrix}\end{bmatrix} \begin{bmatrix}\text{VAL} & \begin{bmatrix}\text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle \boxed{1} \rangle\end{bmatrix}\end{bmatrix}$$

The rule in (29) will license a phrase structure tree whose mother is, for example, a NOM just in case the head daughter is an expression of category NOM and the modifier daughter's MOD value is also of category NOM:

(30)


This NOM can combine with a determiner as its specifier to build an NP like (31):

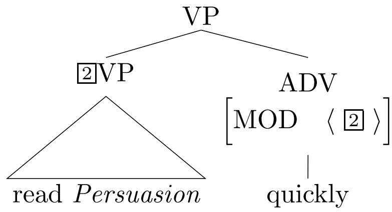
[15]In this rule, and in the A and AP nodes of (30), we have omitted the feature name 'SYN' to the left of 'VAL'. In the remainder of the book, we will often simplify our feature structure descriptions in this way, leaving out some of the outer layers of feature names when the information of interest is embedded within the feature structure description. We will only simplify in this way when no ambiguity about our intended meaning can arise.

This is the 'near-final version' of the Head-Modifier Rule. It will receive a further minor modification in Chapter 14.

(31) a student unaware of the regulations

The Head-Modifier Rule in (29) will also license the verb phrase in (32), under the assumption that adverbs are lexically specified as [MOD ⟨ VP ⟩]:

(32)



And a VP satisfying this description can combine with a subject like the one in (31) to build sentence (33):

(33) A student unaware of the regulations read *Persuasion* quickly.

Note that the value of MOD is list of *expression*s, which contain semantic as well as syntactic information. This will allow us to give an analysis of how the semantics of modifiers work. We will illustrate this analysis with the sentence in (34):

(34) Pat aches today.

Let us assume that an adverb like *today* has a lexical entry like the one in (35):[16] (We assume here that there is a subtype of *pos* for adverbs (*adv*).)

(35)

$$\left\langle \text{today}\ ,\ \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{adv} \\ \text{VAL} & \begin{bmatrix} \text{MOD} & \left\langle \begin{matrix}\text{VP}\\ \begin{bmatrix}\text{INDEX} & s_1\end{bmatrix}\end{matrix} \right\rangle \\ \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{none} \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{today} \\ \text{ARG} & s_1 \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

The key point here is that the MOD value identifies the index of the VP to be modified as 's_1', the same situation that is the argument of the relation 'today' in the semantic restriction. This means that once the adverb combines with a VP, the (situational) index of that VP is the argument of 'today'.

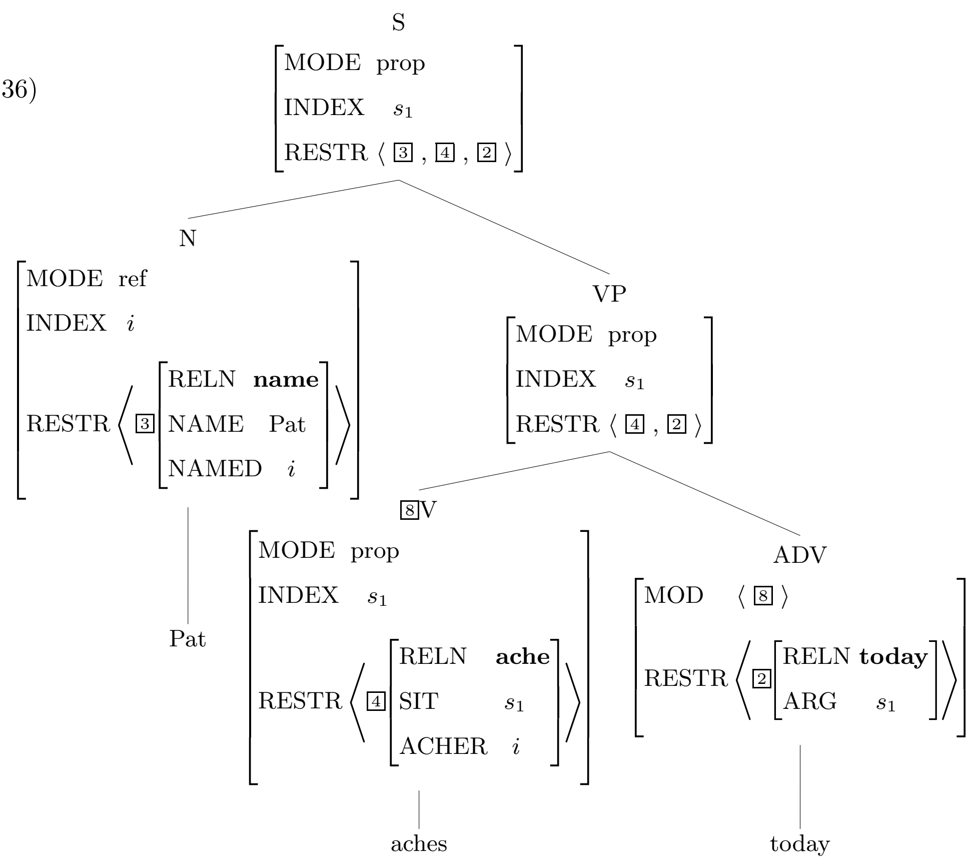
[16] We are suppressing the feature INDEX (along with SIT) here for clarity. For a more detailed analysis of adverbial modification, see Bender et al. 2002.

Exercise 1: The Missing INDEX

We have omitted INDEX from the SEM value in (35), although we said earlier that the value of SEM always consists of MODE, INDEX, and RESTR. Our omission was to simplify the presentation. Including INDEX under SEM would only have cluttered up the feature structure, without adding any useful information. In fact, we could assign any value we want to the missing INDEX, and the semantics of VPs like *aches today* would still be the same. Why?

Our two semantic principles, the Head-Modifier Rule, and the lexical entry in (35) as well as appropriate lexical entries for *aches* and *Pat* thus interact to define structure like (36) (only SEM values are indicated):

(36)



Exercise 2: VP or Not VP?

The lexical entry in (35) has a VP on the MOD list, but the corresponding node in the tree (36) is labeled V. Why isn't this an inconsistency? [*Hint: Remember that VP and V are abbreviations for feature structures, and check what they are abbreviations for.*]

5.7 Coordination Revisited

The analysis of the previous sections specifies how meanings are associated with the headed structures of our grammar, by placing appropriate constraints on those trees that result from our headed rules. It also covers the composition of the RESTR values in nonheaded rules. But nothing in the previous discussion specifies the MODE or INDEX values of coordinate phrases – the kind of phrase licensed by the Coordination Rule, a nonheaded rule.

In the previous chapter, we wrote this rule as follows:

(37)
$$\begin{bmatrix}\text{VAL }\boxed{1}\end{bmatrix} \rightarrow \begin{bmatrix}\text{VAL }\boxed{1}\end{bmatrix}^{+} \begin{bmatrix}\textit{word} & \\ \text{HEAD} & \textit{conj}\end{bmatrix} \begin{bmatrix}\text{VAL }\boxed{1}\end{bmatrix}$$

This is equivalent to the following formulation, where the Kleene plus has been replaced by a schematic enumeration of the conjunct daughters:

(38)
$$\begin{bmatrix}\text{VAL }\boxed{1}\end{bmatrix} \rightarrow \begin{bmatrix}\text{VAL }\boxed{1}\end{bmatrix}_{1} \ldots \begin{bmatrix}\text{VAL }\boxed{1}\end{bmatrix}_{n-1} \begin{bmatrix}\textit{word} & \\ \text{HEAD} & \textit{conj}\end{bmatrix} \begin{bmatrix}\text{VAL }\boxed{1}\end{bmatrix}_{n}$$

We will employ this new notation because it lets us enumerate schematically the arguments that the semantic analysis of conjunctions requires.

Unlike the other predications we have used for semantic analysis, where each predication specifies a fixed (and small) number of roles, the predications that express the meanings of conjunctions like *and* and *or* allow any number of arguments. Thus each conjunct of coordinate structures like the following is a semantic argument of the conjunction:

(39) a. Chris [[walks]$_1$, [eats broccoli]$_2$, and [plays squash]$_3$].

b. [[Chris walks]$_1$, [Pat eats broccoli]$_2$, and [Sandy plays squash]$_3$].

Because the number of arguments is not fixed, the predications for conjunctions allow not just indices as arguments, but lists of indices. Consequently, the sentences in (39) may be represented in terms of a semantic structure like the following:

(40)
$$\begin{bmatrix}\text{MODE} & \text{prop} \\ \text{INDEX} & s_0 \\ \text{RESTR} & \left\langle \begin{bmatrix}\text{RELN} & \textbf{and} \\ \text{SIT} & s_0 \\ \text{ARGS} & \langle s_1 , s_2 , s_3 \rangle\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{walk} \\ \text{SIT} & s_1 \\ & \ldots\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{eat} \\ \text{SIT} & s_2 \\ & \ldots\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{play} \\ \text{SIT} & s_3 \\ & \ldots\end{bmatrix} \right\rangle\end{bmatrix}$$

In (40), the situations s_1, s_2, and s_3 are the simplex situations of walking, eating and playing, respectively. The situation s_0, on the other hand, is the complex situation that involves all three of the simplex situations. Note that it is this situation (s_0) that is the INDEX of the whole coordinated phrase. That way, if a modifier attaches to the coordinated phrase, it will take the index of the complex situation as its semantic argument.

In order to be sure our grammar assigns semantic representations like (40) to sentences like (39), we need to update our lexical entries for conjunctions and revise the Coordination Rule. Let us assume then that the lexical entry for a conjunction looks roughly as shown in (41):

(41)
$$\left\langle \text{and}\ ,\ \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{conj} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{MODE} & \text{none} \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{and} \\ \text{SIT} & s \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

As for the Coordination Rule, we need to revise it so that it relates the indices of the conjuncts to the predication introduced by the conjunction. In addition, we need to say something about the index of the mother. This leads us to the following reformulation of our Coordination Rule (where 'IND' is short for 'INDEX'):

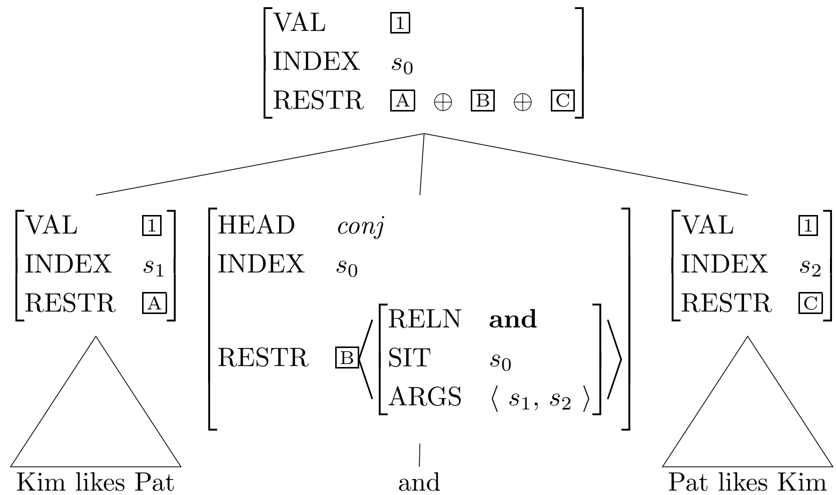
(42) Coordination Rule
$$\begin{bmatrix} \text{SYN [VAL } \boxed{0}] \\ \text{SEM [IND } s_0] \end{bmatrix} \rightarrow \begin{bmatrix} \text{SYN [VAL } \boxed{0}] \\ \text{SEM [IND } s_1] \end{bmatrix} \dots \begin{bmatrix} \text{SYN [VAL } \boxed{0}] \\ \text{SEM [IND } s_{n-1}] \end{bmatrix}$$
$$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD } \textit{conj} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{IND} & s_0 \\ \text{RESTR} & \langle [\text{ARGS } \langle s_1, \dots, s_n \rangle] \rangle \end{bmatrix} \end{bmatrix} \begin{bmatrix} \text{SYN [VAL } \boxed{0}] \\ \text{SEM [IND } s_n] \end{bmatrix}$$

This rule accomplishes a number of goals, including:

- requiring that all conjuncts of a coordinate structure have identical values for SPR, COMPS, and MOD.
- collecting the RESTR values of all daughters into the RESTR list of the mother (guaranteed because the structures built in accordance with this rule must satisfy the Semantic Compositionality Principle),
- identifying the indices of the conjuncts with the semantic arguments of the conjunction, and
- identifying the index of the conjunction with that of the coordinate structure.

These effects are illustrated in the following tree, which shows a (coordinate) phrase structure satisfying the Coordination Rule:

(43)



Our revised Coordination Rule goes a long way toward accounting for sentences containing coordinate structures and associating them with appropriate meanings. We will return to coordination in Chapters 8 and 14 to add further refinements.

5.8 Quantifiers

The final semantic topic we will address in this chapter is quantifiers and quantifier scope ambiguities. Consider the example in (44):

(44) A dog saved every family.

Sentences like this are usually treated as ambiguous, the two distinct READINGS being paraphrased roughly as (45a,b):

(45) a. There was some particular dog who saved every family.
b. Every family was saved by some dog or other (not necessarily the same dog).

Ambiguities of this kind might be familiar from the study of predicate logic, where the two readings in question are often represented in the fashion shown in (46a.b):

(46) a. (**Exist** i: **dog**(i))[(**All** j: **family**(j))[**save**(i,j)]]
b. (**All** j: **family**(j))[(**Exist** i: **dog**(i))[**save**(i,j)]]

The first three parts of these representations are a quantificational relation (e.g. **Exist**, **All**), a variable (e.g. i, j), and a formula called the quantifier's RESTRICTION (e.g. **dog**(i), **family**(j)). The expression in square brackets that follows a quantifier is its SCOPE. In (46a), the scope of the quantifier (**All** j: **family**(j)) is the expression repeated in (47):

(47) [**save**(i,j)]

In the same example, the scope of the quantifier (**Exist** i: **dog**(i)) is the expression repeated in (48):

(48) [(**All** j: **family**(j))[**save**(i,j)]]

The two distinct semantic analyses associated with a sentence like (44) thus differ only in terms of scope: in (46a), the existential quantifier has 'wide' scope; in (46b), the universal quantifier has wide scope.

The semantics we adopt in this book is compatible with recent work on quantification known as the theory of generalized quantifiers. This theory models the interpretation of quantifiers set-theoretically in a way that makes it possible to represent nonstandard quantifiers like '**most**', as well as the standard universal and existential quantifiers of predicate logic. Although our representations look different from those in (46), we can express the notions of quantifier, variable, restriction and scope using feature structures. We achieve this by treating quantifiers in terms of predications like (49):

(49)
$$\begin{bmatrix} \textit{predication} & \\ \text{RELN} & \textbf{exist} \\ \text{BV} & i \\ \text{QRESTR} & \textit{predication} \\ \text{QSCOPE} & \textit{predication} \end{bmatrix}$$

In (49), the quantifier predication has three new features: BOUND-VARIABLE (BV), QUANTIFIER-RESTRICTION (QRESTR) and QUANTIFIER-SCOPE (QSCOPE). The values of the latter two features can be identified with other predications in the RESTR list.

We can then identify the two quantifiers' QSCOPE values in different ways to express the two different scopal readings of (44). If the existential quantifier has wide scope, as in (46a), we can identify the QSCOPE values as shown in (50):

(50)
$$\begin{bmatrix} \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{exist} \\ \text{BV} & i \\ \text{QRESTR} & \boxed{1} \\ \text{QSCOPE} & \boxed{2} \end{bmatrix}, \boxed{1}\begin{bmatrix} \text{RELN} & \textbf{dog} \\ \text{INST} & i \end{bmatrix}, \boxed{2}\begin{bmatrix} \text{RELN} & \textbf{all} \\ \text{BV} & j \\ \text{QRESTR} & \boxed{3} \\ \text{QSCOPE} & \boxed{4} \end{bmatrix}, \right. \\ & \left. \boxed{3}\begin{bmatrix} \text{RELN} & \textbf{family} \\ \text{INST} & j \end{bmatrix}, \boxed{4}\begin{bmatrix} \text{RELN} & \textbf{save} \\ \text{SAVER} & i \\ \text{SAVED} & j \end{bmatrix} \right\rangle \end{bmatrix}$$

And to represent the reading where the universal quantifier outscopes the existential, as in (46b), we can simply identify the QSCOPE values differently, as shown in (51):

(51)
$$\begin{bmatrix} \text{RESTR} & \left\langle \boxed{2}\begin{bmatrix} \text{RELN} & \textbf{exist} \\ \text{BV} & i \\ \text{QRESTR} & \boxed{1} \\ \text{QSCOPE} & \boxed{4} \end{bmatrix}, \boxed{1}\begin{bmatrix} \text{RELN} & \textbf{dog} \\ \text{INST} & i \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{all} \\ \text{BV} & j \\ \text{QRESTR} & \boxed{3} \\ \text{QSCOPE} & \boxed{2} \end{bmatrix}, \right. \\ & \left. \boxed{3}\begin{bmatrix} \text{RELN} & \textbf{family} \\ \text{INST} & j \end{bmatrix}, \boxed{4}\begin{bmatrix} \text{RELN} & \textbf{save} \\ \text{SAVER} & i \\ \text{SAVED} & j \end{bmatrix} \right\rangle \end{bmatrix}$$

Notice that only the QSCOPE specifications have changed; the order of quantifiers on the RESTR list remains constant. That is because there is no semantic significance attached to the order of elements on the RESTR list. But (50) and (51) differ crucially in that the existential quantifier in (50) is not within the scope of any other quantifier, while in (51) it is the universal quantifier that has wide scope.

The differing constraints on QSCOPE values thus carry considerable semantic significance. Our grammar imposes constraints on the RESTR list of a multiply quantified sentence like (44) that can be satisfied in more than one way. Feature structures satisfying either (50) or (51) are allowed by the grammar. Moreover, if we make the further assumption that each index (variable) introduced by a quantificational NP (e.g. *every family*, *a dog*) must be BOUND, i.e. must occur within a feature structure that serves as the QSCOPE value of some quantificational predication with that index as its BV value, then these two are in fact the only possible RESTR lists that will satisfy the constraints of our grammar for a sentence like (44).

Though the feature structures satisfying our sentence descriptions must resolve the scope of quantifiers, note that the descriptions themselves need not. Our semantic representations thus enjoy an advantage that is not shared by standard predicate logic: if we don't specify any constraints on the QSCOPE values, we can essentially leave the quantifier scope unspecified. This kind of underspecification may have considerable appeal from a processing point of view: not only is it difficult for computational natural language applications to resolve the precise scope of quantifiers in even simple sentences, there is also psycholinguistic evidence that people don't always resolve scope.[17] Thus from the perspective of embedding our grammar within a model of human sentence processing or within a computational language processing system, it is significant that we can express generalized quantification in a way that allows unresolved, or even partially resolved, quantifier scope, depending on how many constraints are imposed on the values of QSCOPE.

Despite the interest and importance of these issues, we will leave quantification out of the picture in the semantic analyses we develop in the rest of the book. It will become apparent that we have our hands full with other aspects of meaning that interact in crucial ways with the syntactic phenomena that are our primary focus here.[18] We will therefore use simplified semantic representations for quantifiers as placeholders for the more complete analysis sketched. An example of how this would look for the determiner *a* is given in (52):

[17] See for example Kurtzman and MacDonald 1993.

[18] See the further reading section at the end of this chapter for references to recent work that integrates a view of quantification like the one just sketched with grammars of the sort we will motivate in subsequent chapters.

(52)
$$\left\langle \text{a}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{det} & \\ \text{AGR} & \textit{3sing} \\ \text{COUNT} & + \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{none} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{exist} \\ \text{BV} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Even with this simplified representation, there remains an interesting issue of compositional semantics: the value of the feature BV should end up being the same as the INDEX of the noun for which *a* is the specifier. However, this identity cannot be expressed as a constraint within the lexical entry for the determiner, since the determiner does not select for the noun (note that its COMPS and SPR lists are both empty). Instead, the determiner identifies its OWN index with the value of BV (i), and the lexical entry for a noun identifies its INDEX value with that of its SPR:

(53)
$$\left\langle \text{dog}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \textit{3sing} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \begin{bmatrix} \text{HEAD} & \textit{det} \\ \text{INDEX} & i \end{bmatrix} \right\rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{dog} \\ \text{INST} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

This means that the noun's INDEX value and the determiner's BV value end up being the same. Because *dog* identifies its own index (and the INST value of the **dog** predication) with the index of its specifier, and *a* identifies its index with the BV value of the **exist** predication, the lexical entries together with the grammar rules produce semantic representations like the one shown in (54) for the noun phrase *a dog*, with the value of BV correctly resolved:

(54)
$$\begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix}\text{RELN} & \textbf{exist}\\ \text{BV} & i\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{dog}\\ \text{INST} & i\end{bmatrix}\right\rangle \end{bmatrix}$$

Because the Semantic Inheritance Principle passes the head's INDEX value up to the phrasal level, this analysis generalizes naturally to syntactically complex specifiers, such as possessive NPs (see Problem 5 of Chapter 6).

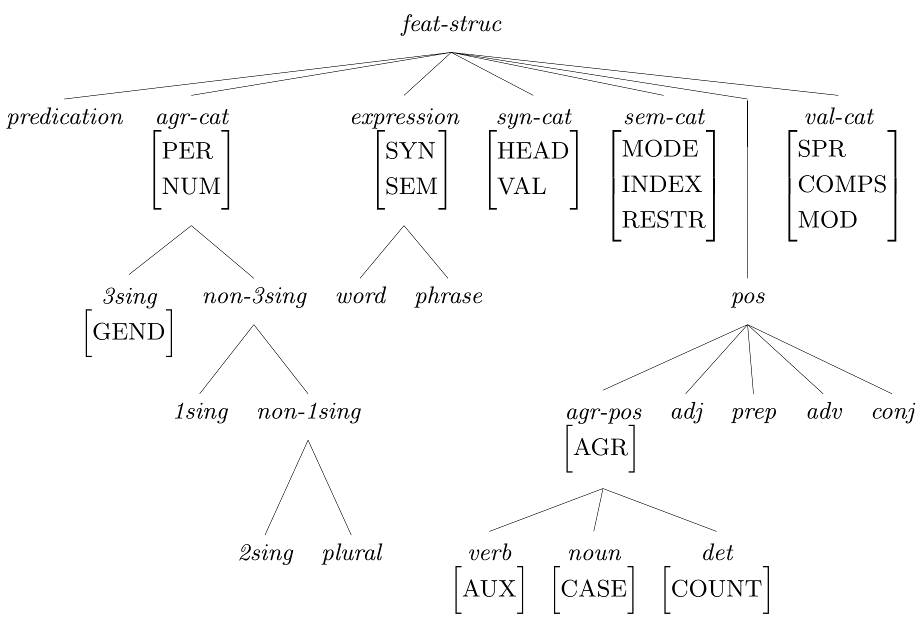
5.9 Summary

In this chapter, we introduced fundamental issues in the study of linguistic meaning and extended our grammar to include semantic descriptions. We then provided a systematic account of the relation between syntactic structure and semantic interpretation based on two constraints: the Semantic Compositionality Principle and the Semantic Inheritance Principle. These principles together provide a general account of how the semantics of a phrase is related to the semantics of its daughters. This chapter also extended the treatments of modification and coordinate structures to include an account of their linguistic meaning.

5.10 The Chapter 5 Grammar

5.10.1 The Type Hierarchy

The current version of our type hierarchy is summarized in (55):



5.10.2 Feature Declarations and Type Constraints

TYPE	FEATURES/CONSTRAINTS	IST
feat-struc		
expression	$\begin{bmatrix}\text{SYN} & \textit{syn-cat}\\ \text{SEM} & \textit{sem-cat}\end{bmatrix}$	*feat-struc*
syn-cat	$\begin{bmatrix}\text{HEAD} & \textit{pos}\\ \text{VAL} & \textit{val-cat}\end{bmatrix}$	*feat-struc*
sem-cat	$\begin{bmatrix}\text{MODE} & \{\text{prop, ques, dir, ref, none}\}\\ \text{INDEX} & \{i,\ j,\ k,\ \ldots,\ s_1,\ s_2,\ \ldots\}^{19}\\ \text{RESTR} & \textit{list}(\textit{predication})\end{bmatrix}$	*feat-struc*
predication	$\begin{bmatrix}\text{RELN} & \{\textbf{love}, \textbf{walk}, \ldots\}\\ \ldots & \end{bmatrix}$	*feat-struc*
word, *phrase*		*expression*
val-cat	$\begin{bmatrix}\text{SPR} & \textit{list}(\textit{expression})\\ \text{COMPS} & \textit{list}(\textit{expression})\\ \text{MOD} & \textit{list}(\textit{expression})\end{bmatrix}$	*feat-struc*
pos		*feat-struc*
agr-pos	$[\text{AGR}\quad \textit{agr-cat}]$	*pos*
verb	$[\text{AUX}\quad \{+,-\}]$	*agr-pos*
noun	$[\text{CASE}\quad \{\text{nom, acc}\}]$	*agr-pos*
det	$[\text{COUNT}\quad \{+,-\}]$	*agr-pos*
adj, *prep*, *adv*, *conj*		*pos*
agr-cat	$\begin{bmatrix}\text{PER} & \{\text{1st, 2nd, 3rd}\}\\ \text{NUM} & \{\text{sg, pl}\}\end{bmatrix}$	*feat-struc*
3sing	$\begin{bmatrix}\text{PER} & \text{3rd}\\ \text{NUM} & \text{sg}\\ \text{GEND} & \{\text{fem, masc, neut}\}\end{bmatrix}$	*agr-cat*
non-3sing		*agr-cat*
1sing	$\begin{bmatrix}\text{PER} & \text{1st}\\ \text{NUM} & \text{sg}\end{bmatrix}$	*non-3sing*
non-1sing		*non-3sing*
2sing	$\begin{bmatrix}\text{PER} & \text{2nd}\\ \text{NUM} & \text{sg}\end{bmatrix}$	*non-1sing*
plural	$[\text{NUM}\quad \text{pl}]$	*non-1sing*

[19] The possible values of the feature INDEX will be grouped together as the type *index* in the formal appendix to Chapter 6.

5.10.3 Abbreviations

(55)

$$\text{S} = \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \qquad \text{NP}_i = \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & i \end{bmatrix} \end{bmatrix}$$

$$\text{VP} = \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle \text{ X } \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \qquad \text{NOM} = \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle \text{ X } \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\text{V} = \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{verb} \end{bmatrix} \end{bmatrix} \qquad \text{N} = \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{noun} \end{bmatrix} \end{bmatrix}$$

$$\text{PP} = \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{prep} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \qquad \text{AP} = \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{adj} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\text{P} = \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{prep} \end{bmatrix} \end{bmatrix} \qquad \text{A} = \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{adj} \end{bmatrix} \end{bmatrix}$$

$$\text{DP}^{20} = \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{det} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

5.10.4 The Grammar Rules

In this summary, we give fully explicit versions of the grammar rules. In later chapters and the summary in Appendix A, we will abbreviate by supressing levels of embedding, e.g. by mentioning features such as SPR and COMPS without mentioning SYN or VAL.

(56) Head-Specifier Rule

$$\begin{bmatrix} \textit{phrase} & \\ \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \rightarrow \boxed{1} \quad \mathbf{H}\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

A phrase can consist of a (lexical or phrasal) head preceded by its specifier.

[20] We replace our old abbreviation D with a new abbreviation DP in anticipation of Problem 4 of Chapter 6, which introduces the possibility of determiner phrases. The abbreviation DP, like NP and VP, is underspecified and may represent either a *word* or a *phrase*.

(57) Head-Complement Rule

$$\begin{bmatrix} \textit{phrase} \\ \text{SYN} \begin{bmatrix}\text{VAL}\begin{bmatrix}\text{COMPS}\ \langle\ \rangle\end{bmatrix}\end{bmatrix}\end{bmatrix} \rightarrow \ \mathbf{H}\begin{bmatrix}\textit{word}\\ \text{SYN}\begin{bmatrix}\text{VAL}\begin{bmatrix}\text{COMPS}\ \langle \boxed{1},\ldots,\boxed{n}\rangle\end{bmatrix}\end{bmatrix}\end{bmatrix}\ \boxed{1}\ \ldots\ \boxed{n}$$

A phrase can consist of a lexical head followed by all its complements.

(58) Head-Modifier Rule

$$[\textit{phrase}] \ \rightarrow \ \mathbf{H}\boxed{1}\begin{bmatrix}\text{SYN}\begin{bmatrix}\text{VAL}\begin{bmatrix}\text{COMPS}\ \langle\ \rangle\end{bmatrix}\end{bmatrix}\end{bmatrix}\ \begin{bmatrix}\text{SYN}\begin{bmatrix}\text{VAL}\begin{bmatrix}\text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\boxed{1}\rangle\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

A phrase can consist of a (lexical or phrasal) head followed by a compatible modifier.

(59) Coordination Rule

$$\begin{bmatrix}\text{SYN [VAL } \boxed{0}]\\ \text{SEM [IND } s_0]\end{bmatrix} \rightarrow \begin{bmatrix}\text{SYN [VAL } \boxed{0}]\\ \text{SEM [IND } s_1]\end{bmatrix} \ldots \begin{bmatrix}\text{SYN [VAL } \boxed{0}]\\ \text{SEM [IND } s_{n-1}]\end{bmatrix}$$

$$\begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{HEAD } \textit{conj}\end{bmatrix}\\ \text{SEM} & \begin{bmatrix}\text{IND} & s_0\\ \text{RESTR} & \langle[\text{ARGS } \langle s_1,\ldots,s_n\rangle]\rangle\end{bmatrix}\end{bmatrix}\ \begin{bmatrix}\text{SYN [VAL } \boxed{0}]\\ \text{SEM [IND } s_n]\end{bmatrix}$$

Any number of elements with matching valence specifications can form a coordinate phrase with identical valence specifications.

5.10.5 The Principles

(60) Head Feature Principle (HFP)

In any headed phrase, the HEAD value of the mother and the HEAD value of the head daughter must be identical.

(61) Valence Principle

Unless the rule says otherwise, the mother's values for the VAL features (SPR, COMPS, and MOD) are identical to those of the head daughter.

(62) Specifier-Head Agreement Constraint (SHAC)

Verbs and common nouns must be specified as:

$$\begin{bmatrix}\text{SYN}\begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{AGR } \boxed{1}\end{bmatrix}\\ \text{VAL} & \begin{bmatrix}\text{SPR } \langle \begin{bmatrix}\text{AGR } \boxed{1}\end{bmatrix} \rangle\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

(63) Semantic Inheritance Principle

In any headed phrase, the mother's MODE and INDEX values are identical to those of the head daughter.

(64) Semantic Compositionality Principle

In any well-formed phrase structure, the mother's RESTR value is the sum of the RESTR values of the daughters.

5.10.6 Sample Lexical Entries

(65)

$$\left\langle \text{dog}\ , \left[\begin{array}{ll} \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll} \textit{noun} & \\ \text{AGR} & \textit{3sing}\end{array}\right] \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \langle\ \text{DP}_i\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle \end{array}\right]\end{array}\right] \\ \text{SEM} & \left[\begin{array}{ll} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \left[\begin{array}{ll} \text{RELN} & \textbf{dog} \\ \text{INST} & i \end{array}\right] \right\rangle \end{array}\right] \end{array}\right] \right\rangle$$

(66)

$$\left\langle \text{Kim}\ , \left[\begin{array}{ll} \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll} \textit{noun} & \\ \text{AGR} & \textit{3sing}\end{array}\right] \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle \end{array}\right]\end{array}\right] \\ \text{SEM} & \left[\begin{array}{ll} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \left[\begin{array}{ll} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{Kim} \\ \text{NAMED} & i \end{array}\right] \right\rangle \end{array}\right] \end{array}\right] \right\rangle$$

(67)

$$\left\langle \text{love}\ , \left[\begin{array}{ll} \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \textit{verb} \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \langle\ \text{NP}_i\ \rangle \\ \text{COMPS} & \langle\ \text{NP[acc]}_j\ \rangle \\ \text{MOD} & \langle\ \rangle \end{array}\right]\end{array}\right] \\ \text{SEM} & \left[\begin{array}{ll} \text{MODE} & \text{prop} \\ \text{INDEX} & s \\ \text{RESTR} & \left\langle \left[\begin{array}{ll} \text{RELN} & \textbf{love} \\ \text{SIT} & s \\ \text{LOVER} & i \\ \text{LOVED} & j \end{array}\right] \right\rangle \end{array}\right] \end{array}\right] \right\rangle$$

(68)

$$\left\langle \text{today}\ , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{adv} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \left\langle \begin{array}{c} \text{VP} \\ \begin{bmatrix} \text{INDEX} & s \end{bmatrix} \end{array} \right\rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{none} \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{today} \\ \text{ARG} & s \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(69)

$$\left\langle \text{and}\ , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{conj} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{MODE} & \text{none} \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{and} \\ \text{SIT} & s \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(70)

$$\left\langle \text{a}\ , \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{det} & \\ \text{AGR} & \textit{3sing} \\ \text{COUNT} & + \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{none} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{exist} \\ \text{BV} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

5.11 Further Reading

Much work on linguistic pragmatics builds directly on the pioneering work of the philosopher H. Paul Grice (see Grice 1989). A seminal work in modern research on natural language semantics is Frege's (1892) essay, 'Über Sinn und Bedeutung' (usually translated as 'On Sense and Reference'), which has been translated and reprinted in many anthologies (e.g. Geach and Black 1980). More recently, the papers of Richard Montague (Thomason, ed. 1974) had a revolutionary influence, but they are extremely technical. An elementary presentation of his theory is given by Dowty et al. (1981). General introductory texts in semantics include Chierchia and McConnell-Ginet 1990, Gamut 1991, and de Swart 1998.

All of these textbooks cover generalized quantifiers. For a more recent, more technical overview of generalized quantifiers, see Keenan and Westerståhl 1997. Shorter overviews of semantics include Bach 1989, Barwise and Etchemendy 1989 and Partee 1995. A short and very elementary introduction to generalized quantifiers is given in Larson 1995. The treatment of quantification sketched in Section 5.8 is developed more fully in Copestake et al. 1995, Copestake et al. 1999, and Copestake et al. 2001.

5.12 Problems

Problem 1: Two Kinds of Modifiers in English

In English, modifiers of nouns can appear either before or after the noun, although any given modifier is usually restricted to one position or the other.

(i) The red dog on the roof
(ii)*The on the roof dog
(iii)*The dog red

Our current Head-Modifier Rule only licenses post-head modifiers (like *on the roof* in (i)).

A. Write a second Head-Modifier Rule that licenses pre-head modifiers (e.g., *red* in (i)).
B. Modify the Head-Modifier 1 and Head-Modifier 2 Rules so that they are sensitive to which kind of modifier is present and don't generate (ii) or (iii). [*Hint: Use a feature* [*POST-HEAD* $\{+,-\}$] *to distinguish* red *and* on the roof*.*]
C. Is POST-HEAD a HEAD feature? Why or why not?
D. Give lexical entries for *red* and *on* that show the value of POST-HEAD. (You may omit the SEM features in these entries.)
E. Is (i) ambiguous according to your grammar (i.e. the Chapter 5 grammar modified to include the two Head-Modifier Rules, instead of just one)? Explain your answer.

This problem assumed that we don't want to make the two Head-Modifier Rules sensitive to the part of speech of the modifier. One reason for this is that modifiers of the same part of speech can occur before and after the head, even though individual modifiers might be restricted to one position or the other.

F. Provide three examples of English NPs with adjectives or APs after the noun.
G. Provide three examples of adverbs that can come before the verbs they modify.
H. Provide three examples of adverbs that can come after the verbs they modify.

Problem 2: Modes of Coordination

Consider the following data:

(i) Kim left and Sandy left.
(ii) ?*Kim left and did Sandy leave.
(iii) ?*Did Sandy leave and Kim left.
(iv) Did Sandy leave and did Kim leave?
(v) Go away and leave me alone!
(vi) ?*Kim left and leave me alone!
(vii) ?*Leave me alone and Kim left.
(viii) ?*Leave me alone and did Kim leave?
(ix) ?*Did Kim leave and leave me alone!

A. Formulate a generalization about the MODE value of conjuncts (and their mother) that could account for these data.

B. Modify the Coordination Rule in (42) so that it enforces the generalization you formulated in (A).

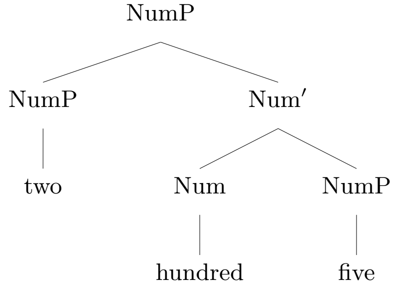
Problem 3: Semantics of Number Names

In Problem 5 of Chapter 3, we considered the syntax of English number names, and in particular how to find the head of a number name expression. Based on the results of that problem, the lexical entry for *hundred* in a number name like *two hundred five* should include the constraints in (i): (Here we are assuming a new subtype of *pos*, *number*, which is appropriate for number name words.)

(i)
$$\left\langle \text{hundred}\ ,\ \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{number} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \begin{bmatrix}\text{HEAD} & \textit{number}\end{bmatrix} \right\rangle \\ \text{COMPS} & \left\langle \begin{bmatrix}\text{HEAD} & \textit{number}\end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle$$

This lexical entry interacts with our ordinary Head-Complement and Head-Specifier Rules to give us the phrase structure shown in (ii):

(ii)



Smith (1999) provides a compositional semantics of number names. The semantics of the top node in this small tree should be (iii):

$$
\text{(iii)}\ \begin{bmatrix}
\text{INDEX} & i \\
\text{MODE} & \text{ref} \\
\text{RESTR} & \left\langle \begin{bmatrix}\text{RELN} & \textbf{constant}\\ \text{INST} & l \\ \text{VALUE} & 2\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{times}\\ \text{RESULT} & k \\ \text{FACTOR1} & l \\ \text{FACTOR2} & m\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{constant}\\ \text{INST} & m \\ \text{VALUE} & 100\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{plus}\\ \text{RESULT} & i \\ \text{TERM1} & j \\ \text{TERM2} & k\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{constant}\\ \text{INST} & j \\ \text{VALUE} & 5\end{bmatrix} \right\rangle
\end{bmatrix}
$$

This may seem long-winded, but it is really just a way of expressing "(two times one hundred) plus five" (i.e. 205) in our feature structure notation.

A. Assume that the two constant predications with the values 2 and 5 are contributed by the lexical entries for *two* and *five*. What predications must be on the RESTR list of the lexical entry for *hundred* in order to build (iii) as the SEM value of *two hundred five*?

B. The lexical entry for *hundred* will identify the indices of its specifier and complement with the value of some feature of a predication on its RESTR list. Which feature of which predication is the index of the specifier identified with? What about the index of the complement?

C. The lexical entry for *hundred* will identify its own INDEX with the value of some feature of some predication on its RESTR list. Which feature of which predication must this be, in order for the grammar to build (iii) as the SEM value of *two hundred five*?

D. Based on your answers in parts (A)–(C), give a lexical entry for *hundred* that includes the constraints in (i) and a fully specified SEM value. [*Note: Your lexical entry need only account for* hundred *as it is used in* two hundred five. *Don't worry about other valence possibilities, such as* two hundred, two hundred and five, *or* a hundred.]

E. The syntax and semantics of number names do not line up neatly: In the syntax, *hundred* forms a constituent with *five*, and *two* combines with *hundred five* to give a larger constituent. In the semantics, the constant predications with the values 2 and 100 are related via the times predication. The result of that is related to the constant predication with the value 5, via the plus predication Why is this mismatch not a problem for the grammar?

6

How the Grammar Works

6.1 A Factorization of Grammatical Information

Three chapters ago, we began modifying the formalism of context-free grammar to better adapt it to the sorts of generalizations we find in natural languages. We broke grammatical categories down into features, and then we broke the values of features down into features, as well. In the process, we moved more and more syntactic information out of the grammar rules and into the lexicon. In effect, we changed our theory of grammar so that the rules give only very general patterns that cut across grammatical categories. Details about which expressions can go with which are specified in lexical entries in terms of valence features.

With the expanded ability of our new feature structure complexes to express cross-categorial generalizations, our four remaining grammar rules cover a wide range of cases. Two of them – the rules introducing complements and specifiers – were discussed extensively in Chapter 4. The third one – a generalization of our old rules introducing PP modifiers to VP and NOM – was illustrated in the previous chapter.[1] The fourth is the Coordination Rule. The formal statements of these rules were given at the end of the previous chapter, along with informal translations (given in italics below the rules).

In addition to our grammar rules, we must provide (as we did in the case of CFGs) some characterization of the 'initial symbol', corresponding to the type of phrases that can stand alone as sentences of the language. We postpone a careful characterization of this until Chapter 8, when we will have introduced a method for distinguishing finite (that is, tensed) clauses from others. For now, we can treat S (which we characterized in terms of features in Chapter 4) as the initial symbol.

We were able to make our grammar rules so general in part because we formulated four general principles about how information must be distributed in well-formed trees: the Head Feature Principle, the Valence Principle, the Semantic Compositionality Principle, and the Semantic Inheritance Principle. These were also reiterated at the end of Chapter 5.

The richer feature structures we are now using, together with our highly schematized rules, have required us to refine our notion of how a grammar is related to the fully

[1]It should be noted that the Head-Modifier Rule does not cover all kinds of modifiers. In particular, some modifiers – such as adjectives inside NPs – precede the heads that they modify. To accommodate such modifiers, we would need an additional grammar rule. This issue was addressed in Problem 1 of Chapter 5.

determinate phrase structure trees of the language. Intuitively, here is how it works:

First, each lexical entry licenses a family of word structures – each of which is a nonbranching tree. More precisely, a lexical entry $\langle \omega, \Phi \rangle$ licenses any word structure of the form:

$$\begin{array}{c} \text{F} \\ | \\ \omega \end{array}$$

if and only if F is a resolved feature structure that satisfies Φ. A resolved feature structure F satisfies Φ if and only if it assigns values to all features appropriate for feature structures of its type, and those values are consistent with all of the information specified in Φ.

Such lexical trees form the bottom layer of well-formed phrasal trees. They can be combined[2] into larger trees in the ways permitted by the grammar rules, obeying the constraints imposed by our four principles. This process can apply to its own output, making ever larger phrasal trees. So long as the local tree at the top of each tree structure that we construct is licensed by a grammar rule and conforms to these principles, it is well formed. Typically, each node in a well-formed tree will contain some information that was stipulated by a rule and other information that percolated up (metaphorically speaking) from lower nodes (and ultimately from the lexical entries) via the principles. In summary, the relation between our trees and the grammatical mechanisms that license them is as follows: a tree is well-formed if, and only if, it satisfies all of the conditions imposed by the lexical entries of the words it contains, by the grammar rules, and by the general grammatical principles.

We have formulated our theory so that the number of tree structures consistent with a given terminal string will shrink considerably as constraints from higher levels of structure are brought into the picture. This important effect of contextual constraints can be illustrated with the CASE value of proper nouns. Consider the lexical entry in (1):

(1)
$$\left\langle \text{Leslie}\ , \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \textit{3sing} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{Leslie} \\ \text{NAMED} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

[2]Our informal discussion is worded in terms of a process of building trees up from the bottom. This is a conceptually natural way of thinking about it, but it should not be taken too literally. The formal definition of well-formed tree structure that we give below is deliberately nonprocedural.

This lexical entry gives fully specified values for every feature except CASE and GEND. (It may look underspecified for PER and NUM as well, but recall that the type *3sing* is constrained to have specific values for each of those features.) Since the features CASE and GEND are left underspecified in the lexical entry, the lexical entry licenses six distinct word structures. We have shown two in (2):

(2) a.

$$\begin{array}{c}\left[\begin{array}{ll}\textit{word} & \\ \text{SYN} & \left[\begin{array}{ll}\text{HEAD} & \left[\begin{array}{ll}\textit{noun} & \\ \text{CASE} & \text{nom} \\ \text{AGR} & \left[\begin{array}{ll}\textit{3sing} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{sg} \\ \text{GEND} & \text{fem}\end{array}\right]\end{array}\right] \\ \text{VAL} & \left[\begin{array}{ll}\text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle\end{array}\right]\end{array}\right] \\ \text{SEM} & \left[\begin{array}{ll}\text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle\left[\begin{array}{ll}\text{RELN} & \textbf{name} \\ \text{NAME} & \text{Leslie} \\ \text{NAMED} & i\end{array}\right]\right\rangle\end{array}\right]\end{array}\right] \\ | \\ \text{Leslie}\end{array}$$

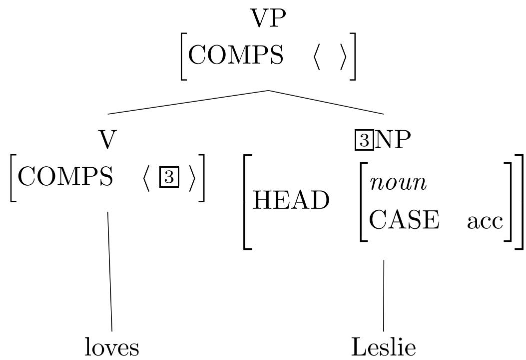
b.

$$\begin{array}{c}\left[\begin{array}{ll}\textit{word} & \\ \text{SYN} & \left[\begin{array}{ll}\text{HEAD} & \left[\begin{array}{ll}\textit{noun} & \\ \text{CASE} & \text{acc} \\ \text{AGR} & \left[\begin{array}{ll}\textit{3sing} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{sg} \\ \text{GEND} & \text{masc}\end{array}\right]\end{array}\right] \\ \text{VAL} & \left[\begin{array}{ll}\text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle\end{array}\right]\end{array}\right] \\ \text{SEM} & \left[\begin{array}{ll}\text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle\left[\begin{array}{ll}\text{RELN} & \textbf{name} \\ \text{NAME} & \text{Leslie} \\ \text{NAMED} & i\end{array}\right]\right\rangle\end{array}\right]\end{array}\right] \\ | \\ \text{Leslie}\end{array}$$

Notice that we could have abbreviated the mother of these tree structures either as ‘N’ or as ‘NP’, since this is a node of type *word* whose HEAD value is of type *noun* with empty SPR and COMPS lists.

Although these two word structures both satisfy the constraints given in the lexical entry equally well, only the tree in (2b) can be embedded within a larger one like (3), licensed by the Head-Complement Rule:

(3)



That is because we have assumed here (following the results of Chapter 4, Problem 6) that the lexical entry for *loves* specifies that its complement is [CASE acc]. Because the Head-Complement Rule identifies the head daughter’s COMPS list with the list of (the feature structures of the) complement daughters, the accusative case specification must be part of the object noun’s HEAD value.[3]

The information specified by our rules and lexical entries is thus partial information. Each rule says, in effect, that subtrees of a certain kind are sanctioned, but the rule only specifies some of the constraints that the trees that it licenses must obey. Likewise, a lexical entry says that certain trees dominating the phonological form in that entry are sanctioned, but the entry only specifies some of the information relevant at higher levels of structure. The general principles of our theory constrain the ways in which feature values can be distributed in well-formed phrase structure trees. The job of determining well-formedness can be distributed among the various pieces of our grammatical system because the licensing mechanism requires simultaneous satisfaction of all of the relevant constraints.

In developing our grammar so far, we have arrived at a particular factorization of the information necessary for a precise account of grammatical structure. By far the richest source of information in this factorization is the lexicon. That is, our grammar embodies the claim that both the problem of determining which strings of words constitute well-formed sentences and the problem of specifying the linguistic meaning of sentences depend mostly on the nature of words. Of course, it must also be recognized that there are many regularities about which words go together (and how they go together). The theoretical constructs summarized here capture a number of such regularities; subsequent chapters will provide ways of capturing more.

[3]Nothing in the syntactic context constrains the GEND value, however. The appropriate value there will depend on the non-linguistic context, in particular, on the gender of the person the speaker intends to refer to.

6.2 Examples

6.2.1 A Detailed Example

The best way to understand how the various components of our grammatical theory interact is to work through detailed analyses of linguistic examples. In this subsection, we show in detail how the grammar of English, as we have developed it to this point, handles one simple sentence of English, namely:[4]

(4) They sent us a letter.

We begin our lexical analysis with the entry for the word *letter*:

(5)
$$
\left\langle \text{letter} , \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \begin{bmatrix} \textit{3sing} & \\ \text{GEND} & \text{neut} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \overset{\text{D}}{\begin{bmatrix} \text{COUNT} & + \\ \text{INDEX} & k \end{bmatrix}} \right\rangle \\ \text{COMPS} & \langle\ (\text{PP}_m)\ \rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & k \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{letter} \\ \text{INST} & k \\ \text{ADDRESSEE} & m \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle
$$

We assume *letter* optionally selects a PP complement, as indicated.

How many word structures satisfy (5)? The answer to this question may be surprising. There are *infinitely many* word structures that satisfy (5). Moreover, this will be true whenever a lexical entry selects something on its COMPS or SPR list, because lexical entries specify such minimal information about the things they select for. For example, in the absence of further constraints, the member of the SPR list in a word structure licensed by (5) could have a RESTR list of any length. Similarly, if the COMPS list in the word structure contains a PP, that PP could have a RESTR value of any length. And this is as it should be, as there is no upper bound on the length of PP complements of this word:

[4]In this section, we present the details of trees over the course of several pages, depicting various subtrees and how they fit together to make larger trees. In doing this, we use tags to mark identity across distinct diagrams of trees that will eventually be put together into a single tree. We also reuse tags across different trees when the same lexical entry is used in different sentences. Strictly speaking, tags only mark identity within a given description. We are taking this liberty with the tag notation only in this section, because it is a convenient heuristic.

(6) a. the letter to Kim...
b. the letter to Kim and Sandy...
c. the letter to Kim, Lee and Sandy...
d. the letter to the person who signed the document that started the mishap that...

That is, depending on the surrounding context (i.e. depending on which words the PP actually contains), the PP's RESTR list might have one, three, thirty-seven, or two hundred predications on it. The same is true of the specifier, as the examples in (7) indicate:

(7) a. the letter...
b. almost every letter...
c. Sandy's friend's mother's letter...
d. the cricket club's former secretary's letter...

If we assume the analysis of quantificational determiners sketched at the end of Chapter 5, then the word structure for *letter* that is relevant to the sentence in (4), however, has a SPR value whose RESTR is singleton:

(8)
$$
\begin{array}{c}
\begin{bmatrix}
\textit{word} & \\
\text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \boxed{1}\begin{bmatrix}\textit{3sing} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{sg} \\ \text{GEND} & \text{neut}\end{bmatrix} \\ \text{CASE} & \text{acc}\end{bmatrix} \\
\text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \begin{bmatrix} \text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\textit{det} & \\ \text{AGR} & \boxed{1} \\ \text{COUNT} & + \end{bmatrix} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle\end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{MODE} & \text{none} \\ \text{INDEX} & k \\ \text{RESTR} & \left\langle \begin{bmatrix}\text{RELN} & \textbf{exist} \\ \text{BV} & k\end{bmatrix}\right\rangle\end{bmatrix}\end{bmatrix}\right\rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix}\end{bmatrix} \\
\text{SEM} & \begin{bmatrix}\text{MODE} & \text{ref} \\ \text{INDEX} & k \\ \text{RESTR} & \left\langle\begin{bmatrix}\text{RELN} & \textbf{letter} \\ \text{INST} & k \\ \text{ADDRESSEE} & m\end{bmatrix}\right\rangle\end{bmatrix}
\end{bmatrix} \\
\Big| \\
\text{letter}
\end{array}
$$

As for the COMPS value, the empty list option has been exercised in this tree, as the sentence whose structure we are building contains no PP complement. Notice that, with no PP, there is no constituent that will realize the ADDRESSEE role. Since we have not imposed any constraint requiring that semantic roles be realized syntactically, this does not present any technical problem. And having an ADDRESSEE role for the noun *letter*, even when no addressee is mentioned, seems quite intuitive. Finally, note that (8) obeys the Specifier-Head Agreement Constraint, which identifies the AGR value of the noun with that of the element on its SPR list.

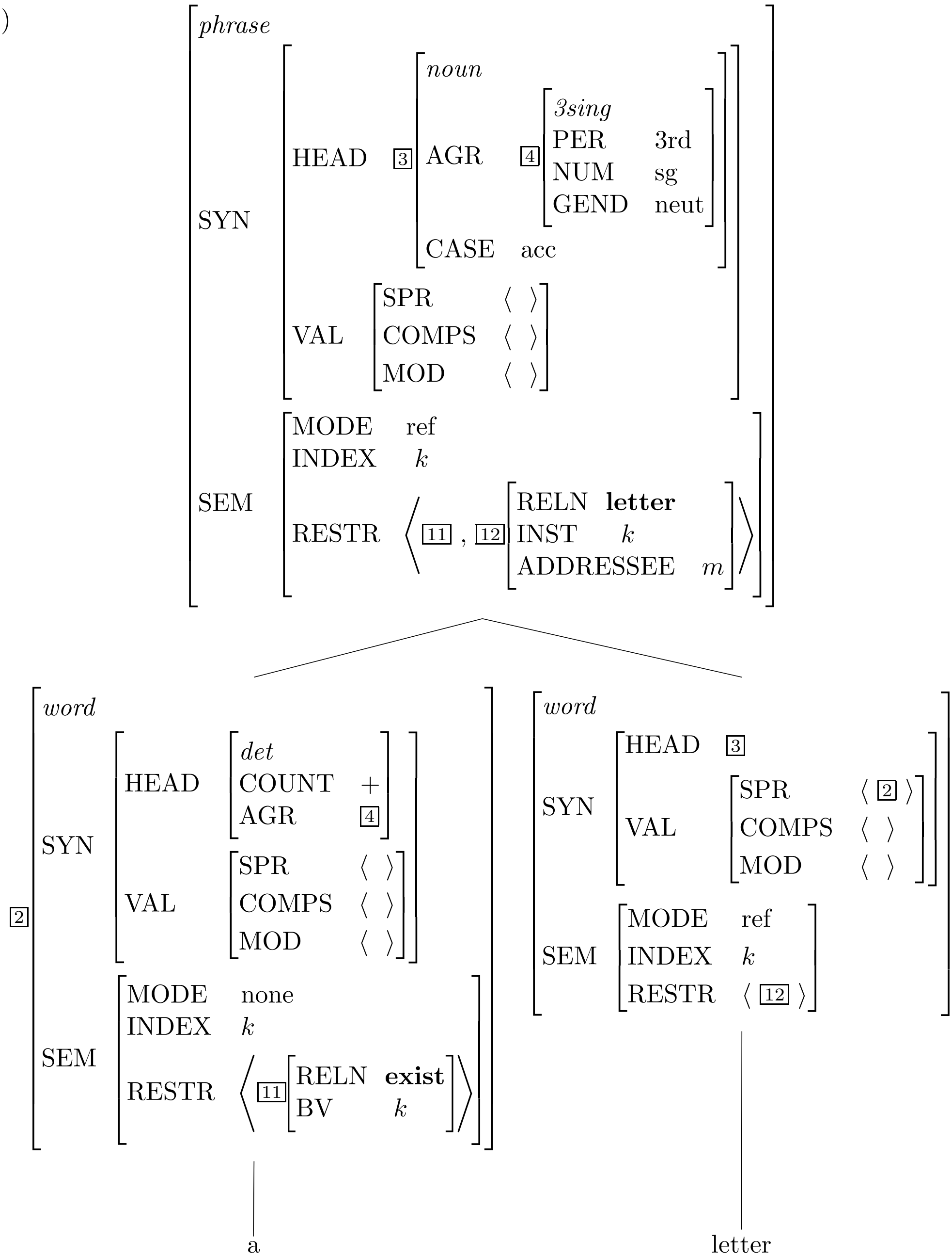
The word structure for the word *a* is abbreviated in (9):[5]

(9)
$$
\begin{array}{c}
\begin{bmatrix}
\textit{word} & \\
\text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{det} & \\ \text{AGR} & \begin{bmatrix} \textit{3sing} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{sg} \\ \text{GEND} & \text{neut} \end{bmatrix} \\ \text{COUNT} & + \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\
\text{SEM} & \begin{bmatrix} \text{MODE} & \text{none} \\ \text{INDEX} & k \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{exist} \\ \text{BV} & k \end{bmatrix} \right\rangle \end{bmatrix}
\end{bmatrix} \\
\Big| \\
\text{a}
\end{array}
$$

[5] What is not shown in this tree is the complete feature specification for the **exist** predication. See Section 5.8 of Chapter 5 for discussion.

The following tree results from combining (8) and (9) via the Head-Specifier Rule:

(10)



In this tree, the left subtree is exactly the one shown in (9). The identification of the element on the head daughter's SPR list ($\boxed{2}$) and the feature structure of the left daughter is guaranteed by the Head-Specifier Rule, which licenses the combination of this determiner with this noun. When the Head-Specifier Rule enforces this identity, it forms a link in a chain of identities: the lexical entry for *letter* identifies the INDEX of the element on its SPR list with its own INDEX and INST values. The lexical entry for *a* identifies its INDEX with its BV value. When these two words combine via the Head-Specifier Rule, the INDEX of the specifier of *letter* and the INDEX of *a* are identified. This chain of identities ensures that the BV of the **exist** predication and the INST of the **letter** predication are one and the same (k).

(10) obeys the HFP: the HEAD value of the head daughter is identified with that of the mother ($\boxed{3}$). And it obeys the Valence Principle: the COMPS value of the phrase is the same as that of the head daughter (the empty list). The mother's SPR value is the empty list, as required by the Head-Specifier Rule.

The Semantic Inheritance Principle says that the MODE and INDEX values of the head daughter must be shared by the mother, which is the case in (10). And the Semantic Compositionality Principle requires that the mother's RESTR value be the sum of the two daughters' RESTR lists. This concludes the analysis of the noun phrase *a letter*, as it appears in the sentence in (4).

The lexical entry for the pronoun *us* is quite straightforward, except for the RESTR list in the semantics. In the following, we have chosen to characterize the meaning of *us* roughly as reference to a group of which the speaker is a member. We have formalized this as a RESTR list with three elements, but there are many other possible ways of doing this. Our version gives rise to the following lexical tree:

(11)

$$
\begin{array}{c}
\left[\begin{array}{ll}
\textit{word} & \\
\text{SYN} & \left[\begin{array}{ll}
\text{HEAD} & \left[\begin{array}{ll}\textit{noun} & \\ \text{CASE} & \text{acc} \\ \text{AGR} & \left[\begin{array}{ll}\textit{plural} & \\ \text{PER} & \text{1st}\\ \text{NUM} & \text{pl}\end{array}\right]\end{array}\right] \\
\text{VAL} & \left[\begin{array}{ll}\text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle\end{array}\right]
\end{array}\right] \\
\text{SEM} & \left[\begin{array}{ll}
\text{MODE} & \text{ref}\\
\text{INDEX} & j \\
\text{RESTR} & \left\langle \left[\begin{array}{ll}\text{RELN} & \textbf{group}\\ \text{INST} & j\end{array}\right], \left[\begin{array}{ll}\text{RELN} & \textbf{speaker}\\ \text{INST} & l\end{array}\right], \left[\begin{array}{ll}\text{RELN} & \textbf{member}\\ \text{SET} & j \\ \text{ELEMENT} & l\end{array}\right]\right\rangle
\end{array}\right]
\end{array}\right] \\
| \\
\text{us}
\end{array}
$$

All this information is lexically specified. Note that because the AGR value is of type *plural*, it contains no GEND specification.

Now consider the lexical entry for the word *sent*:[6]

(12)

$$\left\langle \text{sent}\ , \left[\begin{array}{ll} \textit{word} & \\ \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \textit{verb} \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \left\langle \begin{array}{c}\text{NP}_i \\ \left[\text{CASE} \quad \text{nom}\right]\end{array} \right\rangle \\ \text{COMPS} & \left\langle \begin{array}{c}\text{NP}_j \\ \left[\text{CASE} \quad \text{acc}\right]\end{array} , \begin{array}{c}\text{NP}_k \\ \left[\text{CASE} \quad \text{acc}\right]\end{array} \right\rangle \\ \text{MOD} & \langle\ \rangle \end{array}\right] \end{array}\right] \\ \text{SEM} & \left[\begin{array}{ll} \text{MODE} & \text{prop} \\ \text{INDEX} & s_7 \\ \text{RESTR} & \left\langle \left[\begin{array}{ll} \text{RELN} & \textbf{send} \\ \text{SIT} & s_7 \\ \text{SENDER} & i \\ \text{SENDEE} & j \\ \text{SENT} & k \end{array}\right] \right\rangle \end{array}\right] \end{array}\right] \right\rangle$$

Note that, as a past tense form, this lexical entry has an underspecified AGR value. All of the word structures licensed by (12), however, have fully resolved AGR values, and by the SHAC, must share those AGR values with their specifiers. Similarly, although the lexical entry in (12) places no restrictions on the AGR value of the complements, those AGR values are fully specified in the word structures. The word structure for *sent* that is relevant to the sentence in (4) is shown in (13):[7]

[6]We are ignoring the semantic contribution of the past tense in this discussion.

[7]Although the tree in (13) represents a fully resolved word structure, we have abbreviated somewhat. In particular, we have not shown the SEM values within the elements of the SPR and COMPS lists. Similar remarks apply to many of the trees in the remainder of this chapter.

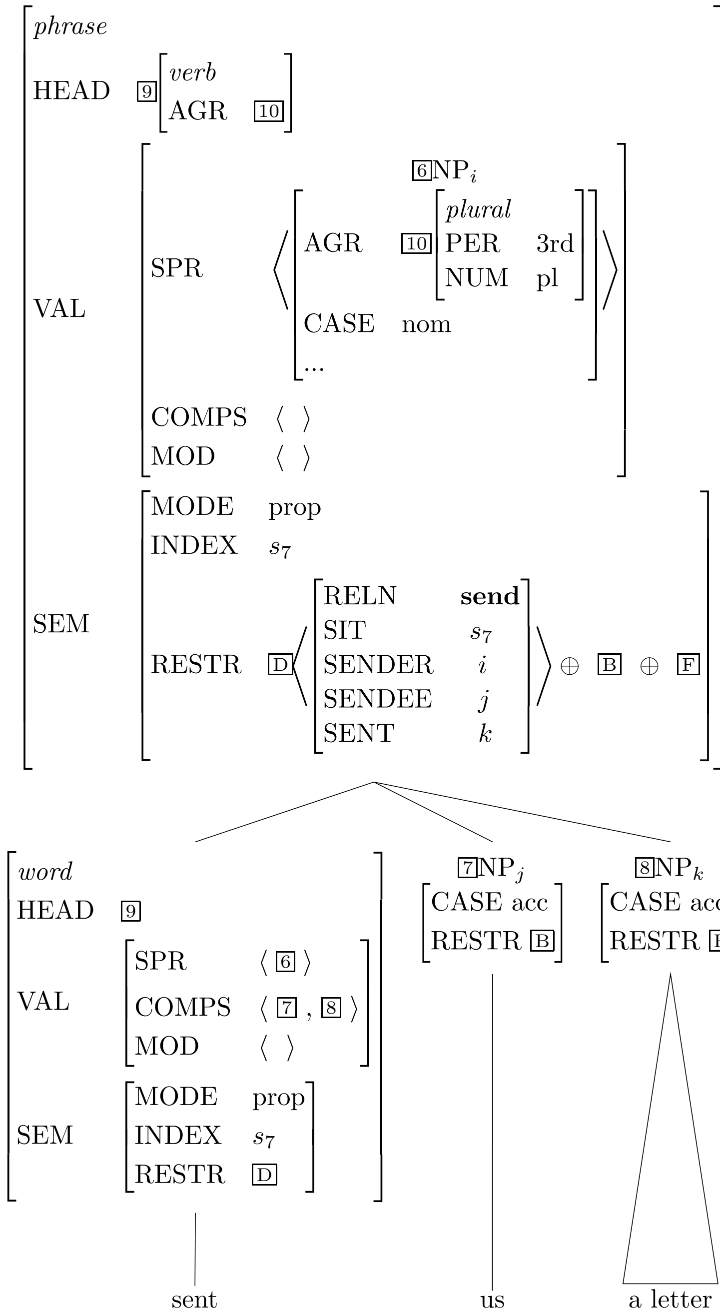
(13)

$$\begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \boxed{10} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \overset{\text{NP}_i}{\begin{bmatrix} \text{CASE} & \text{nom} \\ \text{AGR} & \boxed{10}\begin{bmatrix} \textit{plural} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{pl} \end{bmatrix} \\ \ldots & \end{bmatrix}} \right\rangle \\ \text{COMPS} & \left\langle \overset{\text{NP}_j}{\begin{bmatrix} \text{CASE} & \text{acc} \\ \text{AGR} & \begin{bmatrix} \textit{plural} & \\ \text{PER} & \text{1st} \\ \text{NUM} & \text{pl} \end{bmatrix} \\ \ldots & \end{bmatrix}} , \overset{\text{NP}_k}{\begin{bmatrix} \text{CASE} & \text{acc} \\ \text{AGR} & \begin{bmatrix} \textit{3sing} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{sg} \\ \text{GEND} & \text{neut} \end{bmatrix} \\ \ldots & \end{bmatrix}} \right\rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s_7 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{send} \\ \text{SIT} & s_7 \\ \text{SENDER} & i \\ \text{SENDEE} & j \\ \text{SENT} & k \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

sent

The three trees we have now built up combine via the Head-Complement Rule to give the following tree structure:

(14)



We have done a bit more abbreviating here. The node tagged $\boxed{7}$ is identical to the top node of the word structure in (11). Likewise, the node tagged $\boxed{8}$ is identical to the top node in (10).

The [CASE acc] constraints on both these NPs comes from the COMPS value of the lexical entry for *sent* (see (12)), and hence appears on this node, as required by the Head-Complement Rule. The RESTR values in the semantics for the two NP nodes are the ones shown in (11) and (10). We abbreviated these with the tags $\boxed{\text{B}}$ and $\boxed{\text{F}}$, respectively.

(14) obeys the conditions on COMPS values specified in the Head-Complement Rule, that is, the head daughter's complements are identified with the non-head daughters and the mother's COMPS value is empty. (14) obeys the Valence Principle, as the SPR value of the head daughter, not mentioned in the rule, is preserved as the mother's SPR value. Likewise, the HEAD value of mother and head daughter are correctly identified here, in accordance with the Head Feature Principle. Finally, the MODE and INDEX values of the mother are those of the head daughter, while the RESTR value of the mother is the sum of those of all the daughters, as specified by the semantic principles.

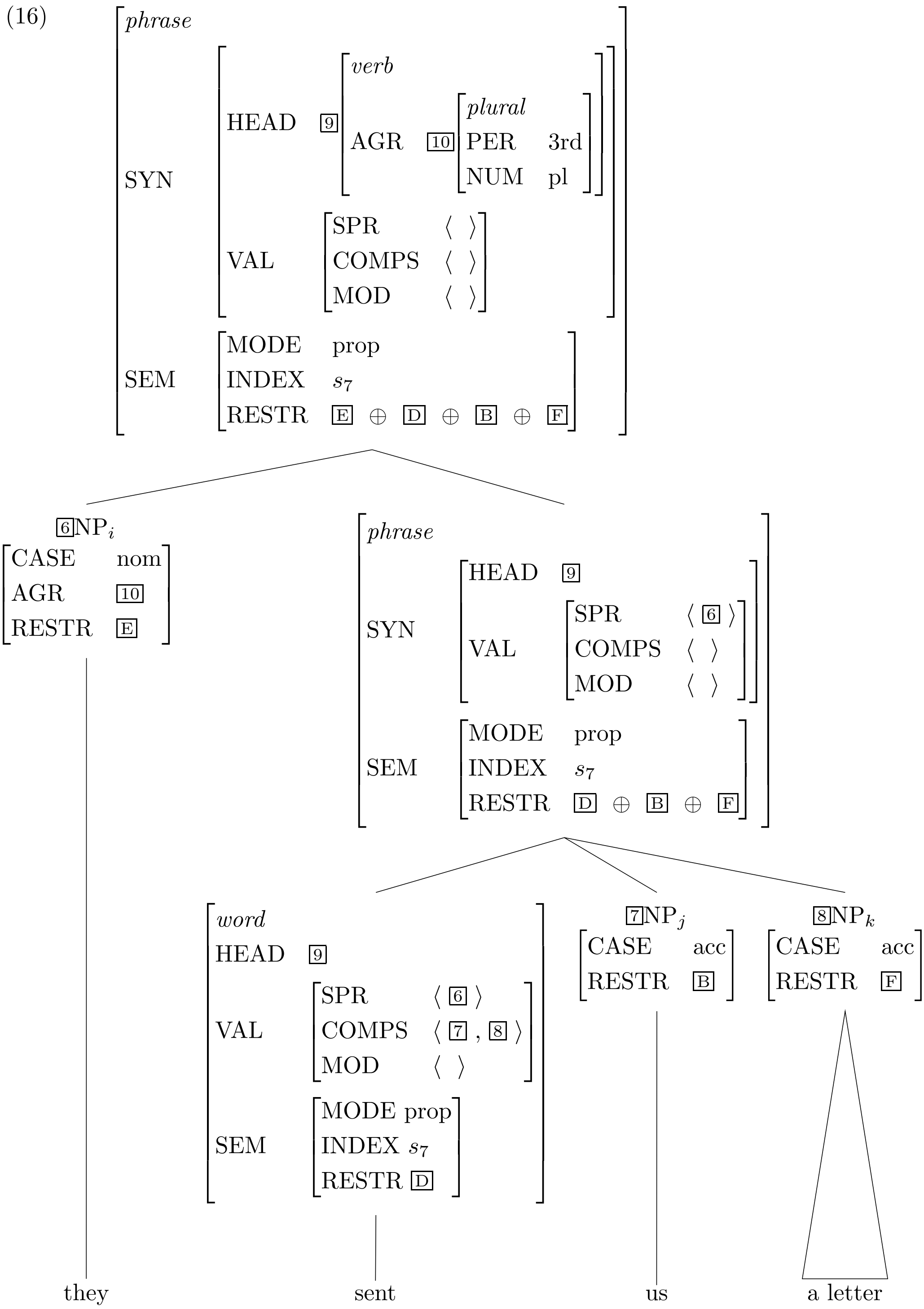
The last step is to combine the VP in (14) with the tree structure for its subject NP. The following is the word structure for the pronoun *they*, as licensed by an appropriate lexical entry:

(15)

$$
\begin{array}{c}
\left[\begin{array}{ll}
\textit{word} & \\
\text{SYN} & \left[\begin{array}{ll}
\text{HEAD} & \left[\begin{array}{ll}
\textit{noun} & \\
\text{CASE} & \text{nom} \\
\text{AGR} & \left[\begin{array}{ll}
\textit{plural} & \\
\text{PER} & \text{3rd} \\
\text{NUM} & \text{pl}
\end{array}\right]
\end{array}\right] \\
\text{VAL} & \left[\begin{array}{ll}
\text{SPR} & \langle\ \rangle \\
\text{COMPS} & \langle\ \rangle \\
\text{MOD} & \langle\ \rangle
\end{array}\right]
\end{array}\right] \\
\text{SEM} & \left[\begin{array}{ll}
\text{MODE} & \text{ref} \\
\text{INDEX} & i \\
\text{RESTR} & \left\langle \left[\begin{array}{ll}
\text{RELN} & \textbf{group} \\
\text{INST} & i
\end{array}\right] \right\rangle
\end{array}\right]
\end{array}\right] \\
| \\
\text{they}
\end{array}
$$

The result is the tree in (16):

(16)


phrase
SYN
HEAD 9
verb
AGR 10
plural
PER 3rd
NUM pl
VAL
SPR ⟨ ⟩
COMPS ⟨ ⟩
MOD ⟨ ⟩
SEM
MODE prop
INDEX s7
RESTR E ⊕ D ⊕ B ⊕ F
6 NPi
CASE nom
AGR 10
RESTR E
phrase
SYN
HEAD 9
VAL
SPR ⟨ 6 ⟩
COMPS ⟨ ⟩
MOD ⟨ ⟩
SEM
MODE prop
INDEX s7
RESTR D ⊕ B ⊕ F
word
HEAD 9
VAL
SPR ⟨ 6 ⟩
COMPS ⟨ 7 , 8 ⟩
MOD ⟨ ⟩
SEM
MODE prop
INDEX s7
RESTR D
7 NPj
CASE acc
RESTR B
8 NPk
CASE acc
RESTR F
they
sent
us
a letter

Again, we have abbreviated. The node labeled $\boxed{6}$ is just the top node in (15). The nodes labeled $\boxed{7}$ and $\boxed{8}$ are exactly as they were in (14), as is the VP node. We have abbreviated the RESTR values, simply putting in tags or sums of tags. The RESTR value of the top node, fully spelled out (except for the somewhat abbreviated contribution of the word *a*), is the list consisting of the following seven predications (in the indicated order):

(17) $$\begin{bmatrix}\text{RELN} & \textbf{group}\\ \text{INST} & i\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{send}\\ \text{SIT} & s_7\\ \text{SENDER} & i\\ \text{SENDEE} & j\\ \text{SENT} & k\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{group}\\ \text{INST} & j\end{bmatrix},$$

$$\begin{bmatrix}\text{RELN} & \textbf{speaker}\\ \text{INST} & l\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{member}\\ \text{SET} & j\\ \text{ELEMENT} & l\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{exist}\\ \text{BV} & k\end{bmatrix},$$

$$\begin{bmatrix}\text{RELN} & \textbf{letter}\\ \text{INST} & k\\ \text{ADDRESSEE} & m\end{bmatrix}$$

The AGR value in the top node of (16) is identical to that in the subject NP, as required by the interaction of the HFP, the Head-Specifier Rule, and the SHAC. In general, this tree structure obeys the Head Feature Principle, the Valence Principle, and the two semantic principles.

This concludes our analysis of the sentence *They sent us a letter*. The various constraints in our grammar interact to ensure that this structure and infinitely many related to it are well-formed, while guaranteeing that infinitely many other structures similar to it are ill-formed.

Exercise 1: The Non-infinity of *Us*
The lexical entry for *letter* licenses infinitely many word structures, while the lexical entry for *us* licenses exactly one. What feature specifications in the lexical entries are behind this difference?

6.2.2 Another Example

The detailed analysis we just went through built the sentence from the bottom up. This is one way to use the grammatical machinery we have developed, but it is not the only way. We could equally well have started with at the top of the tree, showing how our rules, principles, and lexical entries interact to license all its parts.

To see this top-down approach in action, consider the following sentence:[8]

(18) We send two letters to Lee.

[8]This example sounds a bit odd in isolation, but it would be perfectly natural in the appropriate context, for example, in response to the question, *What do we do if Alex writes to us?*

Example (18) is structurally ambiguous in a way analogous to the familiar example, *I saw the astronomer with a telescope.* That is, the PP *to Lee* can be attached either to the VP or to the NP headed by *letters*. In our semantic representation, the two readings correspond to two different RESTR lists, shown in (19) and (20):

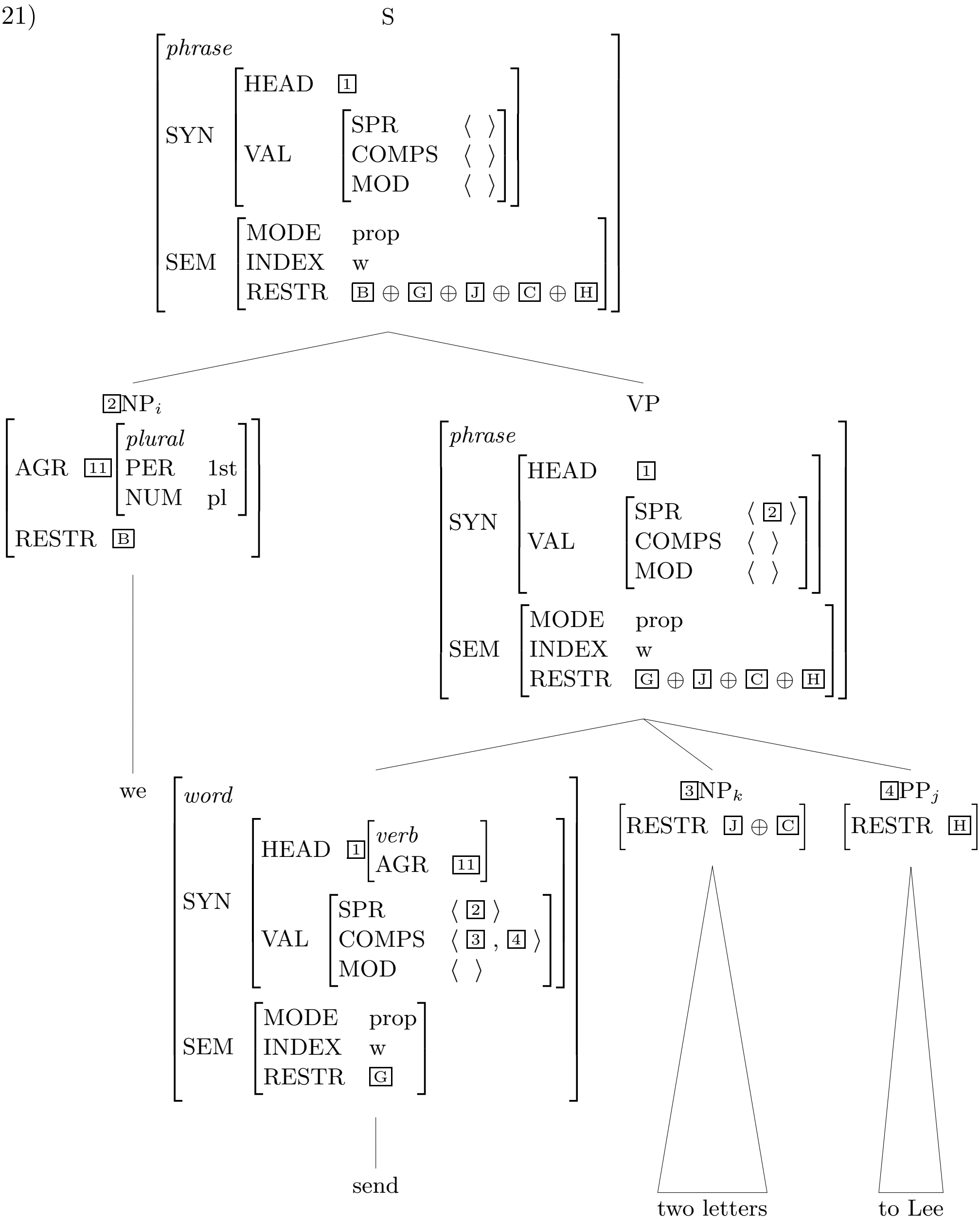
(19)
$$\begin{bmatrix}\text{RELN} & \textbf{group}\\ \text{INST} & i\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{speaker}\\ \text{INST} & l\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{member}\\ \text{SET} & i\\ \text{ELEMENT} & l\end{bmatrix},$$
$$\begin{bmatrix}\text{RELN} & \textbf{send}\\ \text{SIT} & s_7\\ \text{SENDER} & i\\ \text{SENDEE} & j\\ \text{SENT} & k\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{two}\\ \text{BV} & k\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{letter}\\ \text{INST} & k\\ \text{ADDRESSEE} & m\end{bmatrix},$$
$$\begin{bmatrix}\text{RELN} & \textbf{name}\\ \text{NAME} & \text{Lee}\\ \text{NAMED} & j\end{bmatrix}$$

(20)
$$\begin{bmatrix}\text{RELN} & \textbf{group}\\ \text{INST} & i\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{speaker}\\ \text{INST} & l\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{member}\\ \text{SET} & i\\ \text{ELEMENT} & l\end{bmatrix},$$
$$\begin{bmatrix}\text{RELN} & \textbf{send}\\ \text{SIT} & s_7\\ \text{SENDER} & i\\ \text{SENDEE} & j\\ \text{SENT} & k\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{two}\\ \text{BV} & k\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{letter}\\ \text{INST} & k\\ \text{ADDRESSEE} & m\end{bmatrix},$$
$$\begin{bmatrix}\text{RELN} & \textbf{name}\\ \text{NAME} & \text{Lee}\\ \text{NAMED} & m\end{bmatrix}$$

The only difference between the two semantic representations is which other role the NAMED value of the **name** predication (i.e. Lee) is identified with: the SENDEE value of the **send** predication or the ADDRESSEE value of the **letter** predication.

In this subsection, we will show how our grammar licenses two distinct trees for this sentence, and how it associates each with one of the semantic representations in (19) and (20). For expository convenience, we begin with the rather schematic tree in (21) (similar to (16)), waiting to show the detailed feature structures it contains until we look at its subtrees:

(21)



The top node in this tree is licensed by the Head-Specifier Rule. It differs from its second daughter, the VP, in only two ways: its SPR value is the empty list (as required by the Head-Specifier Rule), and its RESTR value includes the RESTR of the subject NP (as required by the Semantic Compositionality Principle). The HEAD features of the top node and of the VP are identical, as required by the Head Feature Principle. The COMPS list is empty both at the top and in the VP, in accordance with the Valence Principle. And both MODE and INDEX have the same value at the top as in the VP, in keeping with the Semantic Inheritance Principle. The first daughter (the subject NP) is identical to the sole element on the second daughter's SPR list, as required by the Head-Specifier Rule.

The subtree dominating *we* – that is the subject of the sentence – is labeled 'NP' here, but it could just as well have been labeled 'N'. It is simply a word structure, identical in its feature structure to the one in (11), except that the value of the CASE feature is 'nom', not 'acc'. This structure is the word structure licensed by the lexical entry for *we*.

The other daughter of the top node – the VP – is the mother of a tree licensed by the Head-Complement Rule. The VP's feature values are the same as those of its head (leftmost) daughter, except for COMPS and RESTR. The COMPS list of the VP is empty, as specified in the Head-Complement Rule. The RESTR value is the sum of its three daughters' RESTR values, by the Semantic Compositionality Principle. Again, the VP's HEAD, SPR, MODE, and INDEX values are the same as those of the head daughter, in accordance with the HFP, the Valence Principle, and the Semantic Inheritance Principle. The COMPS value of the head daughter is the list consisting of the other two daughters; this is specified by the Head-Complement Rule.

The subtree dominating the verb *send* is the following:

(22)

$$
\begin{array}{c}
\left[\begin{array}{ll}
\textit{word} & \\
\text{SYN} & \left[\begin{array}{ll}
\text{HEAD} & \left[\begin{array}{ll} \textit{verb} & \\ \text{AGR} & \boxed{11} \end{array}\right] \\
\text{VAL} & \left[\begin{array}{ll}
\text{SPR} & \left\langle \begin{array}{c} \text{NP}_i \\ \left[\begin{array}{ll} \text{CASE} & \text{nom} \\ \text{AGR} & \boxed{11}\left[\begin{array}{ll} \textit{plural} & \\ \text{PER} & \text{1st} \\ \text{NUM} & \text{pl} \end{array}\right] \\ \ldots & \end{array}\right] \end{array} \right\rangle \\
\text{COMPS} & \left\langle \begin{array}{c} \text{NP}_k \\ \left[\begin{array}{ll} \text{CASE} & \text{acc} \\ \text{AGR} & \left[\begin{array}{ll} \textit{plural} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{pl} \end{array}\right] \\ \ldots & \end{array}\right] \end{array}, \text{PP}_j \right\rangle \\
\text{MOD} & \langle\ \rangle
\end{array}\right]
\end{array}\right] \\
\text{SEM} & \left[\begin{array}{ll}
\text{MODE} & \text{prop} \\
\text{INDEX} & s_7 \\
\text{RESTR} & \boxed{G}\left\langle \left[\begin{array}{ll} \text{RELN} & \textbf{send} \\ \text{SIT} & s_7 \\ \text{SENDER} & i \\ \text{SENDEE} & j \\ \text{SENT} & k \end{array}\right] \right\rangle
\end{array}\right]
\end{array}\right] \\
| \\
\text{send}
\end{array}
$$

This is different from the verb subtree in our previous example (i.e. from (13)) in several ways. The most obvious is that the form is *send*, not *sent*. Although our SEM value does not reflect the clear meaning difference between the present and past tense forms, there are nonetheless several syntactic differences that are represented. Many of these differences follow from differences in the lexical entries that license the word structures. (22) is licensed by the lexical entry in (23):

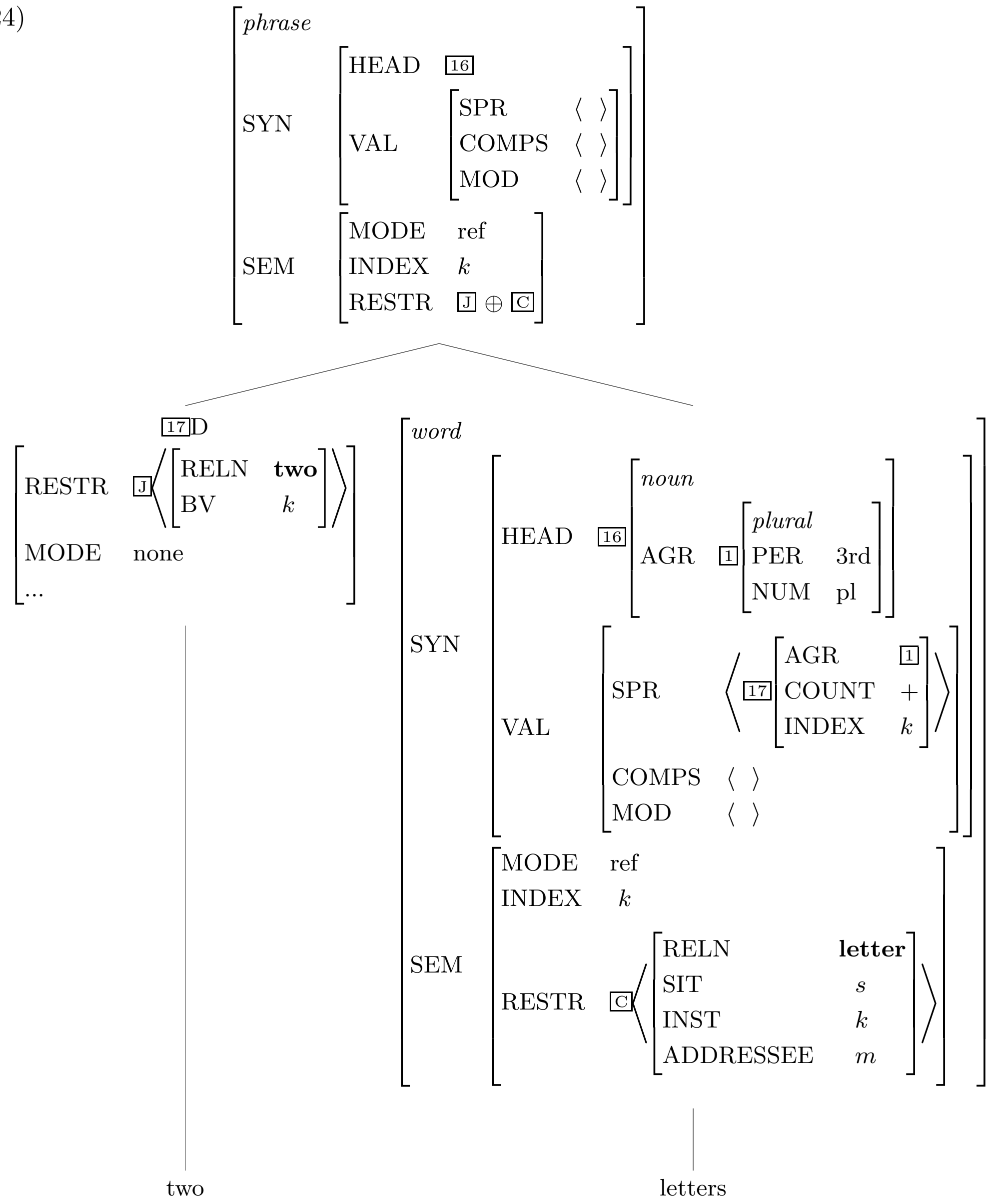
(23)

$$\left\langle \text{send}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \overset{\text{NP}_i}{\begin{bmatrix} \text{CASE} & \text{nom} \\ \text{AGR} & \textit{non-3sing} \end{bmatrix}} \right\rangle \\ \text{COMPS} & \left\langle \overset{\text{NP}_k}{\begin{bmatrix} \text{CASE} & \text{acc} \end{bmatrix}} (,\ \text{PP}_j) \right\rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s_7 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{send} \\ \text{SIT} & s_7 \\ \text{SENDER} & i \\ \text{SENDEE} & j \\ \text{SENT} & k \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(23)'s specifier is specified as [AGR *non-3sing*]; that is because the verb *send* (unlike *sent*) cannot be combined with a third-person singular subject (like *Terry*). Another difference is that the second element of the COMPS list in (22) is an optional PP, not an obligatory NP. Related to that is the fact that the first complement in (22) refers to the thing sent (indicated by the role 'SENT' in the predication on the verb's RESTR list), and the second complement corresponds to the sendee (also indicated in the RESTR). Problem 3 in Chapter 10 addresses the relation between pairs of lexical entries like (12) and (23).

The subtree for the object NP, *two letters*, is shown in (24):

(24)



This tree is licensed by the Head-Specifier Rule, which says that the top node must have an empty SPR list and that the second (i.e. head) daughter must have a SPR list whose sole member is identical to the first daughter. The identity of the AGR values of the head noun *letters* and its determiner *two* (indicated by $\boxed{1}$) is required by the SHAC. The HEAD value of the top node is identical to that of the second daughter, according to the Head Feature Principle. The COMPS values of these two nodes are identical, as guaranteed by the Valence Principle. The MODE and INDEX values of the second daughter and its mother are likewise shared, courtesy of the Semantic Inheritance

Principle. Finally, the Semantic Compositionality Principle requires that the RESTR value of the determiner combines with the RESTR value for the noun to give the RESTR value of the NP.

Licensing (24) via the Head-Specifier Rule requires the word structures for each of its words. The following is the word structure for *two*, which is similar to (9) above:

(25)

$$\begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{det} & \\ \text{COUNT} & + \\ \text{AGR} & \textit{plural} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{none} \\ \text{INDEX} & k \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{two} \\ \text{BV} & k \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

two

The relevant word structure for *letters* is sketched in (26):

(26)

$$\begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \boxed{1} \begin{bmatrix} \textit{plural} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \begin{array}{c} \text{D} \\ \begin{bmatrix} \text{AGR} & \boxed{1} \\ \text{COUNT} & + \\ \text{INDEX} & k \\ \ldots & \end{bmatrix} \end{array} \right\rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & k \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{letter} \\ \text{INSTANCE} & k \\ \text{ADDRESSEE} & m \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

letters

This tree is quite similar to (8). The principle difference is that the type of the AGR value is *plural*, and it therefore lacks the GEND feature. If our treatment of semantics were more detailed, the RESTR value would also be different, since it would have to include some information about the meaning of the plurality of *letters*; but for present purposes, we will ignore that difference. This word structure is licensed by the entry for *letters*, shown in (27):

(27)

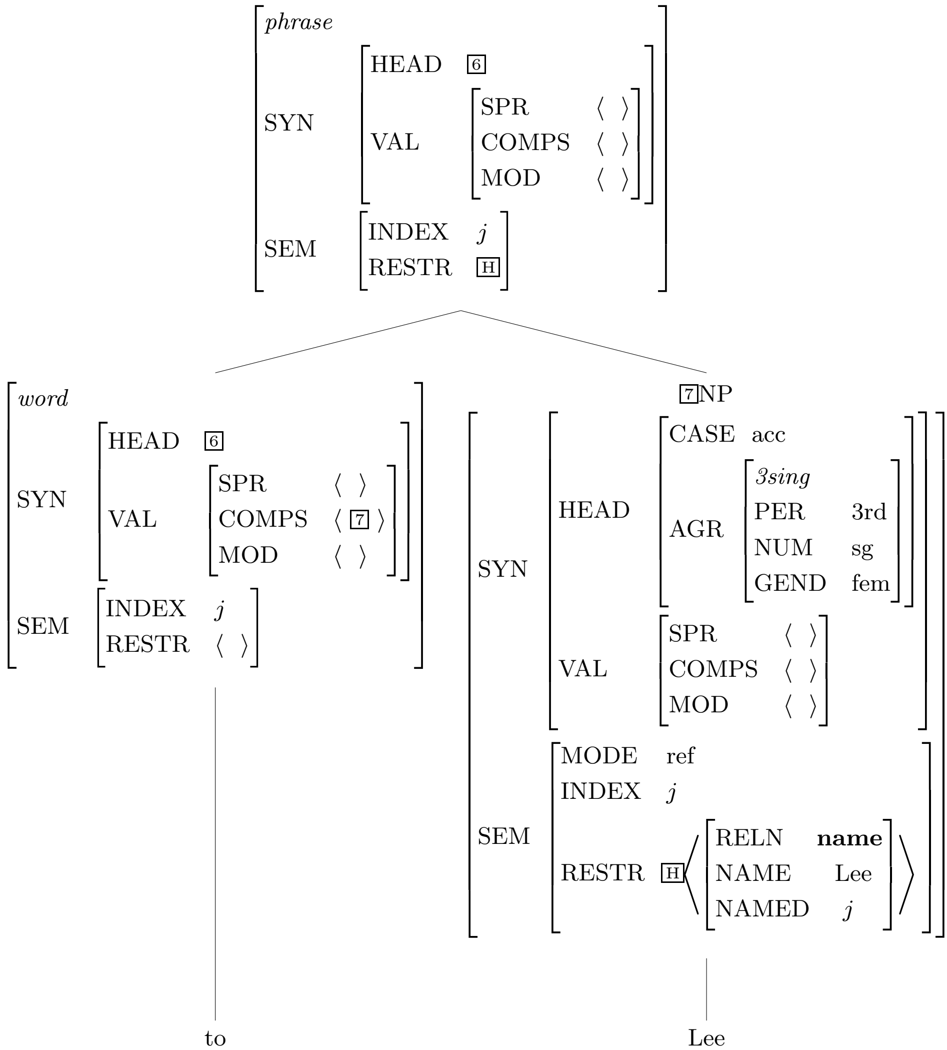
$$\left\langle \text{letters}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \textit{plural} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \overset{\text{D}}{\begin{bmatrix} \text{COUNT} & + \\ \text{INDEX} & k \end{bmatrix}} \right\rangle \\ \text{COMPS} & \langle\ (\text{PP}_m)\ \rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & k \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{letter} \\ \text{INST} & k \\ \text{ADDRESSEE} & m \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Notice that this lexical entry, like the one for *letter* in (5), provides for a possible PP complement. The word structure in (26) above uses the empty COMPS list option. We will return to the PP complement possibility below.

The subtree for the PP, *to Lee*, is highly schematized in (21). A more detailed version of the tree is given in (28):[9]

[9] As with the proper noun *Leslie* discussed in Section 6.1 above, the lexical entry for *Lee* is underspecified for GEND. All of the word structures that satisfy that lexical entry are fully specified, and therefore contain a value for GEND. Here we have arbitrarily chosen a word structure that is [GEND fem].

(28)



The most interesting thing about this subtree is how we have analyzed the semantics. The preposition *to* in this sentence is functioning to mark the role of its object NP with respect to the verb. That is, it does what many languages would do by means of case inflections on the noun. Since English has only a vestigial system of case marking, it relies on prepositions and word order to mark the roles of various NPs in the sentence. Note that the preposition can be omitted if the verb's arguments are presented in another order: *We sent Lee two letters*. Consequently, we have given the preposition no semantics of its own. Its RESTR value is the empty list, and its index is simply identified as the index of the object NP. We have said nothing about the MODE value, but in the next chapter, we will argue that it, too, should be identified with the MODE of the object NP.

The PP assumes the same INDEX value as the preposition (and hence as the NP) by the Semantic Inheritance Principle. Other identities in (28) should by now be familiar:

the one element of the preposition's COMPS list must be the object NP, by the Head-Complement Rule; the same rule specifies that the PP has an empty COMPS list; the Valence Principle is responsible for the fact that the PP and P have the same (empty) SPR list; the PP and the P share the same HEAD features in virtue of the Head Feature Principle; and the PP's RESTR value is the same as the NP's, in accordance with the Semantic Compositionality Principle (together with the fact that the preposition's RESTR is the empty list).

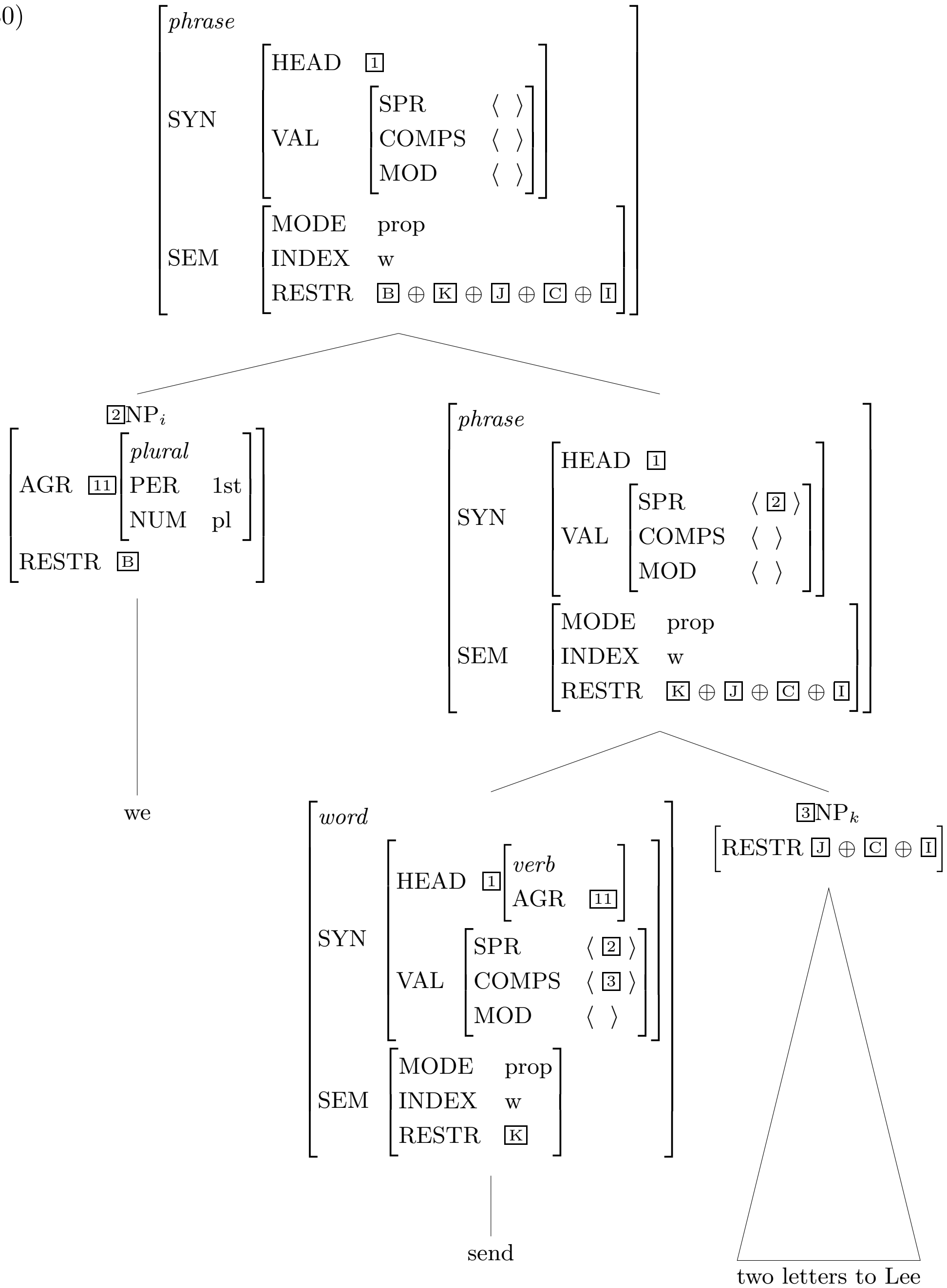
The NP in (28) is [CASE acc] because objects of prepositions in English are always accusative (although there is no morphological marking of it in this sentence). This requirement is encoded in the lexical entry for the preposition, as we will see when we look at the word structure for *to*, which is shown in (29):

(29)

$$
\begin{array}{c}
\left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{word}} \\
\text{SYN} & \left[\begin{array}{ll}
\text{HEAD} & \textit{prep} \\
\text{VAL} & \left[\begin{array}{ll}
\text{SPR} & \langle\ \rangle \\
\text{COMPS} & \left\langle \begin{array}{c}\text{NP} \\ \left[\begin{array}{ll}\text{CASE} & \text{acc} \\ \text{INDEX} & j \\ ... & \end{array}\right]\end{array} \right\rangle \\
\text{MOD} & \langle\ \rangle
\end{array}\right]
\end{array}\right] \\
\text{SEM} & \left[\begin{array}{ll}
\text{INDEX} & j \\
\text{RESTR} & \langle\ \rangle
\end{array}\right]
\end{array}\right] \\
| \\
\text{to}
\end{array}
$$

This completes the analysis of one parse of *We send two letters to Lee.* A schematic tree for the other parse is given as (30):

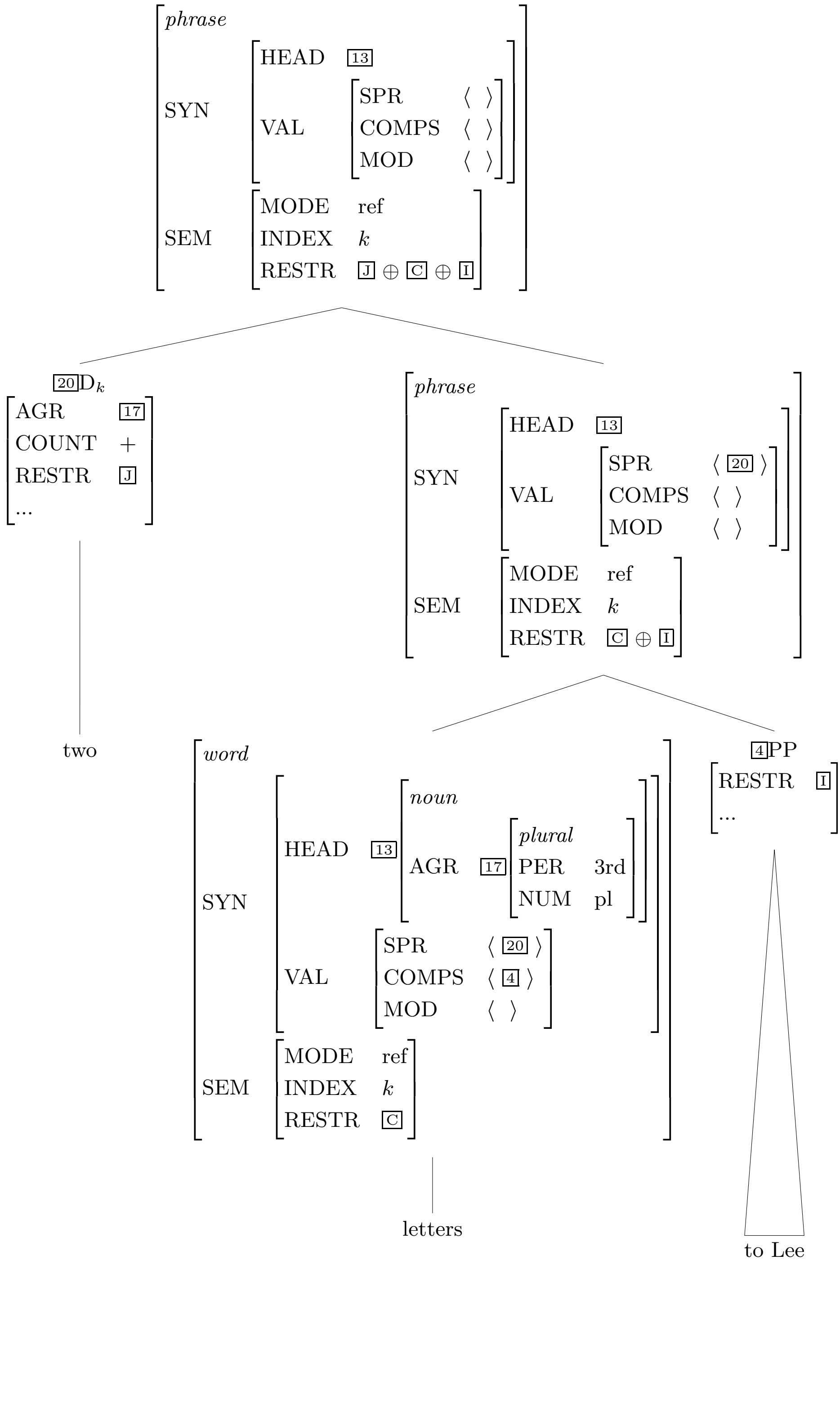
(30)



The subject NP *we* and the PP *to Lee* are exactly the same in this structure as in (21). The verb *send*, however, has two complements in (21) and only one in (30). That is because the lexical entry in (23) above, which licenses both verbal word structures, specifies that its second (PP) complement is optional. The noun *letters* in the two examples is licensed by the same lexical entry (27), which takes an optional PP complement.

In (21), there was no node spanning the string *two letters to Lee*. In (30), however, there is such a node. A more detailed subtree for that NP is the following:

(31)



The top node in this subtree is licensed by the Head-Specifier Rule which requires the identity of the determiner with the one element on the head's SPR list. The second daughter, dominating *letters to Lee* is licensed by the Head-Complement Rule, which also requires that the element on the COMPS list of the head noun is identical to the PP complement. The other identities are enforced by various principles in ways that should now be familiar.

Notice that the tag on the RESTR of *to Lee* in (30) and (31) is different from the tag in (21). That is because the role played by Lee is subtly different in the two sentences. In (30), the SENDEE role does not correspond to any syntactic constituent; in (21), the PP *to Lee* (and the noun *Lee*, with which it is coindexed) plays the SENDEE role. On the other hand, in (30), the PP plays the ADDRESSEE role with respect to the noun *letters* – a role that is syntactically unrealized in (21). While most letters are sent to their addressees, it is possible for the sendee and the addressee to be different, as in *I sometimes inadvertently send letters to my sister to my brother.* We have annotated this difference by giving *Lee* the two minimally different RESTR values in (32):[10]

(32) a. $\boxed{\text{H}} \left\langle \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{Lee} \\ \text{NAMED} & j \end{bmatrix} \right\rangle$

b. $\boxed{\text{I}} \left\langle \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{Lee} \\ \text{NAMED} & m \end{bmatrix} \right\rangle$

Since j is the index for the SENDEE role in all of our trees in this section, $\boxed{\text{H}}$ is used when *Lee* is the SENDEE argument of the verb *send*. We use m as the index for the ADDRESSEE role, so we use $\boxed{\text{I}}$ when *Lee* plays the ADDRESSEE role with respect to the noun *letters*. [11]

[10] For readers who are still skeptical of the existence of this second structure (and interpretation), we provide an alternative appropriate embedding context:

The Corrupt Postal Worker Ransom Context:

[Postal workers A, B and C have stolen some important letters. C, who is negotiating ransom money for release of the letters addressed to Lee, is going over the plan with A and B:]

C: So if the phone rings twice, what do you send us?

B: We send two letters to Lee.

[11] This difference could have been annotated in another way. We could have used the same RESTR value for *to Lee* in both cases and assigned alphabetically different values to the SENDEE and ADDRESSEE roles in the two sentences. These two alternatives are not substantively different. They only appear to be distinct because of the way we use tag identity across different sentences in this section.

6.3 Appendix: Well-Formed Structures

In this appendix, we lay out more precisely the constructs of the theory whose effects we have been illustrating in this chapter. This presentation (like the elaborations of it given in Chapter 9 and Appendix A) is intended for readers concerned with the formal foundations of our theory. For most purposes and for most readers, the relatively informal presentation in the body of text, taken together with the definitions in section 6.3.6 below, should be sufficient.

6.3.1 Preliminaries

According to our approach, a grammar G is defined by the following components:

- a finite set of features: $\mathcal{F} = \{\text{SYN, SEM, HEAD, AGR, } \ldots\}$,
- a finite set of primitive items:
  $\mathcal{A}_{atom} = \mathcal{A}_{pol} \cup \mathcal{A}_{gr.atom} \cup \mathcal{A}_{mode} \cup \mathcal{A}_{reln}$, where:
  1. $\mathcal{A}_{pol} = \{+, -\}$,
  2. (a set of ground atoms) $\mathcal{A}_{gr.atom} = \{\text{1st}, \text{2nd}, \text{3rd}, \text{sg}, \text{pl}, \ldots, \text{run}, \text{dog}, \ldots\}$,
  3. $\mathcal{A}_{mode} = \{\text{prop}, \text{ques}, \text{dir}, \text{ref}, \text{none}\}$, and
  4. $\mathcal{A}_{reln} = \{\textbf{walk}, \textbf{love}, \textbf{person}, \ldots\}$,
- a denumerably infinite set of primitive items: $\mathcal{A}_{index} = \mathcal{A}_{ind} \cup \mathcal{A}_{sit}$, where:
  1. $\mathcal{A}_{ind} = \{i, j, \ldots\}$ and
  2. $\mathcal{A}_{sit} = \{s_1, s_2, \ldots\}$,
- the distinguished element *elist* (*empty-list*), discussed below,
- a finite set of types: $\mathcal{T} = \{$*noun*, *agr-pos*, *plural*, *expression*, ...$\}$,
- a type hierarchy with a tree structure associated with constraint inheritance (for instance, the type hierarchy represented by the tree and table in Section 5.10.1 and 5.10.2),
- a set $\mathcal{LT} \subset \mathcal{T}$ called the *leaf types* (a type τ is a *leaf type* if it is associated with a leaf in the type hierarchy tree, i.e. if τ is one of the most specific types),
- a set of list types (if τ is a type, then *list*(τ) is a type),
- a set of grammar rules (like the ones we have already encountered, see Section 5.10.4),
- a set of principles (like those in Section 5.10.5), and
- a lexicon (which is a finite set of lexical entries like those in Section 5.10.6).

Thus a grammar G comes with various primitives grouped into two sets: $\mathcal{A}_{atom}$ ($\mathcal{A}_{pol}$, $\mathcal{A}_{gr.atom}$, $\mathcal{A}_{mode}$, $\mathcal{A}_{reln}$) and $\mathcal{A}_{index}$ ($\mathcal{A}_{ind}$, and $\mathcal{A}_{sit}$). G assigns the type *atom* to all elements of $\mathcal{A}_{atom}$. The elements of $\mathcal{A}_{index}$ are used by the grammar for describing individual objects and situations; they are associated with the leaf type *index*. We assume that no items in these sets of primitives can be further analyzed via grammatical features.

Our grammar appeals to several ancillary notions which we now explicate: *feature structure description*, *feature structure*, *satisfaction of a description*, and *tree structure*.

6.3.2 Feature Structure Descriptions

For expressing the constraints associated with the grammar rules, principles, types, and lexical entries, we introduce the notion of a feature structure description. The feature structure descriptions are given as attribute-value matrices, augmented with the connective '|', set descriptors ($\{\ldots\}$), list descriptions ($\langle\ldots\rangle$, attribute-value matrices with FIRST/REST, or two list descriptions connected by $\oplus$), and a set *Tags* of tags (labels represented by boxed integers or letters).

6.3.3 Feature Structures

The set of feature structures $\mathcal{FS}$ is given by the following recursive definition:

(33) $\phi \in \mathcal{FS}$ (i.e. ϕ is a feature structure) iff

a. $\phi \in \mathcal{A}_{atom} \cup \mathcal{A}_{index}$, or

b. ϕ is a function from features to feature structures, $\phi : \mathcal{F} \longrightarrow \mathcal{FS}$ satisfying the following conditions

1. ϕ is of a leaf type τ;
2. $DOM(\phi) = \{F \mid \text{G declares } F \text{ appropriate for } \tau\} \cup$

$$\{F' \mid \exists\tau' \text{ such that } \tau' \text{ is a supertype of } \tau \text{ and}$$
$$\text{G declares } F' \text{ appropriate for } \tau'\},$$

i.e. ϕ is defined for any feature that is declared appropriate for τ or for any of τ's supertypes;

3. for each $F \in DOM(\phi)$, G defines the type of the value $\phi(F)$ (we call the value $\phi(F)$ of the function ϕ on F the value of the feature F); and
4. ϕ obeys all further constraints ('type constraints') that G associates with type τ (including those inherited from the supertypes τ' of τ), or

c. ϕ is of type *list*(τ), for some type τ, in which case either:

1. ϕ is the distinguished element *elist*, or else:
2. A. *DOM*(ϕ) is $\{\text{FIRST}, \text{REST}\}$,
   B. the type of ϕ(FIRST) is τ, and
   C. the type of ϕ(REST) is *list*(τ).

6.3.4 Satisfaction

We explain how feature structures satisfy descriptions indirectly – in terms of denotation, which we define as follows:

Denotation of Feature Structure Descriptions

The denotation of a feature structure description is specified in terms of a structure $\mathcal{M}$:

(34) $\mathcal{M} = \langle \mathcal{A}, \mathcal{F}, \mathcal{T}, \mathbf{Type}, I \rangle$, where:

1. $\mathcal{A} = \mathcal{A}_{atom} \cup \mathcal{A}_{index} \cup \{elist\}$,
2. $\mathcal{F}$ is a finite set of features,
3. $\mathcal{T}$ is a finite set of types,
4. **Type** is a function mapping feature structures to types –
   Type : $\mathcal{FS} \longrightarrow \mathcal{LT}$, where $\mathcal{LT}$ is the set of the leaf types, and

5. I is a function mapping feature names and atomic descriptors to features and atoms of the appropriate sort:

$$I = I_{\widetilde{\mathcal{F}}} \cup I_{\widetilde{\mathcal{A}}_{atom}} \cup I_{\widetilde{\mathcal{A}}_{ind}} \cup I_{\widetilde{\mathcal{A}}_{sit}} \cup \{\langle \mathit{elist}, \mathit{elist}\rangle\},$$

where
$I_{\widetilde{\mathcal{F}}} \in \mathcal{F}^{\widetilde{\mathcal{F}}}$, $I_{\widetilde{\mathcal{A}}_{atom}} \in \mathcal{A}_{atom}{}^{\widetilde{\mathcal{A}}_{atom}}$, $I_{\widetilde{\mathcal{A}}_{ind}} \in \mathcal{A}_{ind}{}^{\widetilde{\mathcal{A}}_{ind}}$, $I_{\widetilde{\mathcal{A}}_{sit}} \in \mathcal{A}_{sit}{}^{\widetilde{\mathcal{A}}_{sit}}$,

and $\widetilde{X}$ denotes the set of expressions that have denotations in the set X.[12]

The function I is called an *interpretation function.* An *assignment function* is a function

$$g : \mathit{Tags} \longrightarrow \mathcal{FS}.$$

We say that a feature structure ϕ is of type $\tau \in \mathcal{T}$ iff there is a (unique) leaf type $\tau' \in \mathcal{LT}$ such that:

(35)
1. τ' is a subtype of τ, and
2. $\mathbf{Type}(\phi) = \tau'$.

Given $\mathcal{M}$, the interpretation $[\![d]\!]^{\mathcal{M},g}$ of a feature structure description d with respect to an assignment function g is defined recursively as follows:

(36)
1. if $v \in \widetilde{\mathcal{F}} \cup \widetilde{\mathcal{A}}_{atom} \cup \widetilde{\mathcal{A}}_{index} \cup \{\mathit{elist}\}$, *then* $[\![v]\!]^{\mathcal{M},g} = I(v)$;
2. if τ is a type, i.e. $\tau \in \mathcal{T}$, then $[\![\tau]\!]^{\mathcal{M},g} = \{\phi \in \mathcal{FS} : \phi \text{ is of type } \tau\}$;
3. if $F \in \widetilde{\mathcal{F}}$, and d is a feature structure description, then $[\![[F\ d]]\!]^{\mathcal{M},g} = \{\phi \in \mathcal{FS} : \text{there is some } \phi' \text{ such that } \phi' \in [\![d]\!]^{\mathcal{M},g} \text{ and } \langle I(F), \phi'\rangle \in \phi\}$;[13]
4. if $d = \begin{bmatrix} d_1 \\ \dots \\ d_n \end{bmatrix}$
   where $n \geq 1$, and $d_1, \dots, d_n$ are feature structure descriptions, then
$$[\![d]\!]^{\mathcal{M},g} = \bigcap_{i=1}^{n} [\![d_i]\!]^{\mathcal{M},g};$$
5. if d is a set descriptor $\{d_1, \dots, d_n\}$, then
$$[\![d]\!]^{\mathcal{M},g} = \bigcup_{i=1}^{n} [\![d_i]\!]^{\mathcal{M},g}$$
   ($[\![\{\ \}]\!]^{\mathcal{M},g} = \emptyset$);
6. $[\![d_1 \mid d_2]\!]^{\mathcal{M},g} = [\![d_1]\!]^{\mathcal{M},g} \cup [\![d_2]\!]^{\mathcal{M},g}$;
7. if $d \in \mathit{Tags}$, then $[\![d]\!]^{\mathcal{M},g} = g(d)$;
8. if $d \in \mathit{Tags}$ and d' is a feature structure description, then $[\![d\ d']\!]^{\mathcal{M},g} = \{\phi \in \mathcal{FS} : g(d) = \phi \text{ and } \phi \in [\![d']\!]^{\mathcal{M},g}\}$;
   (Note that tagging narrows the interpretation down to a singleton set.)

---

[12] Y^X is the standard notation for the set of all functions $f : X \rightarrow Y$.

[13] Note that the definition of a feature structure in (33), taken together with this clause, ensures that each element ϕ of the set $[\![[F\ d]]\!]^{\mathcal{M},g}$ is a proper feature structure.

9. List Addition:[14]
   a. $[\![elist \oplus d]\!]^{\mathcal{M},g} = [\![d]\!]^{\mathcal{M},g}$,
   b. if $d = \begin{bmatrix} \text{FIRST} & d_1 \\ \text{REST} & d_2 \end{bmatrix} \oplus d_3$,
   then $[\![d]\!]^{\mathcal{M},g} =$
   $\{\phi \in \mathcal{FS} : \phi(\text{FIRST}) \in [\![d_1]\!]^{\mathcal{M},g} \text{ and } \phi(\text{REST}) \in [\![d_2 \oplus d_3]\!]^{\mathcal{M},g}\}$.

Satisfaction of Feature Structure Descriptions

A feature structure $\phi \in \mathcal{FS}$ satisfies a feature structure description d iff there is some assignment function g such that $\phi \in [\![d]\!]^{\mathcal{M},g}$.

Examples:

(37) a. ϕ satisfies [NUM sg] iff $\langle \text{NUM}, \text{sg} \rangle \in \phi$.
b. ϕ satisfies [AGR [NUM sg]] iff there is a feature structure ϕ' (which is unique) such that $\langle \text{AGR}, \phi' \rangle \in \phi$ and $\langle \text{NUM}, \text{sg} \rangle \in \phi'$.
c. ϕ satisfies [AGR *3sing*] iff there is a feature structure ϕ' (which is unique) such that $\langle \text{AGR}, \phi' \rangle \in \phi$ and ϕ' is of type *3sing*.
d. ϕ satisfies [PER $\{$1st, 2nd, 3rd$\}$] iff $\langle \text{PER}, \text{1st} \rangle \in \phi$, $\langle \text{PER}, \text{2nd} \rangle \in \phi$, or $\langle \text{PER}, \text{3rd} \rangle \in \phi$.
e. ϕ satisfies [ARGS $\langle s_1, s_2, s_3 \rangle$] iff:
$\langle \text{ARGS}, \{\langle \text{FIRST}, s_1 \rangle, \langle \text{REST}, \{\langle \text{FIRST}, s_2 \rangle, \langle \text{REST}, \{\langle \text{FIRST}, s_3 \rangle, \langle \text{REST}, elist \rangle\}\rangle\}\rangle\}\rangle \in \phi$.
f. ϕ satisfies:

$$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \text{ [SYN [HEAD [AGR } \boxed{1}\text{]]] } \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

iff

1. $\phi(\text{SYN})(\text{HEAD})(\text{AGR}) =$
   $\phi(\text{SYN})(\text{VAL})(\text{SPR})(\text{FIRST})(\text{SYN})(\text{HEAD})(\text{AGR})$,[15] and
2. $\phi(\text{SYN})(\text{VAL})(\text{SPR})(\text{REST}) = elist$

6.3.5 Tree Structures

Finally, we assume a notion of tree structure described informally as follows:

(38) A tree structure is a directed graph that satisfies a number of conditions:[16]

[14] Where no confusion should arise, we use 'FIRST', 'SYN', etc. to refer either to the appropriate feature (an element of $\mathcal{F}$) or to its name (an element of $\widetilde{\mathcal{F}}$).

[15] Note that parentheses here are 'left associative': '$\phi(X)(Y)$' is equivalent to '$(\phi(X))(Y)$'. That is, both expressions denote the result of applying the function ϕ to (the feature) X and then applying the result to (the feature) Y.

[16] Here, we assume familiarity with notions such as *root*, *mother*, *terminal node*, *nonterminal node*, and *branches*. These and related notions can be defined more precisely in set-theoretic terms, as is done in various texts. See, for example, Hopcroft et al. 2001 and Partee et al. 1990.

1. it has a unique root node,
2. each non-root node has exactly one mother,
3. sister nodes are ordered with respect to each other,
4. it has no crossing branches,
5. each nonterminal node is labelled by a feature structure, and
6. each terminal node is labeled by a phonological form (an atom).

6.3.6 Structures Defined by the Grammar

We may now proceed to define well-formedness of tree structures in terms of the licensing of their component trees (recall from Chapters 2 and 3 that a local subtree consists of a mother and all its daughters):

(39) **Well-Formed Tree Structure**:

Φ is a Well-Formed Tree Structure according to G if and only if:

1. Φ is a tree structure,
2. the label of Φ's root node satisfies S,[17] and
3. each local subtree within Φ is either phrasally licensed or lexically licensed.

(40) **Lexical Licensing**:
A word structure of the form:

$$\begin{array}{c} \phi \\ | \\ \omega \end{array}$$

is licensed if and only if G contains a lexical entry $\langle d_1, d_2 \rangle$, where ω satisfies d_1 and ϕ satisfies d_2.

(41) **Phrasal Licensing**:
A grammar rule $\rho = d_0 \rightarrow\ d_1 \ldots\ d_n$ licenses a local subtree:

$$\Phi = \begin{array}{ccc} & \phi_0 & \\ \diagup & & \diagdown \\ \phi_1 & \ldots & \phi_n \end{array}$$

if and only if:

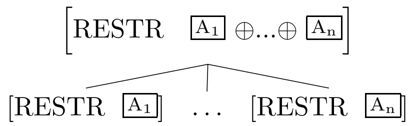
1. for each $i, 0 \leq i \leq n$, ϕ_i is of[18] the type *expression*,
2. there is some assignment function g under which the sequence $\langle \phi_0, \phi_1, ..., \phi_n \rangle$ satisfies the description sequence $\langle d_0,\ d_1, \ldots,\ d_n \rangle$,[19]
3. Φ satisfies the Semantic Compositionality Principle, and
4. if ρ is a headed rule, then Φ satisfies the Head Feature Principle, the Valence Principle and the Semantic Inheritance Principle, with respect to ρ.

[17] Recall once again that S abbreviates a certain feature structure constraint, as discussed in Chapter 4.

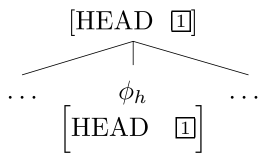
[18] That is, assigned to some leaf type that is a subtype of the type *expression*.

[19] Note that this clause must speak of a sequence of feature structures satisfying a sequence description. This is because of identities that must hold across members of the sequence, e.g. those required by particular grammar rules.

(42) Φ satisfies the Semantic Compositionality Principle with respect to a grammar rule ρ if and only if Φ satisfies:

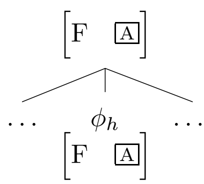


(43) Φ satisfies the Head Feature Principle with respect to a headed rule ρ if and only if Φ satisfies:



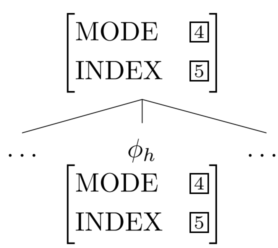
where ϕ_h is the head daughter of Φ.

(44) Φ satisfies the Valence Principle with respect to a headed rule ρ if and only if, for any VAL feature F, Φ satisfies:



where ϕ_h is the head daughter of Φ and ρ does not specify incompatible F values for ϕ_h and ϕ_0.

(45) Φ satisfies the Semantic Inheritance Principle with respect to a headed rule ρ if and only if Φ satisfies:



where ϕ_h is the head daughter of Φ.

6.4 Problems

Problem 1: A Sentence

For the purposes of this problem, assume that the preposition *on* in the example below is like *to* in (18) in that makes no contribution to the semantics other than to pass up the INDEX and MODE values of its object NP. That is, assume it has the following lexical entry:

$$\left\langle \text{on}\ ,\ \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{prep} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \left\langle \overset{\text{NP}}{\begin{bmatrix} \text{CASE} & \text{acc} \\ \text{MODE} & \boxed{1} \\ \text{INDEX} & \boxed{2} \end{bmatrix}} \right\rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \boxed{1} \\ \text{INDEX} & \boxed{2} \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

A. Draw a fully resolved tree structure for the sentence in (i). Use tags to indicate identities required by the grammar. When two feature structures are tagged as identical, you need only show the information in one place.

(i) I rely on Kim.

B. In the VP and PP nodes of your tree, indicate which aspects of the grammar constrain each piece of information (i.e. each feature value). [*Hint: Possible answers include grammar rules and the combined effect of general principles and lexical entries.*]

Problem 2: Spanish NPs II

In this problem we return to Spanish NPs (see Problem 2 in Chapter 4), this time adding adjectives. Unlike English adjectives, Spanish adjectives agree with the nouns they modify, as shown in (i)–(iv):

(i) a. La jirafa pequeña corrió.
The.FEM.SG giraffe small.FEM.SG ran.3SG
'The small giraffe ran.'
b.*La jirafa pequeñas/pequeño/pequeños corrió.

(ii) a. Las jirafas pequeñas corrieron.
The.FEM.PL giraffes small.FEM.PL ran.3PL
'The small giraffes ran.'
b.*Las jirafas pequeña/pequeño/pequeños corrieron.

(iii) a. El pingüino pequeño corrió.
The.MASC.SG penguin small.MASC.SG ran.3SG
'The small penguin ran.'
b.*El pingüino pequeña/pequeñas/pequeños corrió.

(iv) a. Los pingüinos pequeños corrieron.
The.MASC.PL penguins small.MASC.PL ran.3PL
'The small penguins ran.'
b.*Los pingüinos pequeña/pequeñas/pequeño corrieron.

A. Using the MOD feature to specify which nouns the adjective can modify, give a lexical entry for *pequeños*. Be sure to specify both SYN and SEM features.

[*Hint: The semantics of adjectives is very similar to that of adverbs, so the entry for today in Chapter 5 (page 147) may be a helpful guide in doing this.*]

B. Assuming the rules we have developed for English are appropriate for Spanish as well, draw a tree for the NP *los pingüinos pequeños* in (iv). Show values for all features, using tags to show identities required by the grammar.

C. Explain how the INDEX value of *pingüinos* is identified with the argument of the *predication* introduced by *pequeños*. (Your explanation should indicate the role of lexical entries, rules, and principles in enforcing this identity.)

Problem 3: English Possessives I

English uses *'s* to express possession, as in the following examples:

(i) Leslie's coffee spilled.
(ii) Jesse met the president of the university's cousin.
(iii)*Jesse met the president's of the university cousin.
(iv) Don't touch that plant growing by the trail's leaves.
(v)*Don't touch that plant's growing by the trail leaves.
(vi) The person you were talking to's pants are torn.
(vii)*The person's you were talking to pants are torn.

(While examples (iv) and (vi) are a bit awkward, people do use such sentences, and there is certainly nowhere else that the *'s* could be placed to improve them).

A. What is the generalization about where the *'s* of possession appears in English?

One traditional treatment of the possessive marker (*'s*) is to claim it is a case marker. In our terms this means that it indicates a particular value for the feature CASE (say, 'poss' for 'possessive') on the word it attaches to. If we tried to formalize this traditional treatment of *'s*, we might posit a rule along the following lines, based on the fact that possessive NPs appear in the same position as determiners:

$$D \rightarrow \begin{matrix}\text{NP}\\ [\text{CASE poss}]\end{matrix}$$

Taken together with our assumption that CASE is a HEAD feature, such an analysis of *'s* makes predictions about the grammaticality of (i)–(vii).

B. Which of these sentences does it predict should be grammatical, and why?

Problem 4: English Possessives II

An alternative analysis of the possessive is to say that *'s* is a determiner that builds a determiner phrase (abbreviated DP), via the Head-Specifier Rule. On this analysis, *'s* selects for no complements, but it obligatorily takes an NP specifier. The word *'s* thus has a lexical category that is like an intransitive verb in valence.

This analysis is somewhat unintuitive, for two reasons: first, it requires that we have an independent lexical entry for *'s*, which seems more like a piece of a word, phonologically; and second, it makes the nonword *'s* the head of its phrase! However, this analysis does a surprisingly good job of predicting the facts of English possessives, so we shall adopt it, at least for purposes of this text.

A. Ignoring semantics for the moment, give the lexical entry for *'s* assuming its analysis as a determiner, and draw a tree for the NP *Kim's brother*. (The tree should show the value of HEAD, SPR and COMPS on every node. Use tags to show identities required by the grammar. You may omit other features.)

B. Explain how your lexical entry gets the facts right in the following examples:

(i) The Queen of England's crown disappeared.

(ii)*The Queen's of England crown disappeared.

C. How does this analysis handle recursion in possessives, for example, *Robin's brother's wife*, or *Robin's brother's wife's parents*? Provide at least one tree fragment to illustrate your explanation. (You may use abbreviations for node labels in the tree.)

Problem 5: English Possessives III

The semantics we want to end up with for *Pat's book* is the one shown in (i) (*poss* is the name of the general possession relation that we will assume provides the right semantics for all possessive constructions):[20]

(i)
$$\begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAMED} & j \\ \text{NAME} & \text{Pat} \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{poss} \\ \text{POSSESSOR} & j \\ \text{POSSESSED} & i \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{the} \\ \text{BV} & i \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{book} \\ \text{INST} & i \end{bmatrix} \right\rangle \end{bmatrix}$$

[20] We have chosen to use 'the' as the quantifier introduced by possessives, but this is in fact a matter of debate. On the one hand, possessive NPs are more definite than standard indefinites such as *a book*. On the other hand, they don't come with the presupposition of uniqueness that tends to come with *the*. Compare (i) and (ii):

(i) That's the book.

(ii) That's my book.

Part (A) of this problem will ask you to give a SEM value for the determiner *'s* that will allow the grammar to build the SEM value in (i) for the phrase *Pat's book*. Recall that, on our analysis, nouns like *book* select for specifiers like *Pat's*, and the specifiers do not reciprocally select for the nouns. In order to get the correct semantics, *'s* will have to identify its BV value with its INDEX value. In this, it is just like the determiner *a* (see (9) on page 171). This constraint interacts with the constraint on all common nouns shown in (ii) to ensure that the value of BV is correctly resolved:

(ii)
$$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \begin{bmatrix} \text{SEM} & \begin{bmatrix} \text{INDEX} & \boxed{1} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

A. Given the discussion above, what is the SEM value of the determiner *'s*?

B. Draw a tree for the phrase *Pat's book*, showing all SEM features on all nodes and SPR on any nodes where it is non-empty. Use tags (or matching indices, as appropriate) to indicate identities required by the grammar.

C. Describe how your analysis guarantees the right SEM value for the phrase. (Your description should make reference to lexical entries, rules and principles, as appropriate.)

Problem 6: English Possessive Pronouns

Possessive pronouns like *my*, *your*, etc. function as determiners in NPs like *my books* and *your mother*. You might think we should treat possessive pronouns as determiners that have the same AGR value as the corresponding nonpossessive pronoun. That is, you might think that *my* should be specified as:

(i)
$$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{det} & \\ \text{AGR} & \begin{bmatrix} \textit{1sing} & \\ \text{PER} & \text{1st} \\ \text{NUM} & \text{sg} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

A. Explain why this analysis (in particular, the AGR value shown in (i)) will fail to provide an adequate account of *my books* and *your cousin*.

B. The semantics we want to end up with for *my book* is this:

(ii)
$$\begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{speaker} \\ \text{INST} & j \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{poss} \\ \text{POSSESSOR} & j \\ \text{POSSESSED} & i \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{the} \\ \text{BV} & i \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{book} \\ \text{INST} & i \end{bmatrix} \right\rangle \end{bmatrix}$$

Formulate the SEM value of the determiner *my*.

C. Draw an explicit tree for the phrase *my book*.

[*Hint: Refer to Problem 5.*]

Problem 7: French Possessive Pronouns

Problem 6 asked you to provide an argument as to why *my* isn't [PER 1st, NUM sg], but didn't concern what the AGR value should be instead.

A. Provide an argument, with suitable data, that the AGR value of English possessive pronouns (e.g. *my* or *our*) should be left unspecified for number.

Now consider the following data from French. French nouns, like Spanish nouns, are all assigned either masculine or feminine gender. In these examples, *pie* is feminine and *moineau* is masculine.

(i) ma pie
my magpie

(ii)*mon/mes pie

(iii) mon moineau
my sparrow

(iv)*ma/mes moineau

(v) mes pies
my magpies

(vi)*ma/mon pies

(vii) mes moineaux
my sparrows

(viii)*ma/mon moineaux

B. Give the AGR values for *ma*, *mon*, and *mes*.

7

Binding Theory

7.1 Introduction

This chapter revisits a topic introduced very informally in Chapter 1, namely, the distribution of reflexive and nonreflexive pronouns. In that discussion, we noticed that the well-formedness of sentences containing reflexives usually depends crucially on whether there is another expression in the sentence that has the same referent as the reflexive; we called such an expression the 'antecedent' of the reflexive. Nonreflexive pronouns, on the other hand, often lack an antecedent in the same sentence. The issue for a nonreflexive pronoun is typically whether a particular NP could have the same referent (or, as linguists often put it, be coreferential with it) – that is, whether that NP could serve as the antecedent for that pronoun.

In discussing these phenomena, we will use the notation of subscripted indices to mark which expressions are intended to have the same referent and which are intended to have distinct referents. Two expressions with the same index are to be taken as coreferential, whereas two expressions with different indices are to be understood as having distinct referents.

Thus the markings in (1) indicate that *himself* must refer to the same person as *John*, and that the referent of *her* must be someone other than Susan:

(1) a. John$_i$ frightens himself$_i$.
b.*Susan$_i$ frightens her$_i$.
c. Susan$_i$ frightens her$_j$.

As mentioned in Chapter 5, the subscript notation is shorthand for the value of the feature INDEX.

In examples like (1a), the reflexive *himself* is often said to be 'bound' by its antecedent. This terminology derives from an analogy between natural language pronouns and variables in mathematical logic. The principles governing the possible pairings of pronouns and antecedents are often called BINDING PRINCIPLES, and this area of study is commonly referred to as BINDING THEORY.[1] The term ANAPHORIC is also used for

[1]Much of the literature on Binding Theory actually restricts the term 'binding' to elements in certain syntactic configurations. Specifically, an element A is often said to bind an element B if and only if: (i) they have the same index; and (ii) A c-commands B. The technical term 'c-command' has been defined in several (nonequivalent) ways in the literature; the most commonly used definition is the following:

expressions (including pronouns) whose interpretation requires them to be associated with other elements in the discourse; the relationship of anaphoric elements to their antecedents is called ANAPHORA.

With this notation and terminology in place, we are now ready to develop a more precise and empirically accurate version of the Binding Theory we introduced in Chapter 1.

7.2 Binding Theory of Chapter 1 Revisited

Recall that in Chapter 1, on the basis of examples like (2)–(9), we formulated the hypothesis in (10):

(2) a. Susan$_i$ likes herself$_i$.
b.*Susan$_i$ likes her$_i$.

(3) a. Susan$_i$ told herself$_i$ a story.
b.*Susan$_i$ told her$_i$ a story.

(4) a. Susan$_i$ told a story to herself$_i$.
b.*Susan$_i$ told a story to her$_i$.

(5) a. Susan$_i$ devoted herself$_i$ to linguistics.
b.*Susan$_i$ devoted her$_i$ to linguistics.

(6) a. Nobody told Susan$_i$ about herself$_i$.
b.*Nobody told Susan$_i$ about her$_i$.

(7) a.*Susan$_i$ thinks that nobody likes herself$_i$.
b. Susan$_i$ thinks that nobody likes her$_i$.

(8) a.*Susan$_i$'s friends like herself$_i$.
b. Susan$_i$'s friends like her$_i$.

(9) a.*That picture of Susan$_i$ offended herself$_i$.
b. That picture of Susan$_i$ offended her$_i$.

(10) Reflexive pronouns must be coreferential with a preceding argument of the same verb; nonreflexive pronouns cannot be.

Our task in this chapter is to reformulate something close to the generalization in (10) in terms of the theoretical machinery we have been developing in the last five chapters. We would also like to extend the empirical coverage of (10) to deal with examples that our informal statement did not adequately handle. Toward this end, let us divide (10) into two principles, one for reflexive pronouns and the other for nonreflexive pronouns. Our first try at formulating them using the new binding terminology is then the following:

node A in a tree c-commands node B if and only if every branching node dominating A dominates B. Intuitively, this means roughly that A is at least as high in the tree as B. Our investigations into Binding Theory will not impose any such configurational limitation, as we will be deriving a similar, arguably superior characterization of constraints on binding in terms of ARG-ST lists (see below).

Note that we are interested in determining the conditions governing the pairing of pronouns and antecedents in a sentence. We will not, however, consider what possible things outside the sentence (be they linguistic expressions or entities in the world) can serve as antecedents for pronouns.

(11) Principle A (version I)

A reflexive pronoun must be bound by a preceding argument of the same verb.

Principle B (version I)

A nonreflexive pronoun may not be bound by a preceding argument of the same verb.

7.3 A Feature-Based Formulation of Binding Theory

Our binding principles make use of several intuitive notions that need to be explicated formally within the theory we have been developing. The terms 'reflexive pronoun' and 'nonreflexive pronoun' have not been defined. What distinguishes reflexive pronouns is a semantic property, namely, that they require linguistic antecedents (of a certain kind) in order to be interpreted. Hence, we introduce a new value of the semantic feature MODE that we will use to distinguish reflexive pronouns; we will call that value 'ana'. Nonreflexive pronouns, like nonpronominal nouns, are [MODE ref].[2] In addition, we will assume (building on the conclusions of Problem 2 in Chapter 1) that reciprocals (that is, *each other* and perhaps *one another*) are [MODE ana]. This will allow us to reformulate the binding principles in terms of the feature MODE, keeping open the possibility that reflexives and reciprocals might not be the only elements subject to Principle A.

7.3.1 The Argument Structure List

Both of our binding principles contain the phrase 'a preceding argument of the same verb'. Formalizing this in terms of our theory will take a bit more work. The features that encode information about what arguments a verb takes are the valence features SPR and COMPS. Though we have not said much about the linear ordering of arguments, we have placed elements on our COMPS lists in the order in which they appear in the sentence. Hence, to the extent that precedence information is encoded in our feature structures, it is encoded in the valence features. So the valence features are a natural place to start trying to formalize the binding principles.

There is a problem, however. For examples like (2)–(5), the binding in question involves the subject NP and one of the nonsubject NPs; but our valence features separate the subject (specifier) and the nonsubject (complements) into two different lists. To facilitate talking about all of the arguments of a verb together, we will posit a new list-valued feature, ARGUMENT-STRUCTURE (ARG-ST), consisting of the sum (in the sense introduced in Chapter 5) of the SPR value (the subject) and the COMPS value (the complements)[3].

Words obey the following generalization, where '$\oplus$' again denotes the operation we have called 'sum', appending one list onto another:[4]

[2] Note that the Semantic Inheritance Principle guarantees that NPs headed by [MODE ref] nouns share that specification.

[3] MOD, which we have included among the valence features, does not list arguments of the verb. So the value of MOD is not related to ARG-ST.

[4] We will revisit and revise the Argument Realization Principle in Chapter 14.

(12) Argument Realization Principle (Version I)
A word's value for ARG-ST is $\boxed{A} \oplus \boxed{B}$, where $\boxed{A}$ is its value for SPR and $\boxed{B}$ is its value for COMPS.

So, if a verb is specified as [SPR ⟨ NP ⟩] and [COMPS ⟨ NP ⟩], then the verb's argument structure list is ⟨ NP , NP ⟩. And if some other verb is specified as [SPR ⟨ NP ⟩] and [COMPS ⟨ PP , VP ⟩], then that verb's argument structure list is ⟨ NP , PP , VP ⟩, and so on.

Exercise 1: Practice ARG-ST lists
What would be the value of ARG-ST in the lexical entries of each of the following verbs: *devour, elapse, put*, and *rely*? As defined, any word with valence features will have an ARG-ST value. So what would the ARG-ST values be for *letter, of, today*, and *Venezuela*?

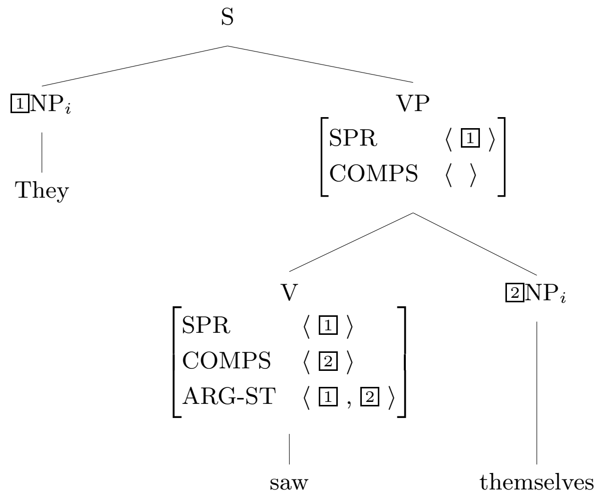
Of course we mean real identity between the members of these lists, as shown by the specifications in (13):

(13) a. $$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle \boxed{2} \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \langle \boxed{1}\text{NP} , \boxed{2}\text{NP} \rangle \end{bmatrix}$$

b. $$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle \boxed{2} , \boxed{3} \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \langle \boxed{1}\text{NP} , \boxed{2}\text{PP} , \boxed{3}\text{VP} \rangle \end{bmatrix}$$

These identities are crucial, as they have the side effect of ensuring that the binding properties of the complements are actually merged into the verb's argument structure, where they will be governed by our binding principles. For example, the Head-Specifier Rule identifies a subject's feature structure with the sole member of the VP's SPR list. It follows (from the Valence Principle) that the subject's feature structure is also the sole member of the verb's SPR list. This, in turn, entails (by the Argument Realization Principle) that the subject's feature structure is the first member of the verb's ARG-ST list. Thus once the distinctions relevant to Binding Theory are encoded in the feature structures of reflexive and nonreflexive NPs, this same information will be present in the ARG-ST of the lexical head of the sentence, where the binding principles can be enforced. This is illustrated in (14):

(14)



The generalization in (12) holds only of words; in fact, it is only word structures that have the feature ARG-ST. Despite its close relationship to the valence features, ARG-ST serves a different function and hence has different formal properties. SPR and COMPS, with the help of the Valence Principle, keep track of elements that a given expression needs to combine with. As successively larger pieces of a tree are constructed, the list values of these features get shorter. By contrast, we introduced the argument structure list as a locus for stating more formal versions of the binding principles. Through a series of identities enforced by the Argument Realization Principle, the phrase structure rules and the Valence Principle, the ARG-ST list of a verb occurring in a tree contains all of the information about that verb's arguments that a precise version of the binding principles needs. It is part of neither SYN nor SEM, but rather serves to express certain relations at the interface of syntax and semantics. These relations can be stated once and for all on the ARG-ST of the lexical head. There is no need to copy the information up to higher levels of the tree, and so ARG-ST is posited only as a feature of *word*s, not *phrase*s.

The elements of an ARG-ST list are ordered, and they correspond to phrases in the phrase structure tree. We can thus use the ordering on the ARG-ST list to impose a ranking on the phrases in the tree. A bit more precisely, we can say:

(15) If *A* precedes *B* on some argument structure (ARG-ST) list, we say that *A* OUTRANKS *B*.

Incorporating both our characterization of reflexive pronouns in terms of MODE and our definition of 'outrank', we can now reformulate our binding principles as follows:

(16) Principle A (Final Version)

A [MODE ana] element must be outranked by a coindexed element.

Principle B (Final Version)

A [MODE ref] element must not be outranked by a coindexed element.

Notice that in this reformulation, Principle B now applies more generally, so as to govern nonpronominal elements like proper names and quantified NPs. This is a happy result, given the following examples, which are now correctly predicted to be ungrammatical:

(17) a.*Sandy$_i$ offended Jason$_i$.
b.*He$_i$ offended Sandy$_i$.
c.*He$_i$ offended each lawyer$_i$.

7.4 Two Problems for Binding Theory

These formulations have certain problems, requiring further discussion and refinement.

7.4.1 Pronominal Agreement

First, (16) says nothing about agreement between pronouns and antecedents; but we do not want Principle A to license examples like (18):

(18) a. *I enjoy yourself.
b. *He enjoys themselves.
c. *She enjoys himself.

We could rule these out by adding a stipulation to Principle A, requiring a reflexive and its antecedent to agree. But this *ad hoc* approach wouldn't explain much. It is intuitively clear why coindexed elements should exhibit a form of agreement: coindexation indicates that the expressions denote the same entity, and the properties indicated by agreement features are characteristically properties of the entity referred to (the expression's DENOTATION). Thus, for example, singular NPs normally denote single entities, whereas plural NPs denote collections. Hence a singular pronoun cannot normally be coindexed with a plural NP, because they cannot have the same denotation.

We will consequently refrain from any mention of agreement in the binding principles. Instead, we adopt the following general constraint:[5]

(19) Anaphoric Agreement Principle (AAP)

Coindexed NPs agree.

By 'agree', we mean 'have the same values for AGR'. Recall that AGR was introduced in Chapter 3 as a feature whose value is a feature structure specifying values for the features PER (person), NUM (number), and (in the case of *3sing* AGR values) GEND (gender). Only PER and NUM matter for the purposes of subject-verb agreement, but pronouns must also agree with their antecedents in gender, as illustrated in (18c). Since GEND is part of AGR, it is covered by the AAP.

One important advantage of leaving agreement out of the formulation of binding principles themselves is that the AAP also covers agreement between nonreflexive pronouns and their antecedents. Since Principle B only says which expressions must NOT be coindexed with nonreflexive pronouns, it says nothing about cases in which such pronouns ARE legally coindexed with something. The AAP rules out examples like (20), which are not ruled out by our formulation of Principle B.

(20) *I$_i$ thought that nobody liked him$_i$.

[5]The use of the term 'anaphoric' in (19) is intended to underscore that coindexing is used to represent the informal notion of anaphora.

It is important to realize that coindexing is not the same thing as coreference; any two coindexed NPs are coreferential, but not all pairs of coreferential NPs are coindexed. There are some tricky cases that might seem to be counterexamples to the AAP, and all of which turn out to be consistent with the AAP, once we make the distinction between coindexing and coreference. One such example is the following:

(21) The solution to this problem is rest and relaxation.

Here the singular NP *the solution to this problem* appears to refer to the same thing as the plural NP *rest and relaxation*. And indeed we would say that the two NPs are coreferential, but they are not coindexed. Thus while coindexing and coreference usually go hand in hand, they don't in this case. The whole point of identity sentences of this kind is to convey the information that two distinct (i.e. distinctly indexed) expressions refer to the same thing. If you are familiar with mathematical logic, this might remind you of situations in which two distinct variables are assigned the same value (making, e.g. '$x = y$' true). Indices are like variables; thus Binding Theory constrains variable identity, not the assignments of values to variables.

Other examples that appear to violate the AAP turn out to be cases where the pronoun isn't even coreferential with its apparent antecedent. Rather, the phrase that the pronoun is 'referring back to' only indirectly introduces the referent of the pronoun into the domain of discourse. For example, consider the sentence in (22):

(22) An interesting couple walked in. He was four foot nine; she was six foot two.

Here, the NP *an interesting couple* refers to the two people denoted by *he* and *she*, but these three expressions all have distinct indices. This is consistent with the AAP. In fact, the referent of the NP *an interesting couple* is just one entity – the couple, which is a collection of two individuals. As the collection is introduced into the discourse, however, it also makes salient each individual that is in the collection, and it is these individuals that the pronouns in the next sentence refer to. Thus in this discourse, the NP *an interesting couple*, the pronoun *he* and the pronoun *she* all refer to different things. So the AAP doesn't apply.

Similar examples involve collective nouns like *family*, which can denote a single entity, as shown by the singular verb agreement in (23), but which can, as a 'side effect', introduce a collection of entities that can serve as the antecedent for a subsequent plural pronoun:

(23) My family hates cornflakes. But they love granola.

Again there are two distinct entities being referred to by distinct indices.[6]

7.4.2 Binding in Prepositional Phrases

A second problem with our formulation of the binding principles is that reflexives and their antecedents can be objects of prepositions. A PP that consists of a prepositional head daughter like *to* or *about* and a reflexive NP object can then become a complement

[6] For some speakers, this is even possible in the context of reflexive pronouns, i.e. in examples like (i):

(i) Pat's family is enjoying themselves.

The theory we develop does not allow examples of this sort.

of the verb; and when this happens, the reflexive NP inside the PP enters into binding relations with the other arguments of the verb. Similarly, when a nonreflexive pronoun functions as a prepositional object, it can behave like an argument of the verb for purposes of binding. Thus we find the pattern of binding illustrated in (24) and (25):

(24) a. They$_i$ talk [to themselves$_i$].
b.*They$_i$ talk [to them$_i$].

(25) a. Nobody told Susan$_i$ [about herself$_i$].
b.*Nobody told Susan$_i$ [about her$_i$].

And in similar examples, the prepositional object can serve as the binder of a reflexive, but not of a nonreflexive:

(26) a. Nobody talked [to Susan$_i$] [about herself$_i$].
b.*Nobody talked [to Susan$_i$] [about her$_i$].

In examples like these, the binding principles, as formulated above, make the wrong predictions: the Argument Realization Principle (henceforth ARP) requires that the verb's ARG-ST contain the feature structure of the PP, not that of the NP within the PP. Hence if a reflexive pronoun is inside a PP that is a complement to a verb, the reflexive's feature structure will not appear on the same ARG-ST list as (the feature structures of) the verb's subject and object NPs. The Binding Theory, as formulated, thus fails to take into account the fact that certain prepositions seem to be transparent for binding purposes. That is, if prepositions such as these were simply not there and the prepositional object were an object of the verb, then Binding Theory would make just the right predictions about (24)–(26) and related examples.

This problem raises both empirical and formal questions. The empirical question is the issue of precisely when objects of prepositions can enter into binding relations with elements outside the PP. As we noted in our initial discussion of Binding Theory in Chapter 1, there is some variability about the binding possibilities of objects of prepositions. This is illustrated in (27):[7]

(27) a. The house$_i$ had a fence around $\left\{\begin{array}{l}\text{it}_i\\ \text{*itself}_i\end{array}\right\}$.

b. To make a noose, you wind the rope$_i$ around $\left\{\begin{array}{l}\text{itself}_i\\ \text{*it}_i\end{array}\right\}$.

c. Susan$_i$ wrapped the blanket around $\left\{\begin{array}{l}\text{her}_i\\ \text{herself}_i\end{array}\right\}$.

[7]Some readers may have a strong preference for one version of (27c) over the other. It appears that there is some cross-speaker variation regarding such examples. For readers who do not accept both versions of (27c), here are some additional examples in which many speakers accept both reflexive and nonreflexive pronouns:

(i) Jane$_i$ put the TV remote down beside $\left\{\begin{array}{l}\text{her}_i\\ \text{herself}_i\end{array}\right\}$.

(ii) Mary$_i$ took a quick look behind $\left\{\begin{array}{l}\text{her}_i\\ \text{herself}_i\end{array}\right\}$.

These examples also show that it is not simply the choice of preposition that determines whether a prepositional object can be reflexive, but also the particular verb that the preposition combines with.

One possible explanation of such differences is based on the intuitive idea underlying our Binding Theory: that reflexives and their antecedents are always arguments of the same predicate. It seems plausible to claim that English prepositions have two distinct semantic functions. In some uses, they function much like verbs, introducing new predications in which they assign argument roles to the nouns they combine with. In other uses, they are simply functioning as argument markers – that is, they indicate what role their object plays in the situation denoted by the verb of the clause they appear in. The clearest examples of this argument-marking use of prepositions are sentences like (4a), *Susan*$_i$ *told a story to herself*$_i$, in which *to* is used to mark what traditional grammarians called the indirect object. In these cases, the preposition can actually be omitted if the order of the complements is reversed: *Susan told herself a story.*

In (27a), the preposition arguably functions as a separate predicate (making the sentence mean roughly, 'The house had a fence, and the fence was around the house'), whereas in (27b), the preposition simply marks one of the arguments of the verb *wind*. Notice that nothing in the meaning of the verb *had* leads one to expect that anything is or goes around its subject. In contrast, the verb *wind* indicates that something is going around something else, so the preposition is introducing an expected participant in the situation. These remarks are intended to provide intuitive motivation for the formal distinction we make between the two types of prepositions, but the real reason we need the distinction is to account for the distribution of reflexive and nonreflexive pronouns. Cases like (27c), then, will be treated as having prepositions that are ambiguous between being independent predicates and argument markers.[8]

Let us now formalize this intuition. For the purposes of Binding Theory, nothing new needs to be said about the prepositions that function as independent predicates. If the object of such a preposition is [MODE ana], then Principle A will require it to be coindexed with something that outranks it on the preposition's ARG-ST list. This is not the case in (27a).[9] If the prepositional object is [MODE ref], it must not be coindexed with anything that outranks it on the preposition's ARG-ST list. Since the subject of the sentence in (27a) does not appear on the ARG-ST list of *around*, Principle B permits a nonreflexive pronoun *it* coindexed with *the house* to appear as the object of *around*.

For prepositions that function as argument markers, however, we need to provide some way by which they can transmit information about their object NP up to the PP that they project. In particular, in order for the binding principles to make the right predictions with respect to objects of argument-marking prepositions, we need to be able

[8]This leads in certain cases to prepositions like *around* being unintuitively treated as not directly contributing to the semantics of the sentence. A full analysis of these facts is beyond the scope of this book.

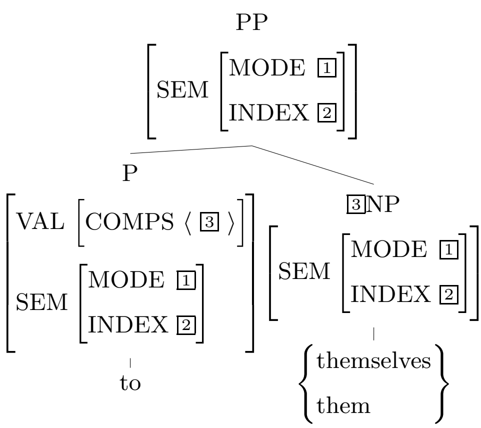
[9]We leave open for now the question of how many ARG-ST members such predicational prepositions have. If *around* in (27a) has two arguments (as seems intuitive from its relational meaning), then the first argument should be identified with *a fence*; hence, *itself* could still not be coindexed with *the house*. In Chapter 12, we will investigate mechanisms by which different ARG-ST lists can have elements with the same index.

to determine at the level of the PP both whether the object NP is a reflexive pronoun (that is, whether it is [MODE ana]) and also what its INDEX value is. If the object's MODE and INDEX values can be transmitted up to the PP, then the higher verb that takes the PP as its complement will have the MODE and INDEX information from the object NP in its ARG-ST, within the PP's SEM value. Note that without some method for transmitting this information up to the PP, the information about the preposition's object is invisible to the higher verb selecting the PP as its complement. The COMPS list of the PP, for example, is empty.

The method we use to transmit this information is straightforward: argument-marking prepositions, such as (some uses of) *to*, *about*, and *of*, share the MODE and INDEX values of their objects. This is illustrated in the lexical entry in (28):

(28)
$$\left\langle \text{to}\ , \begin{bmatrix} \text{SYN} & \begin{bmatrix}\text{HEAD} & \textit{prep}\\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \rangle\end{bmatrix}\end{bmatrix} \\ \text{ARG-ST} & \left\langle \overset{\text{NP}}{\begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{CASE} & \text{acc}\end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{MODE} & \boxed{1}\\ \text{INDEX} & \boxed{2}\end{bmatrix}\end{bmatrix}} \right\rangle \\ \text{SEM} & \begin{bmatrix}\text{MODE} & \boxed{1}\\ \text{INDEX} & \boxed{2}\end{bmatrix}\end{bmatrix}\right\rangle$$

The MODE and INDEX values are projected up from the preposition to the PP by the Semantic Inheritance Principle, as shown in (29):

(29)


A PP like this can be selected by a verb like *tell* or *wind*. Hence, the PP on its ARG-ST list will contain the object NP's MODE and INDEX values within it. Put another way, the information about the object of the preposition that we need in order to apply the binding principles is available in the verb's ARG-ST list.

To get the right binding results for the objects of argument-marking prepositions, we now need to make a slight modification to our definition of 'outranks'. In particular, we need to say that an argument-marking PP and its object NP are 'of equal rank', by which we mean that they outrank exactly the same elements and are outranked by exactly the same elements. More precisely:

(30) (i) If a node is coindexed with its daughter, their feature structures are of equal rank.
(ii) If there is an ARG-ST list on which A precedes B, then A has a higher rank than (i.e. outranks) B.

Part (ii) of this definition is just the definition we gave earlier. Part (i) is needed to account for the binding facts in argument-marking PPs. Consider, for example, the case where the object of such a PP is a reflexive pronoun (e.g. *The children fended for themselves*). The reflexive's INDEX is shared by the preposition *for*, as is the [MODE ana] specification, as required by the lexical entry for the argument-marking *for*. These values are also shared by the whole PP, *for themselves*, as required by the Semantic Inheritance Princple. So the PP and the reflexive pronoun it contains are coindexed; hence, by part (i) of the definition above, the PP and the reflexive pronoun are of the same rank. In the ARG-ST of *fended*, the feature structure of *the children* outranks that of *for themselves*. Consequently, the feature structure of *the children* outranks that of *themselves*. Thus, if *the children* and *themselves* are coindexed, Principle A of the Binding Theory is satisfied. Without part (i) of the definition, the reflexive pronoun would not satisfy Principle A.[10] We will go through a similar example, as well as one with a nonreflexive pronoun, below.

The formal machinery we have just developed is designed to capture the fact that objects of prepositions in English exhibit different binding properties in different environments. It involves positing two kinds of lexical entries for prepositions: one contributes its own MODE and INDEX values; the other adopts those of its object, thereby serving as a conduit for that information to be passed on to the dominating PP. We attempted to motivate this distinction through an intuition that the two kinds of prepositions serve different semantic functions. But such intuitions vary considerably from speaker to speaker, so it would be dangerous to put too much weight on them. Our analysis provides a more reliable means of classifying prepositions as argument marking or predicational, namely, exploring their binding properties. Prepositions that are transparent for purposes of binding should be analyzed as argument markers; those whose objects cannot be bound by a preceding NP in the clause should be analyzed as predicational.

7.5 Examples

So far, this chapter has motivated several technical innovations in our theory (ARG-ST, the concept of 'outranking', and the distinction between the two types of prepositions). In this subsection, we present two examples to illustrate the formal machinery we have been discussing.

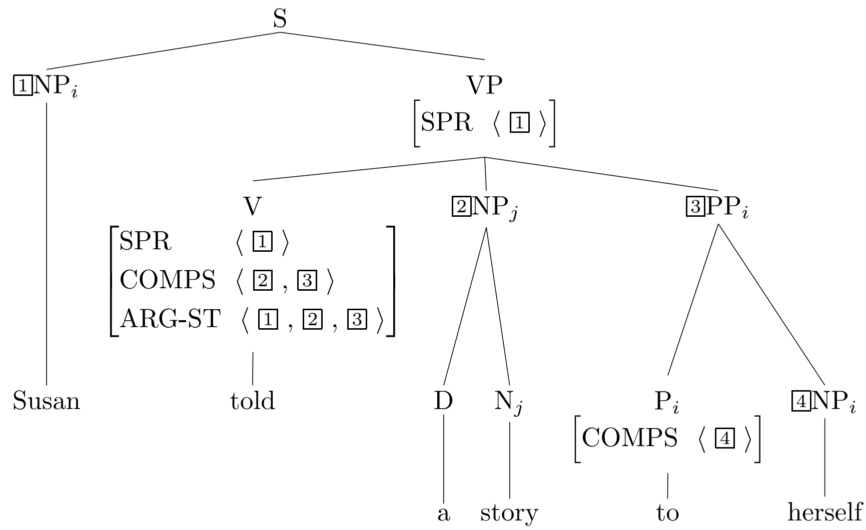
[10] As a consequence of the way we've formalized our analysis, the P *for* is also [MODE ana] and therefore subject to Principle A. It satisfies Principle A in the same way the object NP does: by part (i) of (30), its rank is equal to that of the PP and thus it is outranked by *the children*.

Consider first (4a), repeated here for convenience as (31):

(31) Susan$_i$ told a story to herself$_i$.

The structure licensed by our grammar is the following (omitting irrelevant details):

(32)



The geometry of this tree is given by our phrase structure rules in ways that are by now familiar. The aspect of the tree we are concerned with here is the coindexing of the nodes, indicated by the subscripted *i* and the resulting argument structure of the verb *told*, which is displayed in (33):

(33)
$$\begin{bmatrix}\text{ARG-ST} & \left\langle \overset{\text{NP}_i}{\begin{bmatrix}\text{MODE ref}\end{bmatrix}}, \overset{\text{NP}_j}{\begin{bmatrix}\text{MODE ref}\end{bmatrix}}, \overset{\text{PP}_i}{\begin{bmatrix}\text{MODE ana}\end{bmatrix}} \right\rangle\end{bmatrix}$$

This ARG-ST conforms to the Binding Theory: the [MODE ana] PP is outranked by a coindexed NP, namely the first NP on the list. Similarly, the NP tagged $\boxed{4}$ in (32), which is also [MODE ana], is of equal rank with the PP dominating it (by the definition of rank), so it is outranked by the first NP in the list. Again, Principle A is satisfied. Notice that Principle A requires coindexing between the prepositional object and one of the other arguments, in this case, the subject. The ARG-ST list of *told* plays a crucial role in enforcing this coindexing, even though the verb is one level below the subject and one level above the prepositional object in the tree.

Principle A would also be satisfied if the anaphor were coindexed with the direct object NP:

(34)
$$\begin{bmatrix}\text{ARG-ST} & \left\langle \overset{\text{NP}_j}{\begin{bmatrix}\text{MODE ref}\end{bmatrix}}, \overset{\text{NP}_i}{\begin{bmatrix}\text{MODE ref}\end{bmatrix}}, \overset{\text{PP}_i}{\begin{bmatrix}\text{MODE ana}\end{bmatrix}} \right\rangle\end{bmatrix}$$

Although this is implausible with *told* (because of the nonlinguistic fact that people are not the kind of thing that gets told to others), it is much easier to contextualize grammatically analogous sentences with the verb *compared*:

(35) a. We compared him$_i$ [to himself$_i$] (at an earlier age).
b. We compared them$_i$ [to each other$_i$].

Thus in both (33) and (34), the PP – and hence its NP object as well – is outranked by some coindexed element. It seems correct to say that as far as grammar is concerned, both the ARG-ST configurations in (33) and (34) are acceptable, although there are independent factors of plausibility that interact to diminish the acceptability of many grammatical examples.

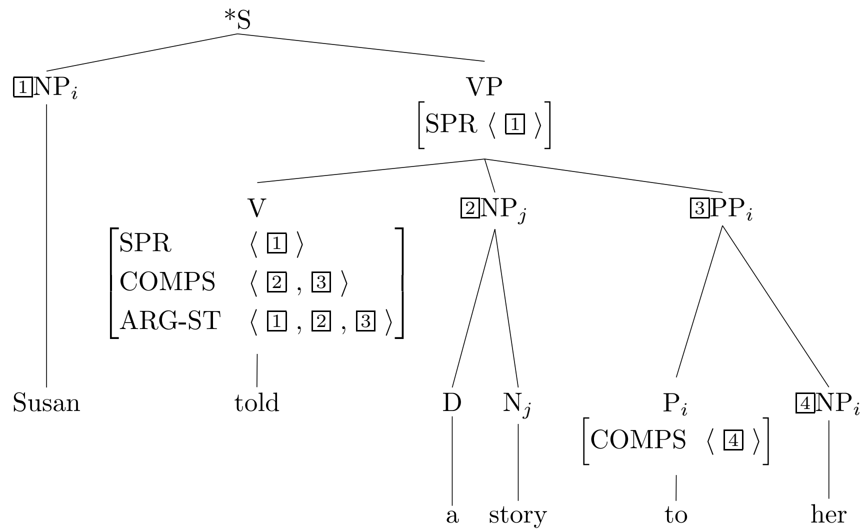
Exercise 2: The Distribution of ARG-ST
Which nodes in (32) have the feature ARG-ST?

Now consider (4b), repeated here for convenience as (36):

(36) *Susan$_i$ told a story to her$_i$.

The tree structure that our grammar must rule out is the following:

(37)



The lexical entry for *her* specifies that it is [MODE ref] – that is, that it is not a reflexive (or reciprocal) pronoun. As in the case of the previous example, the lexical entry for *to* and the Semantic Inheritance Principle pass information to the P and the PP. The verb's ARG-ST list then looks like (38):

(38) $$*\left[\text{ARG-ST} \left\langle \begin{matrix}\text{NP}_i \\ [\text{MODE ref}]\end{matrix}, \begin{matrix}\text{NP}_j \\ [\text{MODE ref}]\end{matrix}, \begin{matrix}\text{PP}_i \\ [\text{MODE ref}]\end{matrix} \right\rangle\right]$$

The PP in (38) violates Principle B: it is a [MODE ref] element that is coindexed with another element that outranks it – namely, the first NP on the list. Consequently, the coindexing indicated is not permitted.

7.6 Imperatives and Binding

In Chapter 1 we noted that the behavior of reflexive and nonreflexive pronouns in sentences like (39) is what one would expect if they had second-person subjects:

(39) a. Protect yourself!

b. $*\text{Protect} \begin{Bmatrix} \text{myself} \\ \text{himself} \end{Bmatrix}!$

c. *Protect you!

d. $\text{Protect} \begin{Bmatrix} \text{me} \\ \text{him} \end{Bmatrix}!$

Sentences like these are known as IMPERATIVE sentences. Their characteristic properties are that they lack an overt subject, employ an uninflected form of the verb, and are used to express directives. Such sentences are sometimes said to have 'understood' second-person subjects. The distribution of reflexives illustrated in (39) shows that imperatives do indeed behave in at least one way as if they had second-person subjects.

Our theory provides a straightforward way of capturing the intuition that imperatives have understood subjects. First we need to allow for verb forms that lack the inflections of the verb forms we have been considering thus far. These forms, produced by a lexical rule discussed in the next chapter, have no inflectional endings and are distinguished from other kinds of verbal forms in terms of differing values for the HEAD feature FORM.[11] This basic form of a verb has the FORM value 'base'.

We introduce a new grammar rule to analyze imperative sentences. This rule allows a sentence to consist of a single daughter: a VP specified as [FORM base]. In requiring that the daughter be so specified, we ensure that the lexical head of that phrase will be an uninflected verbal form, such as *be*, *get*, *run*, or *look*. The new rule we need for imperative sentences is a nonheaded rule that says a sentence may consist of a [FORM base] VP that behaves as though it had a second-person subject and is interpreted as a directive:

(40) Imperative Rule

$$\begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{dir} \\ \text{INDEX} & s \end{bmatrix} \end{bmatrix} \rightarrow \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{base} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \text{NP}\begin{bmatrix} \text{PER} & \text{2nd} \end{bmatrix} \right\rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \end{bmatrix} \end{bmatrix}$$

Recall that imperative sentences require their subject to be second-person, a fact that is captured by the constraint on the SPR of the daughter in (40). And though all verbs are lexically specified as [MODE prop] (which is in turn passed up to the [FORM base] VP that enters into the imperative construction), (40) ensures that any phrase it sanctions is

[11] We will have more to say about the feature FORM in Chapter 8.

specified as [MODE dir] – that is, that it has a meaning appropriate for an imperative.[12]

The Imperative Rule sanctions structures like the one depicted in (41):

(41)

$$
\begin{array}{c}
\text{S}\\
\begin{bmatrix}
\text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \rangle\\ \text{COMPS} & \langle\ \rangle\end{bmatrix}\\
\text{SEM} & \begin{bmatrix}\text{MODE} & \text{dir}\\ \text{INDEX} & s\\ \text{RESTR} & \boxed{1}\end{bmatrix}
\end{bmatrix}\\
|\\
\text{VP}\\
\begin{bmatrix}
\text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \text{NP[PER 2nd]}\ \rangle\\ \text{COMPS} & \langle\ \rangle\end{bmatrix}\\
\text{HEAD} & [\text{FORM base}]\\
\text{SEM} & \begin{bmatrix}\text{MODE} & \text{prop}\\ \text{INDEX} & s\\ \text{RESTR} & \boxed{1}\end{bmatrix}
\end{bmatrix}\\
\triangle\\
\text{get a job}
\end{array}
$$

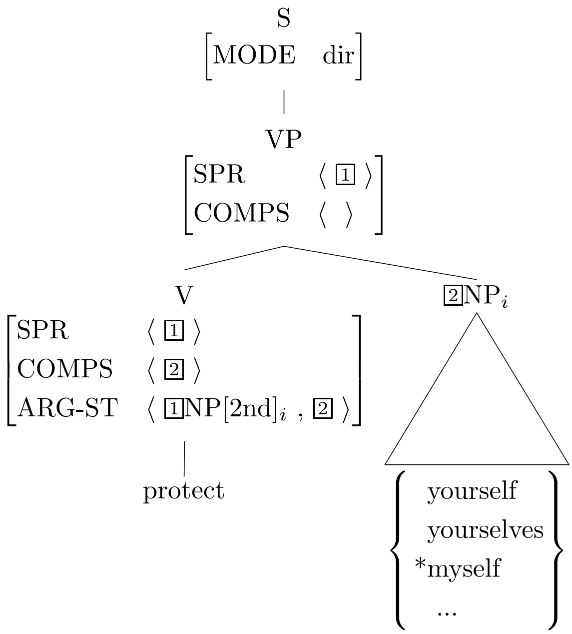
Note that, because the Imperative Rule is a not a headed rule, the Head Feature Principle, the Valence Principle, and the Semantic Inheritance Principle are not relevant to licensing the S node in (41) (though the Semantic Compositionality Principle identifies the RESTR value of the mother in (41) with the RESTR value of the daughter). Instead, the values of the features on the S node are dictated by the rule itself and/or the initial symbol.[13]

The last thing to understand about the rule in (40) is that it explains the observations we have made about anaphor binding in imperative sentences. By requiring the specifier of an imperative VP to be second-person, we constrain the first argument of the VP's lexical head (i.e. the verb) to be second-person as well, thanks to the ARP. This, in turn, entails that in a structure like the following, Principle A will require a reflexive object to be coindexed with (and hence, by the AAP, to agree with) the second person subject:

[12] This analysis of imperatives is incomplete. In a larger grammar, it would need to be scaled up to include a semantic representation for the understood subject, as well as a constraint restricting imperatives to be stand-alone sentences. For more on imperatives and English clauses in general, see Ginzburg and Sag 2000.

[13] There are further constraints on what can be a 'stand alone' clause. In Chapter 9 we will require that the 'initial symbol' of our grammar must include the specification [FORM fin], which will distinguish past and present tense verbs (e.g. *went*, *loves*) from all others. FORM values for verbs are discussed in Chapter 8. Like the specification [COMPS ⟨ ⟩], this information will be supplied to the mother node of imperatives by the initial symbol.

(42)



In this way, our treatment of imperatives interacts with our treatment of ARG-ST so as to provide an account of 'understood' arguments. The ARG-ST may include elements that are not overtly expressed, that is, which correspond to no overt phrase, and these can play a role in binding relations.

Note that we can use Binding Theory to confirm whether or not a given subjectless clause should involve an understood subject. For example, it would be a mistake to analyze exclamations of the form *Damn* NP along the lines just employed for imperatives. If we posited an understood subject NP in the ARG-ST of *damn*, it would license a reflexive pronoun (of the appropriate person, number, and gender) in the position after *damn*. But this is not possible:

(43) $$\text{*Damn} \left\{\begin{array}{l}\text{myself}\\ \text{yourself}\\ \text{herself}\\ \text{himself}\\ \text{itself}\\ \text{themselves}\end{array}\right\}!$$

Hence, *damn* in this use will have to be analyzed as being truly subjectless, in the sense that it has only one element in its argument structure (and an empty SPR list). Examples like (43) are then ruled out because the reflexive element in the ARG-ST is not outranked by any coindexed element.

We have given a preview here of the analysis of verb forms that will be developed in the next chapter. There we will address the question of how the forms are differentiated formally, and how to manage the proliferation of entries for different forms of the same word.

7.7 The Argument Realization Principle Revisited

ARG-ST lists in general, and the ARP in particular, will play an increasingly important role in the chapters to come. We will place various constraints on the ARG-ST values of particular kinds of words, yet these would be vacuous without the ARP, which relates ARG-ST values to the values of the valence features SPR and COMPS. This connection is central, if the constraints we place on lexical heads are to interact with the elements that heads syntactically combine with. The Binding Theory presented in this chapter illustrates the importance of both ARG-ST and the ARP in our theory. Note that the order of arguments on the ARG-ST list also determines their linear order, given the way our grammar works. That is, subjects precede objects and other arguments, direct objects precede other arguments except the subject, and so forth. The ordering in (44) predicts the linear order that arguments occur in reasonably well:

(44) Subject $\succ$ Direct Object $\succ$ 2nd Object $\succ$ Other Complement

ARG-ST also has other uses that we cannot examine in detail here. Many grammarians have sought to explain various regularities exhibited by subjects, objects, and other syntactic dependents of the verb by making reference to the hierarchy in (44). For example, attempts to account for regularities about the semantic roles assigned to syntactic arguments (e.g. a more 'agent-like' argument of a verb will be linked to its subject argument) have led linguists to assume an ordering of the verb's arguments like the ARG-ST ordering. Such theories (which we regrettably cannot do justice to here) are often called LINKING THEORIES.

Various other phenomena have moved linguists to posit an ARG-ST hierarchy. One has to do with what is called 'relativization' i.e. using a clause to modify a noun. In these relative clauses, there is usually a 'gap' – that is, a missing NP that is understood as coreferential with the NP containing the relative clause. For example, in the following sentences, the bracketed portion is the relative clause, and the underlining indicates the location of the gap[14]:

(45) a. I met the person [who __ left].

b. I met the person [who they visited __].

It turns out that there are languages where only subjects can be 'relativized', i.e. where the analog of (45a) is grammatical, but the analog of (45b) is not: But there are apparently no human languages where the facts are the other way around, i.e. where (45b) is grammatical, but (45a) is not. These observations also extend to examples like (46):

(46) I met the person [to whom they handed a present __].

If a language allows (46), it will also allow both (45a) and (45b). The cross-linguistic generalization then is:

(47) If a language can relativize X, then it can relativize any element that outranks X.

In addition, there are languages where a verb agrees not only with its subject, but also with its direct object or with some other argument. An examination of the agreement systems of many of the world's languages, however, will reveal the following generalization to be true:

[14]We return to the analysis of such gaps in Chapter 14.

(48) If a language has words that show agreement with X, then it also has words that show agreement with the elements that outrank X.

Thus the ARG-ST hierarchy appears to have considerable motivation beyond the binding facts that we have used it to explain, some of it cross-linguistic in nature.

The ARP is simply a constraint on the type *word* and may be formulated as follows:

(49)
$$word : \left[\begin{array}{ll} \text{SYN} & \left[\text{VAL} \left[\begin{array}{ll} \text{SPR} & \boxed{A} \\ \text{COMPS} & \boxed{B} \end{array}\right]\right] \\ \text{ARG-ST} & \boxed{A} \oplus \boxed{B} \end{array}\right]$$

This constraint interacts with other constraints in our grammar to give appropriate values to SPR and COMPS. For example, suppose we had a lexical entry for *loves* that specified nothing about SPR and COMPS, as in (50):[15]

(50)
$$\left\langle \text{loves}\ , \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{word}} \\ \text{SYN} & [\text{HEAD}\ \ \textit{verb}] \\ \text{ARG-ST} & \left\langle \begin{array}{c} \text{NP}_i \\ [\text{AGR}\ \ \textit{3sing}] \end{array}, \text{NP}_j \right\rangle \\ \text{SEM} & \left[\begin{array}{ll} \text{MODE} & \text{prop} \\ \text{INDEX} & s \\ \text{RESTR} & \left\langle \left[\begin{array}{ll} \text{RELN} & \textbf{love} \\ \text{SIT} & s \\ \text{LOVER} & i \\ \text{LOVED} & j \end{array}\right] \right\rangle \end{array}\right] \end{array}\right] \right\rangle$$

The effect of the ARP is to ensure that any word structure that (50) gives rise to will also satisfy further identity conditions, for example those indicated by the tags in (51):

(51)
$$\left\langle \text{loves}\ , \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{word}} \\ \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \textit{verb} \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle \boxed{2} \rangle \end{array}\right] \end{array}\right] \\ \text{ARG-ST} & \left\langle \begin{array}{c} \boxed{1}\text{NP}_i \\ [\text{AGR}\ \ \textit{3sing}] \end{array}, \boxed{2}\text{NP}_j \right\rangle \\ \text{SEM} & \left[\begin{array}{ll} \text{MODE} & \text{prop} \\ \text{INDEX} & s \\ \text{RESTR} & \left\langle \left[\begin{array}{ll} \text{RELN} & \textbf{love} \\ \text{SIT} & s \\ \text{LOVER} & i \\ \text{LOVED} & j \end{array}\right] \right\rangle \end{array}\right] \end{array}\right] \right\rangle$$

[15] In fact, as explained in the next chapter, lists like (50), consisting of a phonological form *loves* and a feature structure of type *word*, are to be derived by an inflectional rule.

However, given what we have said so far, (51) is not the only way for both of the elements of the argument structure list in (50) to be identified with complements. The ARP would also be satisfied if both $\boxed{1}$ and $\boxed{2}$ appeared on the COMPS list (with the SPR list empty). Similarly, both $\boxed{1}$ and $\boxed{2}$ could appear on the SPR list (with the COMPS list empty). Such possibilities will need to be ruled out. In the next chapter, we introduce a constraint requiring verbs to have exactly one element on their SPR lists. This will ensure that all words and word structures that satisfy (50) will in fact also satisfy (51).

7.8 Summary

This chapter has developed an acount of anaphoric binding – that is, the association of pronouns with antecedents – within our grammatical framework. We motivated two binding principles, one licensing elements like reflexives and reciprocals and the other restricting the possible coindexing of other NPs. Formalizing this led to a number of innovations, including the feature ARG-ST, the Argument Realization Principle, and the relation 'outrank'. We saw that prepositional phrases exhibit different binding patterns, depending on whether the prepositions serve simply as argument markers or introduce their own predications. Finally, we introduced a new grammar rule for imperative sentences.

7.9 Changes to the Grammar

Most of the changes to our grammar in the remainder of the book will be additions, rather than amendments of rules, principles, or other mechanisms we have already introduced. Hence, it would be redundant and somewhat tedious to have a full grammar summary at the end of each chapter. Instead, we end this chapter and most subsequent ones with a summary of what changes to the grammar we have introduced in the chapter. We will provide two more full grammar summaries: one in Chapter 9, and one in Appendix A.

In this chapter, we added a new value of the MODE feature ('ana'). The type constraint on *sem-cat* now looks like this:

$$\textit{sem-cat}: \begin{bmatrix} \text{MODE} & \left\{\text{prop, ques, dir, ref, ana, none}\right\} \\ \text{INDEX} & \textit{index} \\ \text{RESTR} & \textit{list}(\textit{predication}) \end{bmatrix}$$

We also added a feature ARG-ST (appropriate for feature structures of type *word*) and the Argument Realization Principle (a constraint on the type *word*) which constrains the value of ARG-ST. The value of ARG-ST is a (possibly empty) list of *expression*s. The type constraint on *word* now looks like this:

$$\textit{word}: \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \boxed{A} \\ \text{COMPS} & \boxed{B} \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \boxed{A} \oplus \boxed{B} \end{bmatrix}$$

The Binding Theory itself consists of the definition of 'outrank' and two principles:

The definition of ‘outrank’:

(i) If a node is coindexed with its daughter, their feature structures are of equal rank.

(ii) If there is an ARG-ST list on which A precedes B, then A has a higher rank than (i.e. outranks) B.

The principles of the Binding Theory:

Principle A: A [MODE ana] element must be outranked by a coindexed element.

Principle B: A [MODE ref] element must not be outranked by a coindexed element.

To account for the agreement between pronouns and their antecedents, we introduced a further principle:

The Anaphoric Agreement Principle (AAP): Coindexed NPs agree.

We also introduced a distinction between predicational and argument-marking prepositions, and an analysis of argument-marking prepositions by means of lexical entries with the following specifications:

$$\begin{bmatrix} \text{ARG-ST} & \left\langle \overset{\text{NP}}{\begin{bmatrix} \text{SEM} & \begin{bmatrix} \text{MODE} & \boxed{1} \\ \text{INDEX} & \boxed{2} \end{bmatrix} \end{bmatrix}} \right\rangle \\ \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \boxed{1} \\ \text{INDEX} & \boxed{2} \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \end{bmatrix}$$

Finally, we introduced a new grammar rule, the Imperative Rule:

$$\begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{dir} \\ \text{INDEX} & s \end{bmatrix} \end{bmatrix} \rightarrow \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{base} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \text{NP}\begin{bmatrix} \text{PER} & \text{2nd} \end{bmatrix} \right\rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \end{bmatrix} \end{bmatrix}$$

7.10 Further Reading

The binding of anaphors has been the topic of an extensive literature since the late 1960s. A seminal and very readable paper is Lasnik 1976. To our knowledge, the first proposal to treat reflexive binding in terms of a hierarchy of the verb’s arguments was made by Johnson (1977). The Binding Theory of Chomsky (1981) distilled many of the insights of the research of the preceding decade into three principles; this theory was developed further in a number of works within the Government and Binding Theory of grammar.

A detailed account of binding within Lexical Functional Grammar is presented by Dalrymple (1993). The theory of binding presented in this chapter is based on Pollard and Sag 1992, 1994 with terminological revision ('(out)ranking') due to Bresnan (1995). One of the most detailed attempts to date at formulating a linking theory compatible with the approach presented here is by Davis (2001), whose theory of the alignment of semantics and argument structure allows a further streamlining of all our lexical descriptions. The Argument Structure hierarchy (44) is often referred to as the 'Keenan-Comrie' Hierarchy, because of the pioneering work on this topic reported in Keenan and Comrie 1977.

7.11 Problems

⚠ **Problem 1: Classifying Prepositions**

We have divided prepositions into two sorts: those functioning as predicates and those functioning as argument-markers. For each of the following sentences,

(a) classify the italicized preposition into one of these two sorts (or as being ambiguously both); and

(b) justify your classification by showing (with acceptable and/or unacceptable sentences) what reflexive and nonreflexive coreferential pronouns can or cannot appear as the preposition's object.

(i) The dealer dealt an ace *to* Bo.

(ii) The chemist held the sample away *from* the flame.

(iii) Alex kept a loaded gun *beside* the bed.

(iv) We bought flowers *for* you.

(v) The car has a scratch *on* the fender.

Problem 2: Imperative 'Subjects'

There are imperative sentences that contain an NP that looks like it is the subject of the [FORM base] VP:

(i) You get out of here!

(ii) Everybody take out a sheet of paper!

But the initial NPs in these examples don't seem to participate in the normal agreement pattern with respect to reflexive pronouns. For example, we know that an NP like *everybody* is third person because of its behavior in (iii):[16]

(iii) $$\text{Everybody found} \left\{ \begin{array}{l} \text{?himself} \\ \text{*yourself} \\ \text{?themselves} \\ \text{*myself} \end{array} \right\} \text{a seat.}$$

[16]Following standard practice of generative grammarians, we use designations '?', '??', and '?*' to indicate different levels of naturalness between full acceptability and complete unacceptability.

Yet in imperative sentences, we still find the second-person reflexive pattern illustrated in (iv):

(iv) Everybody find $\left\{\begin{array}{c}\text{??himself}\\ \text{yourself}\\ \text{??themselves}\\ \text{*myself}\end{array}\right\}$ a seat!

Assuming that we do not want to license examples marked '??', what minimal modification of the Imperative Rule would account for the indicated data? Make sure that your proposal still accounts for all relevant facts illustrated above for imperative sentences with no initial NP. For the purposes of this problem, don't worry about the semantics: concentrate on providing a syntactic analysis that will get the binding facts right.

Problem 3: Principle A Revisited

Picking up on an idea from Problem 2 of Chapter 1, we hinted at a couple of places in this chapter that the English reciprocal form *each other* might be [MODE ana] – that is, that it might obey Principle A of the Binding Theory. One immediate obstacle to this suggestion is raised by examples like (i):

(i) They acknowledged each other's contributions.

A. Explain why our current formulation of Principle A together with the assumption that *each other* is [MODE ana] makes the wrong prediction about (i).

At first glance, (i) might be taken to show that reciprocals are not subject to Principle A, but another possibility is that Principle A isn't formulated quite right. It turns out that there are also cases involving reflexives that do not obey Principle A:

(ii) Clinton is writing a book about himself.

(iii) We heard that embarrassing pictures of ourselves had been posted on the internet.

(iv) Pat asked Chris where they had filed the descriptions of themselves.

(v) Pat told Chris to send reminders about the meeting to everyone on the distribution list, with the exception of themselves.

Such data suggest that our formulation of Principle A is in need of revision. We could try to expand the coverage of Principle A, so that it covers such examples. But that approach does not look very promising, particularly for examples (iv) and (v). In those sentences, there is no single NP that serves as the antecedent of the reflexive. Rather, the reflexives in those examples refer to a set consisting of Pat and Chris. This indicates that determining the reference of the reflexive pronouns in these cases is not purely a matter of grammar, but involves some pragmatic inference. Consequently, it seems that the best way to deal with these counterexamples to our current Principle A is to restrict its applicability – that is, to make examples like (ii)–(v) EXEMPT from Principle A.

In doing so, however, we must be careful not to exempt too many anaphors. For example, we want Principle A to continue to account for the distinction in well-formedness between (vi) and (vii):

(vi) They read Mary's story about herself.

(vii) *They read Mary's story about themselves.

B. Reformulate Principle A so that it does not rule out (ii)–(vi), but does rule out (vii). Your formulation should likewise not rule out (i) on the assumption that *each other* is [MODE ana]. [*Hint: Look at what kinds of elements (if any) outrank the* [MODE ana] *elements in (i)–(v), and restrict the applicability of Principle A to cases that have suitable potential antecedents. Note that the objective is simply to remove examples like (i)–(v) from the coverage of Principle A; we are assuming that the generalization that determines how such 'exempt' reflexives and reciprocals are interpreted is outside the domain of grammar.*]

If Principle A is reformulated so as not to block (i), then it will also fail to block examples like (viii).

(viii) *You acknowledged yourself's contribution.

Let us assume the analysis of the English possessive introduced in Chapter 6, Problem 4 – that is, that *'s* is a determiner that takes an obligatory NP specifier. Notice that not all kinds of NPs can serve as specifiers for *'s*; in particular, the forms **I's, *me's, *you's, *he's, *him's, *she's, *her's, *we's, *us's, *they's*, and **them's* are all ill-formed possessive determiner phrases.

C. Formulate a generalization about the possible specifiers of *'s* that will rule out (viii), independent of any facts about binding. How would this be stated formally? [*Hint: You will need to posit a new feature (call it '*PRO*') that distinguishes the kinds of NPs that cannot be specifiers of 's and those that can. The formal statement will involve the* SPR *value of 's.*]

Your reformulation of Principle A probably also exempted examples like (ix) and (x) from its domain. (If it didn't, you should double-check to make sure that its predictions are consistent with (i)–(viii); if so, then you may have discovered a new analysis).

(ix) *Himself is to blame.

(x) *They believe that themselves will win.

D. Suggest a generalization about reflexive pronouns that will rule out (ix) and (x) (again, without relying on binding). [*Hint: Notice that the forms are himself and themselves, not *heself or *theyself.*] How would this generalization be stated formally?

Finally, the reformulation of Principle A to exempt reflexives like those in (ii)–(v) creates problems for the analysis we gave of predicational prepositions. In particular, Principle A will no longer rule out examples like (xi) (repeated from (27a)):

(xi) *The house had a fence around itself.

E. Explain why the reflexive in (xi) is no longer ruled out.

Later in the book, we will introduce formal machinery that will allow us to bring examples like (xi) back within the purview of Principle A.

8

The Structure of the Lexicon

8.1 Introduction

In the course of the last few chapters, we have put more and more of the descriptive burden of our theory into the lexicon. Lexical entries have evolved from simple pairings of phonological forms with grammatical categories into elaborate information structures, in which phonological forms are now paired with more articulated feature structure descriptions. This has permitted us to reduce our inventory of grammar rules to a few very general schemas, to account for a range of syntactic phenomena, and to relate our syntactic representations to semantic ones.

Since our theory relies heavily on rich lexical representations, we need to consider what kind of internal organization the lexicon should have. In particular, we do not want to claim that all information contained in lexical entries is simply listed. A great number of the constraints that we are now putting into lexical entries are not idiosyncratic to individual words. Rather, they reflect general properties of classes of words, e.g. common nouns, proper nouns, verbs, tensed verbs, and so forth. Stipulating all of these constraints redundantly on each individual lexical entry would miss all the significant generalizations about how words and lexical constraints are organized. For example, we handle subject-verb agreement by having the AGR value of a verb be the same as the AGR value of its specifier. We guarantee that this identity holds by imposing the SHAC on a lexical class that includes verbs. Most verbs have two lexical entries that are present tense, one whose AGR value is of type *3sing* and another whose AGR value is *non-3sing*. Aside from the difference in their AGR value (and hence of their specifiers' AGR values), these two entries for each verb are essentially identical: their part of speech is *verb*; they have the same COMPS value; and their semantics includes the same predication. This is no accident, nor is the fact that the same suffix (namely, *-s*) is used to mark almost all third-person singular present tense verb forms.

Notice, by the way, that capturing such generalizations is motivated not only by general considerations of parsimony, but also on psycholinguistic grounds. On encountering a novel English verb (say, a recent coinage such as *email* or an obscure word like *cark*), any competent speaker will add the suffix *-s* when using it in the present tense with a third-person singular subject. In short, speakers know that there are systematic (or, as linguists say, 'productive') relationships among different forms of the same word, and our grammar should reflect this systematicity. The focus of the present chapter is to develop mechanisms for expressing regularities within the lexicon.

8.2 Lexemes

Before we begin developing our lexical theory, however, we want to call attention to what, in everyday English, are two different uses of the term 'word'. In some contexts, people informally distinguish, for example, *runs* and *ran* as two different words: they are pronounced differently, have (subtly) different meanings, and have slightly different co-occurrence restrictions. But in other contexts, the same people would have no hesitation in referring to *runs* and *ran* as two forms of the word *run*. Clearly, these are two different conceptions of 'word': the first refers to a certain pairing of sound and meaning, whereas the latter refers to a family of such pairings. In a formal theory of grammar, these two concepts must not be conflated. Our type *word* corresponds to the first usage (in which *runs* and *ran* are distinct words). The feature structures labeling the preterminal nodes of our trees must all be of type *word*.

But we also want to capture what people have in mind when they use 'word' in the second sense. That is, we want to be able to express the relationship between *runs* and *ran* (and *run* and *running*). We do this by means of a new type *lexeme*. A *lexeme* can be thought of as an abstract proto-word, which, by means to be discussed in this chapter, gives rise to genuine words (that is, instances of the type *word*).

Note that in any language with a rich system of morphological inflection, the need for the notion of 'lexeme' would be apparent. In Spanish, for example, we find PARADIGMS of related words like the following:

(1)

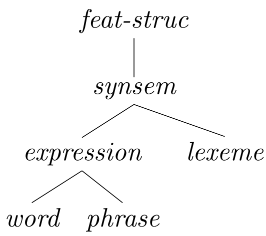
vivo	'I live'	vives	'you(sg.)live'	vive	'(s)he/it lives'
vivimos	'we live'	vivís	'you(pl.) live	viven	'they live'
vivía	'I lived'	vivías	'you(sg.)lived'	vivía	'(s)he/it lived'
vivíamos	'we lived'	vivíais	'you(pl.) lived	vivían	'they lived'
viviré	'I'll live'	vivirás	'you(sg.)'ll live'	vivirá	'(s)he/it'll live'
viviremos	'we'll live'	viviréis	'you(pl.)'ll live	vivirán	'they'll live'

Clearly we need some way of talking about what these forms all have in common. We will say that they are distinct words associated with – or derived from – a common lexeme. Each such lexeme contributes a unique constellation of information – partly phonological (the stem from which all these inflected forms are derived), partly syntactic (including, among other things, the information that this is a verbal lexeme), partly semantic (the meaning that distinguishes this from other verbal lexemes). The reason why it isn't so obvious that we need a notion like lexeme in English is simply that English (for historical reasons) has very little inflectional morphology. Nonetheless, we'll be happy to have a way of analyzing a family of forms like the following, all of which are realizations of a common lexeme:

(2) do, does, did, don't, doesn't, didn't, doing

We incorporate the notion of lexeme into our theory by first revising a high-level distinction in our type hierarchy – the types that distinguish among the syntactic-semantic complexes we have been referring to as expressions, words, and phrases. We will refer to the most general such type of feature structure simply as *synsem* (indicating that it is a complex of syntactic and semantic information). The type *expression* will then be an immediate subtype of *synsem*, as will the new type *lexeme*. And, as before, *word* and *phrase* are the two immediate subtypes of *expression*. This reorganization of the type hierarchy is summarized in (3):

(3)



The feature ARG-ST is defined for both *lexeme* and *word*, so both lexemes and words have argument structure.[1]

Up to now, we have simply stated most lexical constraints in individual lexical entries. For example, whatever generalizations hold for all common nouns have been stipulated redundantly in each common noun's lexical entry. The same is true for the lexical entries we have posited for verbs. But there are many regularities that hold over classes of lexemes – common noun, proper noun, intransitive verb, transitive verb, and so forth. We will now modify our grammar in order to be able to express these generalizations.

Just as we have used a type hierarchy to factor out general properties of linguistic objects in terms of type constraints, our grammar will now organize lexemes into subtypes of the type *lexeme*, in order to provide a home for generalizations about word classes. We'll deal with regularities governing inflectional classes (third-singular present tense verbs, plural nouns, etc.) in terms of LEXICAL RULES, a new construct we introduce and explain in Sections 8.6–8.8 below.

8.3 Default Constraint Inheritance

In previous chapters, we introduced the idea that some types are subtypes of others, with the following effect:

(4) If T_2 is a subtype of T_1, then
  a. every feature specified as appropriate for T_1 is also appropriate for T_2, and
  b. every constraint associated with T_1 affects all instances of T_2.

Formulated in this way, the inheritance of constraints in our type hierarchy is MONOTONIC: constraints on supertypes affect all instances of subtypes, without exception. An intuitive alternative to this conception is to allow for DEFEASIBLE constraints – constraints on a given type that hold BY DEFAULT, i.e. unless contradicted by some other constraint that holds at a more specific level. In this alternative picture, contradictory information associated with a subtype takes precedence over (or OVERRIDES) defeasible constraints that would otherwise be inherited from a supertype. Defeasible constraints

[1]Strictly speaking, a grammar cannot declare a feature to be appropriate at two different places in the type hierarchy. Each feature should be declared only once, and inherited by subtypes. Hence, the current hierarchy, where *lexeme* and *word* have no common supertype, is a simplification. In Chapter 16 we solve this problem by recasting the lexicon as a multiple inheritance hierarchy.

and default inheritance allow a type system to express the idea that language embodies generalizations that have exceptions – subclasses with subregularities and individual elements with idiosyncratic properties.

It has long been recognized that the default generalizations we find in natural languages are layered, i.e. that there are default generalizations governing intermediate-level categories of varying grain[2]. This intuitive idea is simple to express: we need to allow a constraint associated with a given lexical type to be marked as defeasible. Suppose a defeasible constraint C_i applies to a lexical type T_i. Then this constraint holds of any lexical entry of type T_i for which it is not explicitly contradicted. It could be overridden in one of two ways. First, a subtype of T_i might have a constraint associated with it that contradicts C_i. That is, there could be a type T_j that is a subtype of T_i and a constraint C_j associated with T_j that is incompatible with C_i:

(5) T_i:C_i

... ...

... T_j:C_j ...

... ...

... T_m ...

In this case, C_j takes precedence and overrides C_i. A second way to override a defeasible constraint involves information stipulated in a particular lexical entry. That is, a constraint on a particular instance of a leaf type T_m (T_m a subtype of T_i) could contradict C_i.[3] In this case, too, the information associated with the lexical entry takes precedence over the defeasible constraint. But that constraint is true of all instances of T_i in which it is not overridden (as of course are all nondefeasible constraints).

Natural languages exhibit a great many regularities with exceptions that can be modeled elegantly in terms of type hierarchies. For example, names in English (often called PROPER NOUNS) don't usually take specifiers. This is illustrated in (6):

(6) a. Cameron skates.

b. $\left\{\begin{array}{l}\text{*A}\\ \text{*The}\end{array}\right\}$ Cameron skates.

Moreover, proper nouns are normally third-person and singular, as (7) shows:

(7) *Cameron skate.

[2]This concept was explicitly recognized by the Indian grammarians in the first millennium B.C.

[3]Recall that a leaf type (also known as a 'maximal' type) is a type that has no subtypes. We're taking a small liberty here in talking of the lexical entry as describing an instance of a leaf type. In our current set-up (but not the one discussed in Chapter 16), our lexical entries in fact describe pairs consisting of a form and a feature structure belonging to a leaf-type. We will sometimes say, informally, that a lexical entry is of some particular type. What we mean by this is that the second element of (the ordered pair that makes up) the lexical entry describes feature structures of that type.

These generalizations will be captured in our type system by introducing a type for proper nouns with defeasible constraints (stated more formally below) specifying that the value of AGR must be of type *3sing* and that the ARG-ST (and hence both SPR and COMPS lists) must be empty. But there are exceptions to these constraints. In particular, there are several proper nouns in English naming mountain ranges that appear only in the plural and only with a determiner:

(8) a. The $\left\{\begin{matrix}\text{Andes}\\\text{Alps}\end{matrix}\right\}$ are magnificent.

b. *The $\left\{\begin{matrix}\text{Ande}\\\text{Alp}\end{matrix}\right\}$ is magnificent.

c. Hannibal crossed the $\left\{\begin{matrix}\text{Alps}\\\text{Andes}\end{matrix}\right\}$.

d. *Hannibal crossed $\left\{\begin{matrix}\text{Alps}\\\text{Andes}\end{matrix}\right\}$.

In fact, names for mountain ranges may be a lexical type in the lexeme hierarchy, providing an example of a lexical subtype whose constraints override two constraints on a superordinate type.

An even clearer example of this phenomenon is names for US sports teams. In every team sport in the United States, it is in general true that the team names are plural and select *the* as their specifier:

(9) a. The (San Francisco) Giants $\left\{\begin{matrix}\text{are}\\\text{*is}\end{matrix}\right\}$ in first place.

b. The (Bay Area) CyberRays $\left\{\begin{matrix}\text{were}\\\text{*was}\end{matrix}\right\}$ in Boston yesterday.

c. The (Oakland) Raiders $\left\{\begin{matrix}\text{play}\\\text{*plays}\end{matrix}\right\}$ in Denver tonight.

d. The (Boston) Celtics $\left\{\begin{matrix}\text{play}\\\text{*plays}\end{matrix}\right\}$ Indiana today.

An alternative hypothesis about the names of mountain ranges and team names is to treat them as 'words with spaces in them', including *the* as part of the proper noun's form. Such an analysis would treat these names as having the same SPR value ($\langle\ \rangle$) as all other proper nouns. The 'words with spaces' analysis is presumably necessary for other names, e.g. *San Francisco*, *Great Britain*, or (*The*) *Leland Stanford Junior University Marching Band*. However, there is evidence that the proper nouns *Andes*, *Oakland Raiders*, or *Boston Celtics* (unlike *San Francisco* and the like) must be entered in the lexicon as nouns that combine with a specifier syntactically because of other regularities having to do with compound nouns.

Compound nouns can be constructed from pairs of nouns:

(10) a. car thief
  b. department chair
  c. community center
  d. Boston lawyer
  e. Oakland mayor

As (10) shows, the first member of the compound can be either a common noun or a proper noun. And these compound nouns, once constructed (by a lexical rule), can combine syntactically with a determiner in the same way that a non-compound common noun does:

(11) a. $\left\{\begin{array}{l}\text{a}\\ \text{the}\end{array}\right\}$ [car thief]
  b. $\left\{\begin{array}{l}\text{a}\\ \text{the}\end{array}\right\}$ [department chair]
  c. $\left\{\begin{array}{l}\text{a}\\ \text{the}\end{array}\right\}$ [community center]
  d. $\left\{\begin{array}{l}\text{a}\\ \text{the}\end{array}\right\}$ [Boston lawyer]
  e. $\left\{\begin{array}{l}\text{an}\\ \text{the}\end{array}\right\}$ [Oakland mayor]

By including *Andes*, *Oakland Raiders*, and *Boston Celtics* in the lexicon as nouns that select for a determiner syntactically (rather than listing *the Andes*, *the Oakland Raiders* and *the Boston Celtics*), we correctly predict their behavior in compound nouns. That is, it is the determinerless elements that form compounds with other nouns:

(12) a. $\left\{\begin{array}{l}\text{an}\\ \text{the}\end{array}\right\}$ [Andes specialist]
  b. $\left\{\begin{array}{l}\text{an}\\ \text{the}\end{array}\right\}$ [[(Oakland) Raiders] spokesperson]
  c. $\left\{\begin{array}{l}\text{a}\\ \text{the}\end{array}\right\}$ [[(Boston) Celtics] player]

If we were to treat names for mountain ranges and sports teams as 'words with spaces in them', we would incorrectly predict that compound nouns like the following would be well-formed:

(13) a.* $\left\{\begin{array}{l}\text{a}\\ \text{the}\end{array}\right\}$ [[the Andes] specialist]
  b.* $\left\{\begin{array}{l}\text{a}\\ \text{the}\end{array}\right\}$ [[the Oakland Raiders] spokesperson]

c.* $\left\{\begin{array}{l}\text{a}\\ \text{the}\end{array}\right\}$ [[the Boston Celtics] manager]

Hence there is independent justification for our claim that these classes of proper noun are exceptional both in being plural and in selecting a specifier.

Note further that there are exceptions to the subregularity of sports team names. Certain US teams have names that are combinations of determiner plus mass noun:

(14) a. The (Miami) Heat
b. The (Philadelphia) Charge
c. The (Stanford) Cardinal[4]

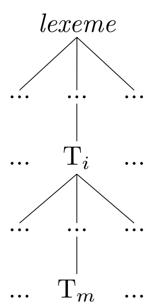
These determiner-selecting nouns have singular uses, as the following examples show (though there appears to be some variation in this):

(15) a. Despite their average age, the Charge boasts an experienced roster.[5]
b. The Cardinal plays Arizona State at 7 p.m Saturday at Stanford.[6]

This is a typical situation: many broad and productive generalizations in languages have exceptions, either idiosyncratic lexical entries or classes of idiosyncratic expressions. For this reason, we shall allow defeasible constraints into our type hierarchy. This will allow us both to restrict the number of types that are required in our grammar and also to keep our constraints simple, without precluding the possibility that some instances or subtypes might be exceptions to the constraints.

By organizing the lexicon as a type hierarchy, together with the use of default constraint inheritance, as described above, we can minimize the stipulations associated with particular lexical entries and express the shared properties of different word classes, at the same time that we allow for idiosyncrasy of the kind we have been discussing. The overall conception of the lexicon is as shown in (16):

(16)



Each of our lexical entries will include a feature structure assigned to some maximal (that is, leaf) type T_m. T_m will in turn have a family of supertypes T_i, that are intermediate between the type *lexeme* and T_m. The various intermediate types correspond to intermediate levels of classification, where type constraints can express linguistic generalizations. Each type in the lexeme hierarchy (which elaborates the hierarchy shown earlier in (3)) has constraints associated with it – some INVIOLABLE, and others that are

[4] This name refers to the color, not the bird.
[5] http://www.wusa.com/charge/, as of September 17, 2001.
[6] *San Jose Mercury News*, September 17, 2001.

defeasible. Since this is a default inheritance hierarchy, we can provide a natural account of the fact that individual lexemes have many properties in common but may differ from one another in terms of particular constraints that override the general constraints governing their supertypes. The idea is that each (basic) lexical entry describes a distinct family of lexemes, each of which is an instance of a maximal type T_m. The members of that family inherit the constraints stipulated in the given lexical entry, the constraints associated with T_m, and those associated with the supertypes of T_m. A lexeme inherits the inviolable constraints and all compatible default constraints. Once a lexical hierarchy (with associated constraints) is put into place, any lexical entry that we write becomes a highly streamlined INITIAL DESCRIPTION (perhaps indicating no more than the phonology and meaning of a given lexeme and which maximal type its satisfiers belong to). All further grammatically relevant constraints (i.e. the rest of the constraints that are part of the FINAL DESCRIPTION that the relevant lexeme instantiation must satisfy) are inherited automatically, according to the method just described.[7]

We use the symbol '/' to indicate that a certain specification is defeasible and hence can be overridden by a conflicting specification.[8] As a simple example of a defeasible constraint, let us go back to the framework for modeling universities we presented in Chapter 3. Suppose we wanted to adapt the system presented there to model New College, a college so small that it relies almost exclusively on a single telephone number. If only the most important individuals had their own telephone number, we might hypothesize a defeasible constraint like the following:

(17) $$\textit{entity} : \begin{bmatrix} \text{TEL} & /\ \text{555-111-1234} \end{bmatrix}$$

Our entry for the New College Music Department (analogous to a lexical entry in our grammar) might then be as shown in (18):

(18) $$\begin{bmatrix} \textit{department} & \\ \text{NAME} & \text{New College Music} \\ \text{FOUNDERS} & \left\langle \begin{bmatrix} \text{NAME} & \text{LaVern Baker} \end{bmatrix}, \begin{bmatrix} \text{NAME} & \text{Clyde McPhatter} \end{bmatrix} \right\rangle \\ \text{CHAIR} & \begin{bmatrix} \text{NAME} & \text{Johnny Otis} \end{bmatrix} \end{bmatrix}$$

Because *department* is a subtype of *entity*, all instances of the type *department* inherit the constraint in (17), unless their entry says otherwise. Thus New College Music has the properties shown in (19), but New College English could have an entry like (20), which overrides (17):

[7] Our defeasible constraints are thus essentially abbreviatory (or 'nonpersistent'). Final lexical descriptions of a lexeme or word contain no defeasible constraints. Hence our hierarchy could be replaced by another (more complicated) one, all of whose constraints are nondefeasible.

[8] The theory of defaults we employ here (as well as the '/' notation) is adapted from Lascarides et al. 1996. See also Lascarides and Copestake 1999 and the further reading section at the end of this chapter.

(19)
$$\begin{bmatrix} \textit{department} & \\ \text{NAME} & \text{New College Music} \\ \text{FOUNDERS} & \left\langle \begin{bmatrix}\text{NAME} & \text{LaVern Baker}\end{bmatrix}, \begin{bmatrix}\text{NAME} & \text{Clyde McPhatter}\end{bmatrix} \right\rangle \\ \text{CHAIR} & \begin{bmatrix}\text{NAME} & \text{Johnny Otis}\end{bmatrix} \\ \text{TEL} & \text{555-111-1234} \end{bmatrix}$$

(20)
$$\begin{bmatrix} \textit{department} & \\ \text{NAME} & \text{New College English} \\ \text{FOUNDERS} & \langle\, \boxed{1} \,\rangle \\ \text{CHAIR} & \boxed{1}\begin{bmatrix}\text{NAME} & \text{Lawrence Ferlinghetti}\end{bmatrix} \\ \text{TEL} & \text{555-111-5555} \end{bmatrix}$$

We will also sometimes want to indicate that two feature values are identical by default. We can also do this using the '/' notation. In Chapter 3, we considered a constraint requiring that a department and its chair have the same telephone number. As we noted there in passing, this constraint is not true of Stanford. But suppose it were the norm, with only occasional exceptions. In that case, we could include in our theory a defeasible version of that constraint, which would be formulated as follows:

(21)
$$\textit{department}: \begin{bmatrix} \text{TEL} & /\,\boxed{1} \\ \text{CHAIR} & \begin{bmatrix}\text{TEL} & /\,\boxed{1}\end{bmatrix} \end{bmatrix}$$

This constraint allows an individual department chair to have a phone number distinct from that of the department (s)he chairs, but will enforce the relevant identity unless there is some specific indication to the contrary. A similar constraint might indicate that the chair of a New College department is its founder, by default. Defeasible identity constraints are a bit tricky, though – we will consider them in more detail in Sections 8.6–8.8 below.

There is one final property of our approach to default constraint inheritance that is important to understand. This has to do with the behavior of complex defeasible constraints. Suppose some type in our grammar T_i requires that the value of the feature MOD be $\langle$ S $\rangle$, by default. Given that 'S' is an abbreviation, this constraint could be formulated more precisely as in (22):

(22)
$$\mathrm{T}_i: \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{MOD} & \left\langle \begin{bmatrix} \text{SYN} & / \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Here the default specification involves three features: HEAD, SPR and COMPS.

Suppose now that T_j, a subtype of T_i, contradicts just part of the constraint in (22), say as in (23):

(23)
$$\mathrm{T}_j : \left[\mathrm{SYN} \left[\mathrm{VAL} \left[\mathrm{MOD} \left\langle \left[\mathrm{SYN} \left[\mathrm{VAL} \left[\mathrm{SPR} \quad \langle\ \mathrm{NP}\ \rangle\ \right]\right]\right] \right\rangle \right]\right]\right]$$

The important thing to understand about the interaction of complex defaults like (22) and constraints like (23) is that the parts of defeasible constraints that are not explicitly contradicted remain in force. That is, the combination of (22) and (23) is the constraint shown in (24), where only the information that is specifically contradicted is overridden:

(24)
$$\mathrm{T}_i \ \&\ \mathrm{T}_j : \left[\mathrm{SYN} \left[\mathrm{VAL} \left[\mathrm{MOD} \left\langle \left[\mathrm{SYN} \begin{bmatrix} \mathrm{HEAD} & /\ \textit{verb} \\ \mathrm{VAL} & \begin{bmatrix} \mathrm{SPR} & \langle\ \mathrm{NP}\ \rangle \\ \mathrm{COMPS} & /\ \langle\ \ \rangle \end{bmatrix} \end{bmatrix} \right] \right\rangle \right]\right]\right]$$

Note that the default part of the constraint has been 'pushed down' to the next level of embedding in such a way as to have the maximum effect that is still consistent with the overriding constraint. Instances of type T_i are thus S-modifiers by default, but instances of the subtype T_j are VP-modifiers.[9]

8.4 Some Lexemes of Our Grammar

The lexical entries, taken together with the constraints inherited via the lexeme hierarchy, characterize the set of basic lexical elements of the language. These are one kind of LEXICAL SEQUENCE, pairs consisting of a phonological form and a feature structure of type *lexeme*.[10] These lexical sequences then give rise to a family of lexical sequences whose second member is a feature structure of type *word*. This is accomplished through the application of inflectional rules. Thus, lexical entries[11] serve as the basis for constructing words and words serve as the building blocks for syntactic structures. In Sections 8.6–8.8 (and much of the remainder of this book) we will discuss a number of lexical rules that play an important role in the grammar of English.[12]

Many of the constraints we present here specify the nature of the ARG-ST lists that are associated with a particular lexeme type, and hence with the lexical entries that are of that type.[13] For example, these constraints specify how many elements are on a given ARG-ST list, what syntactic constraints those elements must obey, and so forth. And words typically have ARG-ST lists that are only minimally different from those of the lexical entries they are derived from, for the simple reason that inflectional rules

[9]The constraints on the HEAD and COMPS values in (24) are defeasible because the constraints on T_j may still be overridden by constraints on one of its subtypes or by constraints on a particular lexical entry.

[10]In Chapter 16, we will show how lexical sequences can be eliminated, once the notion 'sign' is introduced.

[11]Now that we have introduced the term 'lexical sequence', we will reserve the term 'lexical entry' for the pairings of form and linguistic constraints that we list in the lexicon. Lexical entries, like other parts of the grammar, are descriptions. Lexical sequences (both those that satisfy lexical entries and those licensed by lexical rules) are models.

[12]We are assuming that even noninflected words are derived from lexemes. An alternative that we will not pursue here is to enter such words directly into the lexicon with no corresponding lexemes.

[13]More precisely: whose second member is a feature structure of that type.

typically do not add, delete, or rearrange arguments. Thus, the constraints placed on a given lexical entry usually end up having an effect on the words that are derived from it. In particular, because words are subject to the ARP developed in the last chapter, the SPR and COMPS values of a given word is systematically related to its ARG-ST list and hence indirectly to the ARG-ST value of the lexical entry from which that word is derived.[14]

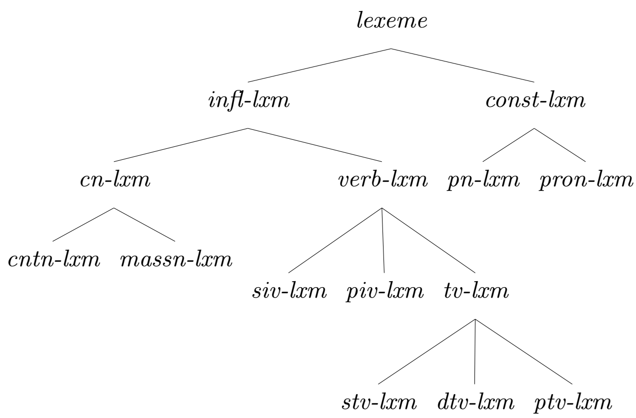
As noted earlier, we are now assuming that *lexeme* and *expression* are the two immediate subtypes of the type *synsem* and that *word* and *phrase* are the two immediate subtypes of *expression.* The type *lexeme* bears the constraints in (25):

(25)
$$\textit{lexeme} : \begin{bmatrix} \text{ARG-ST} & \textit{list}(\textit{expression}) \\ \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{MOD} & / \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

These constraints declare the feature ARG-ST to be appropriate for all lexemes, and make [MOD $\langle\ \rangle$] the default, as most lexemes cannot be modifiers.

Among lexemes, we draw a further distinction between those that give rise to a set of inflected forms and those that do not show any morphological inflection. That is, we posit *inflecting-lexeme* (*infl-lxm*) and *constant-lexeme* (*const-lxm*) as two subtypes of *lexeme.* The type hierarchy we will assume for nominal and verbal lexemes in English is sketched in (26):

(26)



Here, each leaf type corresponds to a lexical class and the various supertypes correspond to larger classes that exhibit regularities that are shared by more than one of the smallest classes. We will explain each of these types in turn.

We begin by commenting briefly on the types at the top of the lexeme hierarchy. Inflecting lexemes are further classified in terms of the subtypes *common-noun-lexeme*

[14] Note that the value of ARG-ST, as before, is a list of feature structures of type *expression.* This now has the important effect of disallowing lexemes as members of ARG-ST lists. Since ARG-ST elements correspond to members of SPR and COMPS lists, and these correspond to the elements selected by the heads of phrases (i.e. to the non-head daughters in our headed phrases), the fact that arguments must be expressions also entails that lexemes cannot appear as specifiers or complements in our syntactic structures. In fact, we want all daughters in syntactic structures to be expressions, rather than lexemes, and will make further modifications in our grammar rules to ensure this.

(*cn-lxm*) and *verb-lexeme* (*verb-lxm*), as these are the only two kinds of English lexeme considered here that give rise to inflected forms. The types *proper-noun-lexeme* (*pn-lxm*) and *pronoun-lexeme* (*pron-lxm*) are two of the subtypes of *const-lxm*. They are discussed more fully in Section 8.4.3 below. This organization has the benefit of providing a natural home for the SHAC. It is now a constraint on the type *infl-lxm*:

(27) Specifier-Head Agreement Constraint (SHAC)

$$\textit{infl-lxm} : \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

The SHAC has two effects: it ensures that elements select for a specifier and that they agree with the specifiers they select. As desired, the SHAC applies only to verbs and to common nouns. Notice that the SHAC is not a defeasible constraint.

8.4.1 Nominal Lexemes

The type *cn-lxm* exhibits numerous regularities that are summarized by the complex constraint in (28):

(28)

$$\textit{cn-lxm} : \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} \\ \text{AGR} & \begin{bmatrix} \text{PER 3rd} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & /\ \text{ref} \\ \text{INDEX} & i \end{bmatrix} \\ \text{ARG-ST} & \langle\ \text{DP}_i\ \rangle \oplus /\langle\ \ \rangle \end{bmatrix}$$

(28) ensures that all common nouns are [HEAD *noun*], that they select determiner phrases (e.g. *the* or *the university's*) as their first argument, and that the rest of their ARG-ST is the empty list, by default.[15] The SHAC (inherited from *infl-lxm*, see (27) above) requires that the SPR list have exactly one element on it.[16] This will mean, once we factor in the effect of the ARP, that their COMPS list is empty, by default. A noun like *picture*, which takes an optional PP complement in examples like (29), provides part of the motivation for making this specification defeasible:

(29) a. [The [picture (of Sandy)]] was awesome.
b. We couldn't find [any [pictures (of Blind Mello Jello)]].

Finally, note that (28) also requires that common nouns be referential ([MODE ref]), by default. This is a defeasible constraint because in Chapter 11 we will encounter some common nouns that are not referential.

The type *cn-lxm* has two subtypes: *count-noun-lxm* (*cntn-lxm*) and *mass-noun-lxm* (*massn-lxm*). These are constrained as shown in (30):

[15] The noun identifies its own INDEX with that of the DP so that the DP can identify that index with its BV value. See Chapter 5, Section 5.8.

[16] The claim that specifiers are obligatory for common nouns appears to be inconsistent with the existence of plural and mass NPs that lack determiners. The analysis of such NPs is the topic of Problem 2 below.

(30) a. *cntn-lxm* : $\left[\text{ARG-ST} \quad \langle\ [\text{COUNT} +]\ , \ldots \rangle\right]$

b. *massn-lxm* : $\left[\text{ARG-ST} \quad \langle\ [\text{COUNT} -]\ , \ldots \rangle\right]$

These type constraints allow the lexical entries for common nouns to be quite streamlined. (31) is a typical lexical entry for a count noun in our grammar:

(31)

$$\left\langle \text{dog}\ ,\ \begin{bmatrix} \textit{cntn-lxm} & \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{dog} \\ \text{INST} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Here, as before, the lexical entry's second member is a feature structure description.

What objects satisfy an entry like (31)? Here again (as in the case of the word structures that were directly licensed by our original lexical entries – see Chapter 6, Section 6.2.1), the second element in (31) is a description that can be satisfied by infinitely many resolved feature structures. Hence there are infinitely many lexical sequences that satisfy a lexical entry like (31). These lexical sequences are the ones that satisfy the constraints stated in (30a) and (31) as well as all of the constraints inherited from the supertypes of *cntn-lxm*. We represent the family of such lexical sequences as in (32), where we show all of the constraints inherited by the feature structure in the pair:

(32)

$$\left\langle \text{dog}\ ,\ \begin{bmatrix} \textit{cntn-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \boxed{1}[\text{PER 3rd}] \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle [\text{AGR} \quad \boxed{1}\ ] \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{dog} \\ \text{INST} & i \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{ARG-ST} & \left\langle \begin{matrix} \text{DP} \\ [\text{COUNT} \quad +] \end{matrix} \right\rangle \end{bmatrix} \right\rangle$$

Note that each of the lexical sequences in the family represented by (32) contains more information than what is shown. For reasons discussed above, however, none of these lexical sequences can be directly associated with a grammatical word structure. The role of a lexical entry, described more fully in the next section, is to define a family of lexical sequences that will give rise to a family of words. It is these words that are used to ground the construction of phrasal structures.

Given the type hierarchy and constraints just outlined, the rather complex set of specifications that we want to associate with a particular lexeme can be largely predicted simply by associating the lexeme with the appropriate type in the lexeme hierarchy.

Essentially, all that remains to be stipulated in a given lexical entry is its phonological form, the particular predication in its semantic restriction, and any exceptional properties it may have. The rest follows from 'the logic of the lexicon'. This is precisely what lexical stipulation should be reduced to.

Exercise 1: Understanding Constraint Inheritance

You should make sure you understand why (32) contains exactly the information it does. For each constraint in (32), identify which type it is a constraint on.

Proper nouns and pronouns instantiate the types *pn-lxm* and *pron-lxm*, which are constrained as follows:

(33) a.

$$\textit{pn-lxm} : \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \begin{bmatrix} \text{PER} & \text{3rd} \\ \text{NUM} & /\ \text{sg} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & [\text{MODE} \quad \text{ref}] \\ \text{ARG-ST} & /\ \langle\ \rangle \end{bmatrix}$$

b.

$$\textit{pron-lxm} : \begin{bmatrix} \text{SYN} & [\text{HEAD} \quad \textit{noun}] \\ \text{SEM} & [\text{MODE} \quad /\ \text{ref}] \\ \text{ARG-ST} & \langle\ \rangle \end{bmatrix}$$

These constraints require all proper nouns and pronouns to be [HEAD *noun*]. It also ensures that proper nouns are referential and that, by default, they are singular and have an empty ARG-ST list. As we saw at the beginning of this chapter, there are systematic exceptions to these last two constraints. (33b), on the other hand, imposes the nondefeasible constraint that pronouns have an empty ARG-ST list. There are no exceptional pronouns analogous to the names of mountain ranges or US sports teams. We have already seen pronouns whose MODE value is 'ana', rather than 'ref'. In addition, in Chapter 11 we will see examples of nonreferential pronouns. For both these reasons, the referentiality requirement in (33b) is defeasible, as indicated.

8.4.2 Verbal Lexemes

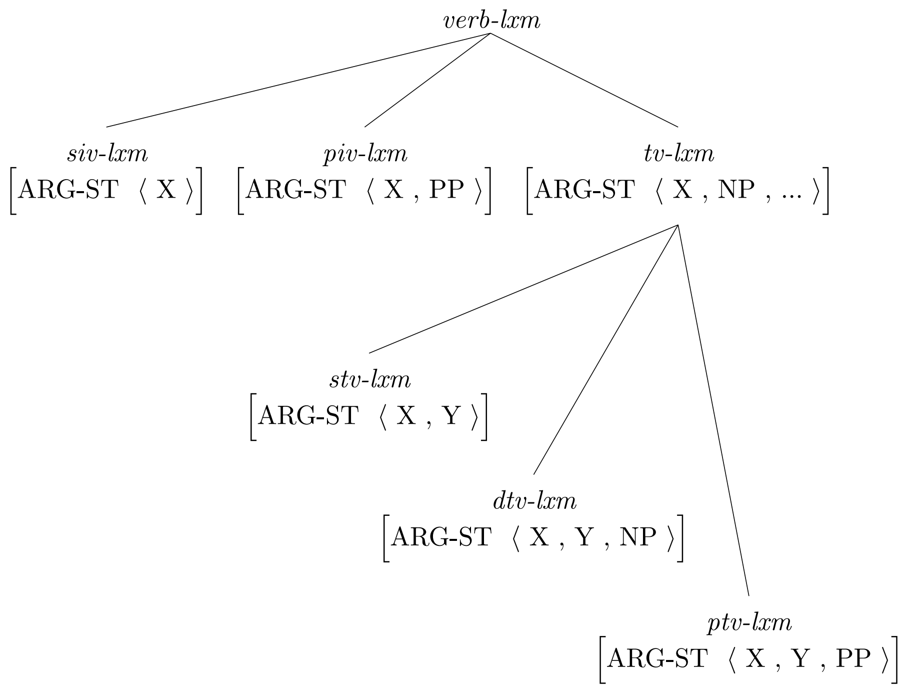
The next class of lexemes to consider is verbs. As we saw in Chapter 3, all verbs have certain properties in common, but there are also subclasses of verbs that differ from one another in systematic ways. Until now, we've had to stipulate these differences for each and every verb. In this section, we will see how the type hieararchy can capture generalizations about those subclasses.

Because *verb-lxm* is a subtype of *infl-lxm*, the SHAC guarantees that verbal lexemes will select for an agreeing specifier. In addition to this inherited constraint, we require that any instance of the type *verb-lxm* must have a HEAD value of type *verb* and the MODE value 'prop'. In addition, the argument structure of a lexeme of this type begins with an NP. (In Chapter 11, we discuss verbs that take non-NP subjects. This will lead us to revise this constraint.) The constraints just noted are consolidated into (34):

(34)
$$verb\text{-}lxm : \begin{bmatrix} \text{SYN} & [\text{HEAD} \;\; \textit{verb}] \\ \text{SEM} & [\text{MODE} \;\; \text{prop}] \\ \text{ARG-ST} & \langle \text{ NP , ... } \rangle \end{bmatrix}$$

The various subtypes of *verb-lxm* are distinguished by their ARG-ST specifications. The relevant part of our lexeme hierarchy is repeated in (35):

(35)



Here we have introduced the type *transitive-verb-lexeme* (*tv-lxm*) as a sister of the two intransitive verb types *strict-intransitive-verb-lexeme* (*siv-lxm*) and *prepositional-intransitive-verb-lexeme* (*piv-lxm*). Instances of *siv-lxm* take no complements at all (e.g. *sleep*); instances of *piv-lxm* take a PP complement (e.g. *rely*):[17]

(36) a. Leslie slept (*the baby).
 b. Dana relied *(on Hilary).

Similarly, the transitive verb lexemes are subclassified into *strict-transitive-verb-lexeme* (*stv-lxm*, e.g. *devour*), *ditransitive-verb-lexeme* (*dtv-lxm*, e.g. *hand*), and *prepositional-transitive-verb-lexeme* (*ptv-lxm*, e.g. *put*):

(37) a. Pat devoured *(the sandwich).
 b. Chris handed *(Bo) *(a ticket).
 c. We put *(the book) *(on the shelf).

As before, these types and their associated constraints (shown in (35)) allow us to replace lexical stipulation with type-based inference.

[17]We use the notation of an asterisk outside of the parentheses to mean that the example is ungrammatical WITHOUT the parenthetical material. An asterisk inside the parentheses means the example is ungrammatical WITH the parenthetical material.

Thus, by adding a lexical entry like (38), we ensure that there is a family of lexical sequences like (39):

(38)
$$\left\langle \text{give}\ ,\ \begin{bmatrix} \textit{dtv-lxm} & \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{give} \\ \text{SIT} & s \\ \text{GIVER} & i \\ \text{GIVEN} & j \\ \text{GIFT} & k \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{ARG-ST} & \langle\ \text{X}_i\ ,\ \text{Y}_j\ ,\ \text{Z}_k\ \rangle \end{bmatrix} \right\rangle$$

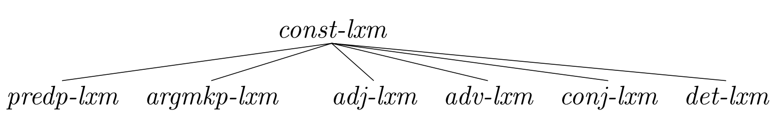
(39)
$$\left\langle \text{give}\ ,\ \begin{bmatrix} \textit{dtv-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} \\ \text{AGR} & \boxed{1} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ [\text{AGR}\ \boxed{1}]\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{give} \\ \text{SIT} & s \\ \text{GIVER} & i \\ \text{GIVEN} & j \\ \text{GIFT} & k \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{ARG-ST} & \langle\ \text{NP}_i\ ,\ \text{NP}_j\ ,\ \text{NP}_k\ \rangle \end{bmatrix} \right\rangle$$

This family of lexical sequences will give rise to structures which must obey the Argument Realization Principle, in consequence of which the first argument will be identified with the member of the SPR list and the remaining ARG-ST members will be identified with the two members of the verb's COMPS list.

Note that the lexical entry in (38) includes stipulations identifying the indices of the arguments with the role values (values of GIVER, GIVEN, and GIFT) of the lexeme's predication. In fact, much of this information is predictable on the basis of the lexeme's meaning. Though we cannot develop such an approach here, there is considerable work that has proposed ways of eliminating further redundancy from lexical entries like (38). Eliminating such redundancy is one of the goals of a 'linking theory', as mentioned in Chapter 7.

8.4.3 Constant Lexemes

Let us turn now to noninflecting lexemes, that is, the various subtypes of the type *const-lxm* that we have not yet considered:

(40)


These correspond to various kinds of lexical entries that undergo no inflection in English.[18] Since only expressions (words or phrases) enter into grammatical structures, these lexemes must all undergo a lexical rule in order to produce (phonologically identical) words that can be of some grammatical use. We'll see this rule in Section 8.7.3.

In Chapter 7 we distinguished two kinds of prepositions – those that function as predicates and those that serve as argument markers. This distinction corresponds to the two types *predicational-preposition-lexeme* (*predp-lxm*) and *argument-marking-preposition-lexeme* (*argmkp-lxm*) in (40). Recall that in our earlier discussion we distinguished these prepositions in terms of their semantics. Only prepositions of type *predp-lxm* introduce their own predication. Argument-marking prepositions simply take on the INDEX and MODE value of their object. These effects are ensured by the following type constraints:

(41) a.

$$\textit{predp-lxm}: \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{prep} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\, \text{X} \,\rangle \\ \text{MOD} & \langle\, \text{Y} \,\rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \\ \text{RESTR} & \langle\, \text{Z} \,\rangle \end{bmatrix} \\ \text{ARG-ST} & \langle\, \text{NP}\,,\, \text{NP} \,\rangle \end{bmatrix}$$

b.

$$\textit{argmkp-lxm}: \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{prep} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \boxed{1} \\ \text{INDEX} & \boxed{2} \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \\ \text{ARG-ST} & \left\langle \overset{\text{NP}}{\begin{bmatrix} \text{MODE} & \boxed{1} \\ \text{INDEX} & \boxed{2} \end{bmatrix}} \right\rangle \end{bmatrix}$$

Only predicational prepositions can be modifiers. Accordingly, *argmkp-lxm* says nothing about MOD and thus inherits the default constraint [MOD / ⟨ ⟩] from *lexeme*. *predp-lxm*, on the other hand, overrides this constraint with [MOD ⟨ Y ⟩]. This non-empty MOD value allows these prepositions to be modifiers.[19] When they appear as complements of verbs (as in (42), discussed in Chapter 7), this non-empty MOD value is irrelevant.

(42) I wrapped the blanket [around me].

[18] The type *adj-lxm* arguably should be classified as a subtype of *infl-lxm*, rather than as a subtype of *const-lxm*, in light of the fact that many adjectival lexemes give rise to comparative and superlative forms, e.g. *tall*, *taller*, *tallest*. We will not pursue this matter here. Note also that the classification of lexemes into inflecting and constant is language-specific. As we saw in Problem 2 of Chapter 4, for example, determiners in Spanish inflect for agreement information.

[19] This MOD value is obviously not constrained enough, as there are things that PPs can't modify (e.g. determiners). *predp-lxm* or its instances need to say something more specific, although we won't explore this refinement here.

Note also that *predp-lxm* specifies a two-place ARG-ST list and a non-empty SPR value. Once a word is built from a predicational preposition, its first argument must be identified with the SPR element, in accordance with the ARP. What plays these roles in (42) is the NP *the blanket*, which is also an argument of the verb *wrapped*. This is the first time we have seen one constituent serving as an argument of more than one predicate at the same time. This is a common phenomenon, however, as we will see in subsequent chapters. Developing an analysis of such cases is the topic of Chapter 12.[20]

The argument-marking prepositions, because of the constraint in (41b), project a nonmodifying PP with an empty specifier list whose MODE and INDEX values are identified with those of the preposition's NP object:

(43) He talks [to himself].

As described in Chapter 7, this analysis allows the objects of argument-marking prepositions to enter into binding relations with other NPs. Finally, recall that some prepositions, for example, *around*, behave either as predicational or as argument-marking. Hence the following example is also well-formed:

(44) I wrapped the blanket [around myself].

This pattern of optional reflexivization is now neatly accounted for by allowing *around* to live a double life (via two separate lexical entries) as either a predicational or an argument-marking preposition.

For the sake of completeness, we include the following four type constraints on the remaining four subtypes of *const-lxm*:

(45) a.
$$\textit{adj-lxm}: \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{adj} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \text{ X } \rangle \\ \text{MOD} & \langle \text{ [HEAD } \textit{noun}] \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \langle \text{ NP , ... } \rangle \\ \text{SEM} & [\text{MODE} \;\; \text{prop}] \end{bmatrix}$$

b.
$$\textit{adv-lxm}: \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{adv} \\ \text{VAL} & [\text{MOD} \;\; \langle \text{ [HEAD } \textit{verb}] \rangle] \end{bmatrix} \\ \text{SEM} & [\text{MODE} \;\; \text{none}] \end{bmatrix}$$

c.
$$\textit{conj-lxm}: \begin{bmatrix} \text{SYN} & [\text{HEAD} \;\; \textit{conj}] \\ \text{SEM} & [\text{MODE} \;\; \text{none}] \\ \text{ARG-ST} & \langle \; \rangle \end{bmatrix}$$

d.
$$\textit{det-lxm}: \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{det} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & / \langle \; \rangle \\ \text{COMPS} & \langle \; \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & [\text{MODE} \;\; \text{none}] \end{bmatrix}$$

[20] Note in addition that nothing in our analysis blocks the projection of subject-saturated PPs like [*My blanket* [*around me*]]. As noted in Chapter 4 these occur only in restricted circumstances, e.g. as 'absolute' or 'small' clauses.

The constraints on the type *det-lxm* are meant to accommodate the results of Chapter 6, Problem 3 – that is, that *'s* is a determiner that exceptionally takes an obligatory NP specifier. The types *adj-lxm*, *adv-lxm* and *conj-lxm* will require further constraints, but we omit discussion of them here.

8.4.4 Lexemes vs. Parts of Speech

It may be somewhat surprising that our type hierarchy posits two distinct types corresponding roughly to each of the traditional parts of speech. In addition to *noun*, *verb*, etc. – the subtypes of *pos* introduced in Chapter 3 – we now have types like *cn-lxm*, *pn-lxm*, *verb-lxm*, and so forth, which are subtypes of the type *lexeme*. It is important to understand that these two sets of types serve rather different functions in our grammar. The subtypes of *pos* specify which features are appropriate for particular categories of words and phrases. They thus serve to organize the various parts of speech that our grammar has to recognize. The subtypes of *lexeme*, on the other hand, introduce constraints on what combinations of feature values are possible, for example, the SHAC or the constraint that verbs require propositional mode. These typically involve argument structure (and/or valence features) as well as HEAD features or SEM features. Consequently, the *pos* subtypes (*noun, verb*, etc.) frequently appear inside of the constraints associated with the *lexeme* subtypes (*noun-lxm, verb-lxm*, etc.).

The type hierarchy simplifies our descriptions in two ways: it saves us from having to assign values to features where they would do no work, for example, PER (person) in prepositions or CASE in verbs; and it allows us to stipulate common combinations of feature values only once, using (default) inheritance to account for their distribution. The hierarchy contains two sets of types corresponding roughly to the traditional parts of speech then, because the hierarchy serves these two separate functions.

8.4.5 The Case Constraint

Up to this point, we have made no mention of CASE specifications in our lexical type hierarchy. Thus, nothing yet guarantees that NPs in English must be accusative except when they are the subject of a finite verb form. One might think this is a constraint on lexemes, but this would make certain incorrect predictions. As we will see in later chapters, certain lexical rules (such as the Passive Lexical Rule introduced in Chapter 10), have the effect of reordering ARG-ST lists. Such reordering never results in ARG-ST-initial elements being specified as [CASE acc]. For this reason, we will treat the assignment of accusative case as a fact about words, not about lexemes. The easiest way to do this is to add the following constraint to our definition of lexical licensing:[21]

(46) Case Constraint

An outranked NP is [CASE acc].

This principle allows us to keep our constraints on verbal lexemes just as we formulated them above, with no mention of case. Thus it is unnecessary to specify lexically the accusative case for most objects, providing a significant improvement on the analysis of English case suggested in Problem 6 of Chapter 4. Notice, however, that (46) is a one-

[21] Thanks to Louis Eisenberg for pointing out the possibility of this formulation of the Case Constraint.

way implication: it says that certain NPs are accusative, but it says nothing about which NPs are not accusative. The nominative case, characteristic of subjects, will need to be specified in some other way (a point to which we return later in this chapter).

Finally, it must be stressed that the Case Constraint is specific to English. Many other languages exhibit far more complex case systems; see, for example, the problems on Icelandic and Wambaya in Chapter 4.

Exercise 2: Case on Objects of Prepositions

Does the Case Constraint as stated in (46) account for the fact that both argument-marking and predicational prepositions require accusative case on their objects? Why or why not?

8.5 The FORM Feature

In the next section, we'll introduce the lexical rules that relate the lexemes discussed above to the inflected words they give rise to. First, however, we return to the feature FORM, which came up briefly in the discussion of imperatives in Chapter 7 (Section 7.6).

8.5.1 FORM Values for Verbs

In general, different inflected words arising from the same lexeme have different distributions. In order to capture those different distributions in our grammar, we must ensure that they have different feature specifications. In many cases, this work is done by features we have already introduced. For example, singular and plural nouns differ in their NUM values. In the case of verbs, however, the inflected forms differ in their distributions without differing in any of the features we have posited for other uses. For example, the verb after a modal must be in the base form, the verb after auxiliary *have* must be a past participle, and the main verb in a sentence must be finite (past or present tense):

(47) a.
$$\text{Kim may} \left\{ \begin{array}{l} \text{leave} \\ \text{*leaves} \\ \text{*leaving} \\ \text{*left} \end{array} \right\}.$$

b.
$$\text{Kim has} \left\{ \begin{array}{l} \text{*leave} \\ \text{*leaves} \\ \text{*leaving} \\ \text{left} \end{array} \right\}.$$

c.
$$\text{Kim} \left\{ \begin{array}{l} \text{*leave} \\ \text{leaves} \\ \text{*leaving} \\ \text{left} \end{array} \right\}.$$

We will use the feature FORM to distinguish between these different forms. For verbs, we will posit the following (atomic) values for the feature FORM:[22]

(48) base — The bare uninflected form, as in *Andy would* **eat** *rice*, *Andy tried to* **eat** *rice*, or **Eat** *rice!*

fin — 'Finite', i.e. present or past tense, as in *Andy* **eats** *rice* or *Andy* **ate** *rice*

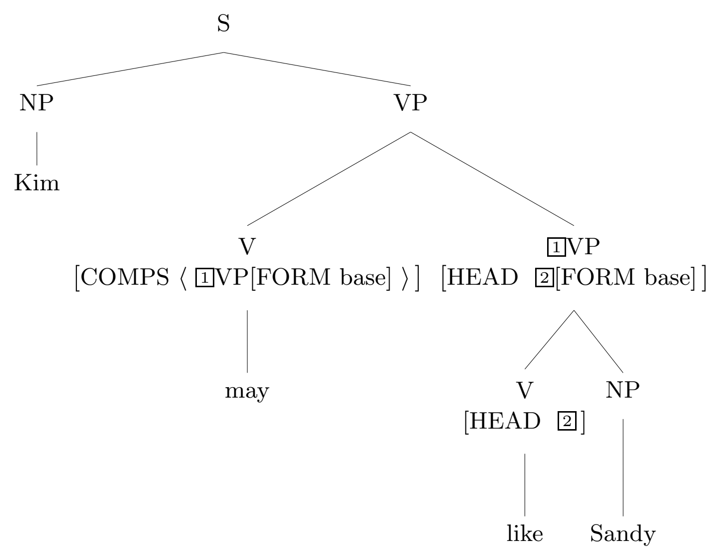
prp — 'Present participle', suffixed with *-ing*, usually following some form of *be*, as in *Andy is* **eating** *rice*

psp — 'Past participle' (or 'perfect participle'), the form that follows *have*, as in *Andy has* **eaten** *rice*

pass — 'Passive', as in *Rice was* **eaten** *by Andy* (to be discussed in Chapter 10)

Treating FORM as a head feature will allow us to get a handle on the co-occurrence restrictions illustrated in (47). As discussed in detail in Chapter 13, we treat auxiliaries like *may* or *has* as verbs that take a VP complement. Each auxiliary specifies a particular FORM value on its complement, and the Head Feature Principle ensures that the FORM value of the selected VP is the same as that of the head verb inside that VP. This is illustrated in (49):

(49)



[22] Particular researchers have made slightly different assumptions about the value for the feature FORM (or its equivalent). For example, 'ger' (for 'gerund') has sometimes been proposed for a kind of word not covered here. Like present participles, gerunds are suffixed with *-ing*, but unlike present participles, gerunds head phrases that have the distribution of NPs. The occurrences of *singing* in (i)–(iii) are present participles; those in (iv)–(vi) are gerunds:

(i) The birds are singing.
(ii) Anyone singing in class will be punished.
(iii) Ashley began singing Christmas carols in October.
(iv) Ashley's singing Christmas carols in October annoyed Jordan.
(v) We denied singing during class.
(vi) Don't even think about singing!

The analysis of gerunds is beyond the scope of this text. Hence, we will not consider the question of whether there should be a FORM value for gerunds.

Another benefit of treating FORM as a head feature is that it will allow us to refine our definition of the initial symbol. In Chapter 6, we gave the initial symbol as ‘S’, i.e. the combination of constraints shown in (50):

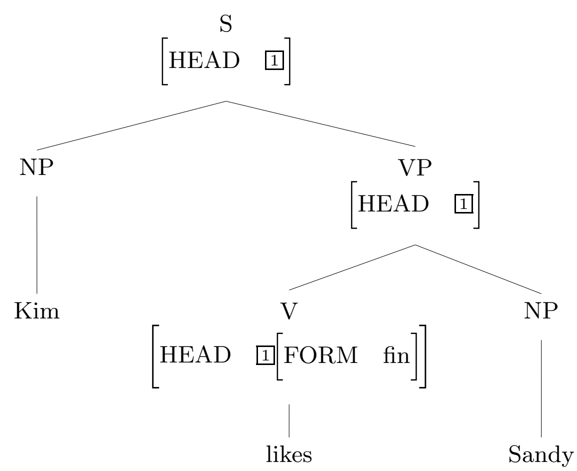
(50)
$$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

We would now like to add the constraint that only FINITE Ss can be stand-alone sentences. We can achieve this by adding the specification [FORM fin] to our definition of the ‘initial symbol’, which specifies which sentences can serve as independent utterances:

(51)
$$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} \\ \text{FORM} & \text{fin} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Since FORM is a HEAD feature, the only Ss that are [FORM fin] are those which are ultimately headed by verbs that are [FORM fin], as illustrated in (52):[23]

(52)



8.5.2 FORM and Coordination

The previous section argued that FORM is best treated as a HEAD feature. The current version of our Coordination Rule (last discussed in Chapter 5) does not identify the HEAD values of the conjuncts with each other or with the mother. It turns out that this makes incorrect predictions. Where verbs select for VPs of a particular form, that selection holds even if the complement is a coordinated VP:

[23]The one exception is imperatives, which we treat as finite Ss that are not headed by a finite verb. This discrepancy comes about because the Imperative Rule is a non-headed rule and it changes the FORM value. In this sense, imperative sentences are not in fact headed by anything.

(53)

$$\text{Dana helped Leslie}\left\{\begin{array}{l}\text{move}\\ ^{*}\text{moves}\\ ^{*}\text{moving}\\ ^{*}\text{moved}\end{array}\right\}.$$

$$\text{Dana helped Leslie}\left\{\begin{array}{l}\text{pack and move}\\ ^{*}\text{packs and move}\\ ^{*}\text{pack and moves}\\ ^{*}\text{packs and moves}\\ ^{*}\text{packing and move}\\ ^{*}\text{pack and moving}\\ ^{*}\text{packing and moving}\\ \ldots\end{array}\right\}.$$

Likewise, stand-alone coordinate sentences must contain a finite verb as the head of each conjunct:

(54) a. Dana walked and Leslie ran.
b.*Dana walking and Leslie ran.
c.*Dana walked and Leslie running.
d.*Dana walking and Leslie running.

In order to capture these facts, we add a constraint to our Coordination Rule that identifies the FORM values of each conjunct with that of the mother. In making this revision, the Coordination Rule has almost reached its final form: (We will revisit it once more in Chapter 14.)

(55) Coordination Rule (Chapter 8 Version)

$$\begin{bmatrix}\text{FORM} & \boxed{1}\\ \text{VAL} & \boxed{0}\\ \text{IND} & s_0\end{bmatrix} \rightarrow$$

$$\begin{bmatrix}\text{FORM} & \boxed{1}\\ \text{VAL} & \boxed{0}\\ \text{IND} & s_1\end{bmatrix} \ldots \begin{bmatrix}\text{FORM} & \boxed{1}\\ \text{VAL} & \boxed{0}\\ \text{IND} & s_{n-1}\end{bmatrix} \begin{bmatrix}\text{HEAD} & \textit{conj}\\ \text{IND} & s_0\\ \text{RESTR} & \langle\, [\text{ARGS}\ \langle s_1 \ldots s_n\rangle]\,\rangle\end{bmatrix} \begin{bmatrix}\text{FORM} & \boxed{1}\\ \text{VAL} & \boxed{0}\\ \text{IND} & s_n\end{bmatrix}$$

Adding FORM identity constraints to the Coordination Rule raises two important (and related) points. The first is that FORM must now be appropriate for all *pos* types that can be coordinated. If it weren't, then expressions with *pos* types that don't bear the FORM feature could never be compatible with the rule. The second point to note is that the FORM values we have posited so far (prp, psp, pass, etc.) are only appropriate for verbs. This means that the Coordination Rule no longer incorrectly allows the coordination of, say, NP and S (cf. Section 4.7 of Chapter 4):

(56)*Dana walked and Kim.

Since FORM must be appropriate for all parts of speech that can coordinate, we can use the FORM identity condition to impose the requirement that conjunct daughters must have the same part of speech, but we can do so without identifying their HEAD values. (Recall from Section 4.7 of Chapter 4 that requiring HEAD identity is too strong, because it disallows conjuncts with different AGR values.) We do this by positing distinct FORM values for each part of speech. Nouns will be [FORM nform], adjectives will be [FORM aform], and so forth. For many lexical classes, we can guarantee these correlations between part-of-speech types and FORM values in a general way by stating defeasible constraints on the relevant subtype of *pos*. (57) is such a constraint:

(57) $\quad \textit{noun}: \begin{bmatrix}\text{FORM} & / \text{ nform}\end{bmatrix}$

This constraint is defeasible, as we will use special FORM values for certain nouns and pronouns in the treatment of expletives and idiomatic expressions that we present in Chapter 11. We will also posit special values of FORM to distinguish among prepositions in our account of selectional dependencies between verbs and prepositions (see Chapter 10). But there is no need to assume a FORM value 'vform' or to give a default FORM value to verbs, as all inflected forms of verbs are given a specific FORM value by one of the inflectional rules discussed in the next section.

8.6 Lexical Rules

The lexical rule is a mechanism for further reducing redundancy and stipulation in the lexicon by stating systematic regularities that hold between lexemes and the words that are 'realizations' of those lexemes.

It is traditional to think of words (or at least certain kinds of words) as being built up from smaller units through the addition of affixes. We have followed this tradition by using our notion of types to distinguish *lexeme* from *word*. For most nouns and verbs, we will assume that there is only one lexical entry. As explained in the previous section, each such lexical entry describes a family of lexical sequences. We then characterize all the nominal and verbal words in terms of lexical rules that relate the basic lexical sequences to others whose second member is a feature structure of type *word*.

Although it is intuitive, as well as traditional, to think of a lexical rule as a process that takes lexemes (or words) as input and gives distinct lexical entities as output, it is not necessary to introduce a new kind of device to capture the essential insights of lexical rules.[24] In fact, lexical rules can be modeled as feature structures of a special type, which we'll call *lexical-rule* (*l-rule*). Feature structures of this type specify values for the features INPUT and OUTPUT. There are a number of advantages to be derived from modeling lexical rules in this way. For example, they can be organized into a type hierarchy, with common properties factored into constraints on common supertypes. This is particularly attractive, as languages that have more complicated morphological paradigms require

[24]There have been many proposals for how to formulate lexical rules, ranging from 'metadescription' approaches that apply generatively to map lexical entries (descriptions) into lexical descriptions and 'redundancy rule' approaches that treat them as stating generalizations that hold over a pre-existing set of entries. Our own approach, following in key respects Briscoe and Copestake (1999), is based on feature structures, whose resolved nature allows us to account for productive lexical rule relations without introducing new analytic devices.

families of lexical rules that have many properties in common. This is true, for example, of the lexical rules that are required for the Spanish verb paradigms we considered at the beginning of this chapter.

A second advantage of modeling lexical rules as feature structures is that we can use defeasible identity constraints on the values of the features INPUT and OUTPUT. A defeasible identity constraint can guarantee that constraints holding of a lexical rule input are carried over to the rule's output, by default. This will let us streamline the formulation of lexical rules, allowing our grammar to stipulate only those properties that add or alter specific pieces of information.

We can thus think of a lexical rule as a feature structure that corresponds to a particular relation holding between pairs of lexical sequences. We will here consider two types of *l(exical)-rule*: *inflectional-rule* (*i-rule*) and *derivational-rule* (*d-rule*), organized into the following type hierarchy:[25]

(58)

l-rule

i-rule *d-rule*

All feature structures of type *l-rule* obey the following constraint:

(59)
$$\textit{l-rule}: \begin{bmatrix} \text{INPUT} & \textit{l-sequence}\langle\ \text{X}\ ,\ [\text{SEM}\ /\ \boxed{2}]\ \rangle \\ \text{OUTPUT} & \textit{l-sequence}\langle\ \text{Y}\ ,\ [\text{SEM}\ /\ \boxed{2}]\ \rangle \end{bmatrix}$$

What (59) says is that both the input and output of a lexical rule are lexical sequences (see page 236) and that the SEM values of the lexical rule's input and output are identical, by default. The types *i-rule* and *d-rule*, and particular lexical rules which are instances of those types, will introduce further constraints, as discussed below.

It is important to note that lexical rules, like lexical entries and phrase structure rules are a kind of description. The objects that satisfy lexical rules are LEXICAL RULE INSTANTIATIONS. Lexical rule instantiations are fully specified feature structures. They are not, however, models of words or sentences. We incorporate the effect of lexical rules into our construction of models of sentences by using the lexical sequences that are the OUTPUT values of lexical rule instantiations to license word structures.[26] (See Chapter 9 for a formal description of how this works.)

8.7 Inflectional Rules

The type *i-rule* is a subtype of *l-rule*, and thus inherits the constraints shown in (59). In addition, inflectional rules obey stronger constraints, namely, those we formulate as in (60):

[25] Another subtype of *l-rule* will be introduced in Chapter 11.

[26] Of course, we only use those lexical sequences whose second member is of type *word*, i.e. those lexical sequences that are the OUTPUT value of an inflectional lexical rule (see Section 8.7) or a post-inflectional lexical rule (see Chapter 11).

(60)

$$
\textit{i-rule}:\begin{bmatrix}
\text{INPUT} & \left\langle \text{X}\,,\begin{bmatrix}\textit{lexeme}\\ \text{SYN} & \boxed{3}\\ \text{ARG-ST} & \boxed{\text{A}}\end{bmatrix}\right\rangle\\[2ex]
\text{OUTPUT} & \left\langle \text{Y}\,,\begin{bmatrix}\textit{word}\\ \text{SYN} & \boxed{3}\\ \text{ARG-ST} & \boxed{\text{A}}\end{bmatrix}\right\rangle
\end{bmatrix}
$$

(60) says that the input of an inflectional rule must be of type *lexeme* and that its output must be of type *word*. (60) also requires that the input and output share both SYN and ARG-ST values. Note that this last requirement allows inflectional rules to add constraints to the output, as long as they are consistent with constraints placed on the input lexeme. However, (60) guarantees that inflectional rules perform no 'destructive' changes to the SYN or ARG-ST value of a lexeme, for this would contradict the indicated identity constraints. We will illustrate this property of inflectional rules in this section. We take up derivational rules in Section 8.8 and in subsequent chapters.

8.7.1 Rules for Common Noun Inflection

Once we have the type constraints just outlined, we may introduce specific inflectional rules. These rules inherit constraints from their types (*i-rule, l-rule*), just as the feature structures of lexical entries do. Let's consider first the inflectional rule that relates common noun lexemes to their singular word realizations, i.e. the rule that is responsible for words like *dog* and *water*. These words are specified as (third-person) singular, but otherwise they contain just the (phonological, syntactic and semantic) information that they inherit from the lexeme they are related to. Given this, we can formulate the rule we need as shown in (61), where the form of the output word is required to be identical to that of the input lexeme:[27]

(61) Singular Noun Lexical Rule

$$
\begin{bmatrix}
\textit{i-rule}\\
\text{INPUT} & \left\langle \boxed{1}\,,\ \textit{cn-lxm}\right\rangle\\[1ex]
\text{OUTPUT} & \left\langle \boxed{1}\,,\begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{AGR} & \begin{bmatrix}\text{NUM sg}\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix}\right\rangle
\end{bmatrix}
$$

Since the Singular Noun Lexical Rule is of type *i-rule* (constrained as shown in (59) and (60)), it follows from the theory of constraint inheritance sketched above that the lexical rule is constrained as follows:

[27]It is thus an 'accident' of English morphology that singular nouns, unlike plural nouns, have no inflectional ending.

(62) Singular Noun Lexical Rule (with inherited constraints)

$$\left[\begin{array}{ll} \textit{i-rule} & \\ \text{INPUT} & \left\langle \boxed{1}\, , \left[\begin{array}{ll} \textit{cn-lxm} & \\ \text{SYN} & \boxed{3} \\ \text{SEM} & \boxed{2} \\ \text{ARG-ST} & \boxed{A} \end{array}\right] \right\rangle \\ \text{OUTPUT} & \left\langle \boxed{1}\, , \left[\begin{array}{ll} \textit{word} & \\ \text{SYN} & \boxed{3}\left[\begin{array}{ll}\text{HEAD} & \left[\begin{array}{ll}\text{AGR} & \left[\begin{array}{ll}\text{PER} & \text{3rd}\\ \text{NUM} & \text{sg}\end{array}\right]\end{array}\right]\end{array}\right] \\ \text{SEM} & \boxed{2} \\ \text{ARG-ST} & \boxed{A}\end{array}\right]\right\rangle \end{array}\right]$$

Notice that nothing in (61) contradicts the defeasible identity constraint in (59). Hence that constraint remains in effect in (62). The set of constraints shown in (62) is exactly what we get as the result of combining the defeasible constraints in (59) with the inviolable constraints in (60) and (61).[28]

Let us consider a simple example. In (32) above, we explained how our grammar gives rise to the family of lexical sequences represented by the following:[29]

(63)

$$\left\langle \text{dog}\, , \left[\begin{array}{ll} \textit{cntn-lxm} & \\ \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll} \textit{noun} & \\ \text{AGR} & \boxed{1}\left[\text{PER 3rd}\right] \end{array}\right] \\ \text{VAL} & \left[\text{SPR} \quad \langle\, [\text{AGR} \quad \boxed{1}]\, \rangle\right] \end{array}\right] \\ \text{SEM} & \left[\begin{array}{ll} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \left[\begin{array}{ll} \text{RELN} & \textbf{dog} \\ \text{INST} & i \end{array}\right] \right\rangle \end{array}\right] \\ \text{ARG-ST} & \left\langle \begin{array}{c} \text{DP} \\ \left[\text{COUNT} \quad +\right] \end{array} \right\rangle \end{array}\right] \right\rangle$$

[28]Note, however, that if an input were specified as [NUM pl] (plausible examples might be *scissors* or *pants*), then it would fail to undergo this lexical rule. That is, there could be no relation between the input lexical sequence and any output lexical sequence that satisfied the constraint specified in (62).

[29]Some of the constraints the lexical entry for *dog* inherits (from *cn-lxm* and *lexeme*) are defeasible constraints on those types. In a fully specified lexical sequence, however, those defeasible constraints that are not overridden become inviolable. Thus the INPUT specifications of a lexical rule cannot override any constraint associated with a lexical entry.

Any of the lexical sequences in (63) is a possible value of the feature INPUT in a feature structure that satisfies the Singular Noun Lexical Rule (with its inherited constraints – shown in (62) above). If the INPUT of (62) is resolved to such a lexical sequence, then the lexical sequences satisfying the value of the feature OUTPUT will all look like (64):

(64)
$$\left\langle \text{dog}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \boxed{1}\begin{bmatrix} \text{PER} & \text{3rd} \\ \text{NUM} & \text{sg} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \boxed{2}[\text{AGR}\quad \boxed{1}]\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{dog} \\ \text{INST} & i \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{ARG-ST} & \left\langle \begin{matrix} \boxed{2}\text{DP} \\ \begin{bmatrix} \text{COUNT} & + \end{bmatrix} \end{matrix} \right\rangle \end{bmatrix} \right\rangle$$

These feature structures are licensed as lexical sequences whose second member is a feature structure of type *word* (and hence obeying the ARP).[30] By the informal definition given in Section 8.6, these *word*s can be used as the daughters of phrase structure rules to build phrases and sentences. We will revise the formal definition of lexical licensing accordingly in Chapter 9.

In the remainder of this section, we will briefly introduce some of the particular lexical rules we posit to relate lexemes to words. In the next section, we discuss briefly DERIVATIONAL RULES, which relate lexemes to lexemes. In Chapter 11, we will also introduce lexical rules that relate words to words.

The next lexical rule to consider is the rule that maps nominal lexemes into lexical sequences for their corresponding plural forms:

(65) Plural Noun Lexical Rule
$$\begin{bmatrix} \textit{i-rule} & \\ \text{INPUT} & \left\langle \boxed{1}\ ,\ \textit{cntn-lxm} \right\rangle \\ \text{OUTPUT} & \left\langle \text{F}_{NPL}(\boxed{1})\ ,\ \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

Here, F_{NPL} is a morphological function that applies to a nominal base in English, giving its plural form. This function is sketched in (66):

[30] In what follows, we will loosely talk of lexical rules relating lexemes to words, etc.

(66)

X	$\mathrm{F}_{NPL}(X)$
child	children
ox	oxen
woman	women
fish	fish
index	indices
...	...
(otherwise)	X-s

There are various issues that arise in connection with such inflectional functions, e.g. how best to accommodate subregularities and similarities across different morphological functions, but we will steer clear of these issues here.

The lexical rule sketched in (65) inherits constraints from the types *i-rule* and *l-rule*. The combination of (65) and (59) and (60) is indicated in (67):

(67) Plural Noun Lexical Rule and Inherited Constraints

$$\begin{bmatrix} \textit{i-rule} & \\ \text{INPUT} & \left\langle \boxed{1}\ ,\ \begin{bmatrix} \textit{cntn-lxm} & \\ \text{SYN} & \boxed{3} \\ \text{SEM} & \boxed{2} \\ \text{ARG-ST} & \boxed{A} \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \mathrm{F}_{NPL}(\boxed{1})\ ,\ \begin{bmatrix} \textit{word} & \\ \text{SYN} & \boxed{3}\left[\text{HEAD}\ \ [\text{AGR}\ \ [\text{NUM}\ \ \text{pl}]]\right] \\ \text{SEM} & \boxed{2} \\ \text{ARG-ST} & \boxed{A} \end{bmatrix} \right\rangle \end{bmatrix}$$

The Plural Noun Lexical Rule thus guarantees that for every count noun lexeme[31] there is a corresponding plural noun word with identical SYN, SEM, and ARG-ST values, whose form is determined by the function F_{NPL}. The requirement that the input be *cntn-lxm* keeps the rule from applying to mass nouns like *furniture*, so that there is no word **furnitures*. The Plural Noun Lexical Rule thus allows for lexical sequences like (68):[32]

[31] Other than those that might be lexically restricted to be singular.

[32] A complete formulation of both lexical rules discussed so far would require the introduction of a fundamental difference between the semantics of singular and plural nouns. But a semantic analysis of singular and plural nouns – which would have to include a treatment of the count/mass distinction – is beyond the scope of this book.

(68)

$$\left\langle \text{dogs}\ , \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \boxed{1}\begin{bmatrix} \text{PER} & \text{3rd} \\ \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \boxed{2}[\text{AGR}\ \ \boxed{1}]\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{dog} \\ \text{INST} & i \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{ARG-ST} & \left\langle \begin{matrix} \boxed{2}\text{DP} \\ [\text{COUNT}\ \ +] \end{matrix} \right\rangle \end{bmatrix} \right\rangle$$

8.7.2 Rules for Inflected Verbal Words

We posit additional lexical rules for the various inflected forms of verbs, beginning with the rule for the 3rd-singular present form:

(69) 3rd-Singular Verb Lexical Rule

$$\begin{bmatrix} \textit{i-rule} & \\ \text{INPUT} & \left\langle \boxed{3}\ , \begin{bmatrix} \textit{verb-lxm} & \\ \text{SEM} & [\text{RESTR}\ \ \boxed{A}] \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \text{F}_{\mathit{3SG}}(\boxed{3})\ , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{fin} \\ \text{AGR} & \textit{3sing} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & [\text{RESTR}\ \ \boxed{A} \oplus \ldots\ ] \\ \text{ARG-ST} & \langle\ [\text{CASE}\ \ \text{nom}]\ , \ldots\ \rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

As with the Plural Noun Lexical Rule, we have glossed over the morphological component of the 3rd-Singular Verb Lexical Rule by simply giving it a name: F_{3SG}.

The semantic effect of this rule is to preserve the basic semantics of the input, but to add the tense information. That is, MODE and INDEX are unchanged, but a predication representing tense is added to the RESTRICTION. Predications of this type will be supressed here and throughout, with ... standing in.[33] What the rule in (69) says, then, is that for any verbal lexeme, there is a corresponding third-person singular finite verb (a word) that takes a nominative subject. Further, the morphology and semantics of the latter verb are systematically related to those of the input lexeme.

[33] One way to represent tense in a system such as ours is to have the present tense predication require that the INDEX value – the situation described by the verb – temporally overlap the utterance time. Thus, according to this rule, using a 3rd singular present form of a verb lexeme imposes the requirement that the situation introduced by the verb be located in some temporal interval that overlaps the time of the utterance. Tense semantics is also beyond the scope of this text.

We turn next to the rule that licenses finite verbs with subjects other than third-person singular NPs. Because the type distinction we have drawn between the AGR values *3sing* and *non-3sing* already distinguishes third-singular NPs from all others, this rule is almost identical to the last one, as shown in (70):

(70) Non-3rd-Singular Verb Lexical Rule

$$\left[\begin{array}{ll} \textit{i-rule} & \\ \text{INPUT} & \left\langle \boxed{1}\,, \left[\begin{array}{ll}\textit{verb-lxm} & \\ \text{SEM} & \left[\text{RESTR}\ \ \boxed{A}\right]\end{array}\right]\right\rangle \\ \text{OUTPUT} & \left\langle \boxed{1}\,, \left[\begin{array}{ll} \text{SYN} & \left[\text{HEAD} \left[\begin{array}{ll}\text{FORM} & \text{fin}\\ \text{AGR} & \textit{non-3sing}\end{array}\right]\right] \\ \text{SEM} & \left[\text{RESTR}\ \ \boxed{A} \oplus ...\right] \\ \text{ARG-ST} & \langle\ [\text{CASE}\ \ \text{nom}]\ , ...\ \rangle \end{array}\right]\right\rangle \end{array}\right]$$

The only differences between (70) and (69) are: (i) no change in morphology is introduced, and (ii) the AGR value of the OUTPUT is *non-3sing* (see Chapter 4, Section 4.6 for further discussion). Outputs of this rule, for example the one shown in (71), sanction word structures that can never combine with a third-person singular subject:

(71)

$$\left\langle \text{give}\ , \left[\begin{array}{ll} \textit{word} & \\ \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll} \textit{verb} & \\ \text{AGR} & \boxed{1}\textit{non-3sing} \\ \text{FORM} & \text{fin}\end{array}\right] \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \langle\ \boxed{2}[\text{AGR}\ \boxed{1}]\ \rangle \\ \text{COMPS} & \langle\ \boxed{3}\ , \boxed{4}\ \rangle \end{array}\right] \end{array}\right] \\ \text{SEM} & \left[\begin{array}{ll} \text{MODE} & \text{prop} \\ \text{INDEX} & s \\ \text{RESTR} & \left\langle \left[\begin{array}{ll} \text{RELN} & \textbf{give} \\ \text{SIT} & s \\ \text{GIVER} & i \\ \text{GIVEN} & j \\ \text{GIFT} & k \end{array}\right] \right\rangle \oplus ... \end{array}\right] \\ \text{ARG-ST} & \left\langle \begin{array}{c} \boxed{2}\text{NP}_i \\ \left[\text{CASE}\ \ \text{nom}\right] \end{array}, \boxed{3}\text{NP}_j\ , \boxed{4}\text{NP}_k \right\rangle \end{array}\right] \right\rangle$$

As with the 3rd-Singular Verb Lexical Rule, the semantics of the output is systematically related to the semantics of the input.

The two rules just discussed license the present tense forms of verbs. The next rule creates lexical sequences for the past tense forms. English makes no distinction between singular and plural in past tense forms (aside from *was* vs. *were*);[34] hence only one rule is needed:

(72) Past-Tense Verb Lexical Rule

$$\begin{bmatrix} \textit{i-rule} & \\ \text{INPUT} & \left\langle \boxed{3}\ , \begin{bmatrix}\textit{verb-lxm} & \\ \text{SEM} & \begin{bmatrix}\text{RESTR} & \boxed{A}\end{bmatrix}\end{bmatrix}\right\rangle \\ \text{OUTPUT} & \left\langle \text{F}_{PAST}(\boxed{3})\ , \begin{bmatrix} \text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{FORM} & \text{fin}\end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{RESTR} & \boxed{A} \oplus \ldots\end{bmatrix} \\ \text{ARG-ST} & \left\langle \begin{bmatrix}\text{CASE} & \text{nom}\end{bmatrix}, \ldots \right\rangle \end{bmatrix}\right\rangle \end{bmatrix}$$

(72) makes use of a function F_{PAST} to account for the morphological relation between verbal lexemes and their past tense forms; in most cases, this consists of suffixing *-ed*, though there are many exceptions (such as *sleep/slept*, *eat/ate*, and *put/put*).

Like the lexical rules for present tense verbs, (72) requires its subject to be nominative (to rule out examples like **Me slept*); but unlike the present tense rules, it puts no number or person restrictions on the subject, since English past tense verbs exhibit no agreement with their subjects. The semantic effect of the rule is parallel to that of the two present tense rules, though the required semantics is different.[35]

[34] Of course, something must be said about this exception and about the first-person singular form *am*. The fact that *be* makes finer distinctions among its verb forms than other verbs does not justify making these distinctions throughout the rest of the verbal system in English. Rather, it is more parsimonious to make *be* an exception to some of these lexical rules, and to stipulate the individual forms in the lexicon or to posit highly specialized lexical rules for the forms of *be*. (The latter course may be desirable because, as we shall see at several points in the rest of this book, there appear to be several different *be* lexemes in English). We will not go into the question of what kind of formal machinery to use to specify that particular lexical entries are exceptions to certain lexical rules, though some such mechanism is surely needed irrespective of *be*.

The inflectional paradigm of *be* looks quite confusing at first, with one form (*am*) that goes only with first-person subjects and others (*are, were*) that go only with subjects that are second-person or plural. The situation looks a bit less arbitrary if we make use of the hierarchy of subtypes of *non-3sing* introduced in Chapter 4. That hierarchy makes available a type *1sing* that is the AGR value we need for *am*. It also provides a type *non-1sing* encompassing just second-person and plural AGR values (that is, it excludes just the first-person singular and third-person singular values). This is precisely the AGR value we need for *are* and *were*. The AGR value of *was* needs to be consistent with both *1sing* and *3sing*, but nothing else. There is no appropriate type in our current hierarchy (although there could be with multiple inheritance – see Chapter 16), but there are two related solutions: a disjunctive AGR value, or two separate lexical entries (alternatively, two separate lexical rules), one specifying [AGR *1sing*] and one specifying [AGR *3sing*].

[35] In the same spirit as the representation of present tensed sketched in note 33, we could represent past tense by adding a 'temporal precedence' predication to the RESTR value. That is, the situation referred to by the index of the verb temporally precedes the time of utterance if the verb is in the past tense. Again, this is only a first approximation of the semantics of English past tense forms, which sometimes

8.7.3 Uninflected Words

Finally, we need a trivial lexical rule for noninflecting lexemes:

(73) Constant Lexeme Lexical Rule

$$\begin{bmatrix} \textit{i-rule} & \\ \text{INPUT} & \left\langle \boxed{1}\ , \textit{const-lxm} \right\rangle \\ \text{OUTPUT} & \left\langle \boxed{1}\ , \text{X} \right\rangle \end{bmatrix}$$

This rule does nothing except allow the requisite words to be licensed from homophonous lexemes. The SYN, SEM and ARG-ST values of these words will be identical to those of the corresponding lexeme. This already follows from the inheritance of the identity constraints in (59) and (60). As *word*s, the OUTPUTs will be subject to the ARP.

8.7.4 A Final Note on Inflectional Rules

Despite the metaphor suggested by the feature names INPUT and OUTPUT, and the informal procedural language we use to describe them, lexical rules do not change or otherwise operate on lexical sequences. Rather they relate lexical sequences to other lexical sequences. They also act in some sense as filters: Our lexical entries are relatively underspecified descriptions, and as such, license many lexical sequences with somewhat surprising feature specifications. For example, because the ARP applies only to *word*s and not *lexeme*s, the lexical entry in (32) licenses lexical sequences that meet the description in (74):

(74) A lexical sequence that doesn't give rise to any words

$$\left\langle \text{dog}\ , \begin{bmatrix} \textit{cntn-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} \\ \text{AGR} \quad \boxed{1}\begin{bmatrix}\text{PER 3rd}\end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \text{ NP[AGR } \boxed{1}] \rangle \\ \text{COMPS} & \langle \text{ NP, NP, VP, NP } \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{dog} \\ \text{INST} & i \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{ARG-ST} & \left\langle \begin{matrix} \text{DP} \\ [\text{COUNT} \quad +] \end{matrix} \right\rangle \end{bmatrix} \right\rangle$$

Such lexical sequences of course need to be barred from licensing bizarre trees, and this work is done by the lexical rules. The input value of the Singular Noun Lexical Rule, for example, could never be resolved to one of the lexical sequences depicted in (74). This is because the output value of that lexical rule contains a *word*, which is subject to

are used to describe future or unrealized actions.

the ARP. Furthermore, the SYN and ARG-ST values of the INPUT and the OUTPUT are identified, which means that the INPUT will always, as a side-effect, also obey the ARP, and crazy lexical sequences like (74) won't be related to any well-formed lexical sequences with feature structures of type *word*.

8.8 Derivational Rules

Each of the lexical rules in the previous section maps lexical sequences of type *lexeme* into seqences of type *word*. We have followed tradition in calling these INFLECTIONAL rules. It is also traditional to distinguish these from another kind of lexical rule (called a DERIVATIONAL rule) that relates lexemes to lexemes (or, in our system, lexical sequences of the appropriate kind to other such lexical sequences). Derivational rules (*d-rules*) are appropriate when the addition of a prefix or suffix creates a new lexical sequence that can itself undergo inflectional rules.[36] We will assume that *d-rules* are constrained as follows:

(75)
$$\textit{d-rule}: \begin{bmatrix} \text{INPUT} & \left\langle \text{X}\ , \begin{bmatrix} \textit{lexeme} & \\ \text{SYN} & /\ \boxed{3} \end{bmatrix} \right\rangle \\ \\ \text{OUTPUT} & \left\langle \text{Y}\ , \begin{bmatrix} \textit{lexeme} & \\ \text{SYN} & /\ \boxed{3} \end{bmatrix} \right\rangle \end{bmatrix}$$

Let us consider agentive nominalizations as a first example. Noun lexemes like *driver* or *eater* might be derived by the following lexical rule:

(76) Agent Nominalization Lexical Rule

$$\begin{bmatrix} \textit{d-rule} & \\ \\ \text{INPUT} & \left\langle \boxed{2}\ , \begin{bmatrix} \textit{stv-lxm} & \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \end{bmatrix} \\ \text{ARG-ST} & \langle\ \text{X}_i\ ,\ \text{NP}_j\ \rangle \end{bmatrix} \right\rangle \\ \\ \text{OUTPUT} & \left\langle \text{F}_{-er}(\boxed{2})\ , \begin{bmatrix} \textit{cntn-lxm} & \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & i \end{bmatrix} \\ \text{ARG-ST} & \left\langle \text{Y} \left(, \begin{matrix} \text{PP}_j \\ \begin{bmatrix} \text{FORM} & \text{of} \end{bmatrix} \end{matrix} \right) \right\rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

Here the function F_{-er} adds the appropriate suffix to the form of the rule output. The input involves a verbal lexeme whose subject's index i is identified with the index of the nominal output. Note that the change in type from *verb-lxm* to *cntn-lxm* has many side effects in terms of values of head features and in terms of the MODE value within the semantics. However, the RESTR value remains unchanged, as the information present in the input is compatible with the type constraints associated with the output type.

[36] There are also derivational rules that have no phonological effect. See (79) below.

The ARG-ST values in (76) deserve some comment. The input must be a strictly transitive verb.[37] Thus we correctly rule out agent nominals of such verbs as *rely* or *put*:

(77) a. *the relier (on Sandy)
b. *the putter (of books) (on the table)

The output, like other common nouns, takes a determiner. In addition, the output's SPR value (and hence the first member of the ARG-ST list (Y)) will be a [COUNT +] determiner, according to constraints on the type *cntn-lxm*. And the agent nominal may take a PP complement whose object is identified with the object of the input verb. This is for agent nominals such as *the discoverer of oxygen* and *a builder of bridges*.[38]

Consider, for example, the lexical entry for the verbal lexeme *drive*, the semantics of which is a proposition whose RESTR value contains a **drive** predication, with the role of driver assigned to the referent of the verb's subject. Applying the Agent Nominalization Lexical Rule to this entry yields a family of lexical sequences whose first member is the form *driver* and whose index is restricted to be the driver in a driving predication (since the RESTR value is unchanged):

(78)
$$\left\langle \text{driver}\ ,\ \begin{bmatrix} \textit{cntn-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \boxed{1}[\text{PER}\ \ \text{3rd}] \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle [\text{AGR}\ \ \boxed{1}] \right\rangle \end{bmatrix} \\ \text{ARG-ST} & \left\langle \text{X}_i\ (,\ \text{PP}[\textit{of}]_j) \right\rangle \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{drive} \\ \text{DRIVER} & i \\ \text{DRIVEN} & j \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{ARG-ST} & \left\langle \begin{matrix} \text{DP} \\ [\text{COUNT}\ \ +] \end{matrix} \right\rangle \end{bmatrix} \right\rangle$$

These lexical sequences can now undergo both our nominal lexical rules, and so we derive two new families of lexical sequences: one for the singular noun word *driver* and one for its plural analog *drivers*.

There are further semantic constraints that must be placed on our derivational rule, however. For example, the subject in the input verb has to be sufficiently agentive – that is, it must play an active (usually volitional) role in the situation. That's why nominalizations like *knower* or *resembler* sound funny. But the formulation in (78) is a reasonable first pass at the problem, and it gives you an idea of how phenomena like this can be analyzed within our framework.

[37] We provide no account here of intransitive agentive nouns like *jumper*, *runner*, *diver*, etc.

[38] Notice that in formulating this rule, we have used the FORM value 'of' to indicate that the preposition heading this PP must be *of*. We return to the matter of FORM values for prepositions in Chapter 10.

There are many other cross-categorial relations that work this way in English. Noun lexemes, both common and proper, can be converted into verbal lexemes:[39]

(79) a. Sandy *porched* the newspaper without difficulty.
b. The senator *houdinied* his way out of the accusations.
c. They have been *computering* me to death all morning.

This kind of derivation without morphological change, an instance of what is often called ZERO DERIVATION, could be handled by one or more derivational rules.

Derivational rules are also a traditional way of approaching the problem of valence alternations, that is, the fact that many verbs allow systematically related valence patterns. Among the most famous of these is the dative alternation illustrated in (80) – (81):

(80) a. Jan gave Dale a book.
b. Jan gave a book to Dale.

(81) a. Jan handed Dale a book.
b. Jan handed a book to Dale.

Rather than list entries for two distinct verbal lexemes for *give*, *hand*, and a family of related elements, it makes much more sense to list only one (with one of the two valence patterns fixed) and to derive the other by a derivational rule. Note however, that there are certain other verbs or particular idiomatic uses that appear in only one of the two valence patterns:

(82) a. Kris donated a book to the library.
b. *Kris donated the library a book.

(83) a. Dale gave Brooke a hard time.
b. ??Dale gave a hard time to Brooke.

These underline once again the need for a theory of lexical irregularity and exceptions to lexical rules.

Other famous examples of valence alternation are illustrated in (84)–(88).

(84) a. The police sprayed the protesters with water.
b. The police sprayed water on the protesters. ('*spray/load*' alternations)

(85) a. The students drove cars.
b. These cars drive easily. ('middle' uses)

(86) a. Pat sneezed.
b. Pat sneezed the napkin off the table. ('caused motion' uses)

(87) a. The horse kicked me.
b. The horse kicked me black and blue. ('resultative' uses)

(88) a. They yelled.
b. They yelled their way into the meeting. (the 'X's way' construction)

[39] For more on the topic of English noun-verb conversions, see Clark and Clark 1979.

All these patterns of valence alternation are governed by both semantic and syntactic constraints of the kind that could be described by finely tuned lexical rules.

Finally, we will use derivational rules to treat verbal participles like those illustrated in (89) (and discussed in Section 8.5):

(89) a. Kim is *standing* here.
b. Sandy has *eaten* dinner.

The *d-rule*s we need are formulated as follows:

(90) Present Participle Lexical Rule

$$\begin{bmatrix} \textit{d-rule} & \\ \text{INPUT} & \left\langle \boxed{3}\ ,\ \begin{bmatrix} \textit{verb-lxm} & \\ \text{SEM} & \begin{bmatrix}\text{RESTR} & \boxed{A}\end{bmatrix} \\ \text{ARG-ST} & \boxed{B} \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \text{F}_{PRP}(\boxed{3})\ ,\ \begin{bmatrix} \textit{part-lxm} & \\ \text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{FORM} & \text{prp}\end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{RESTR} & \boxed{A} \oplus \ldots.\end{bmatrix} \\ \text{ARG-ST} & \boxed{B} \end{bmatrix} \right\rangle \end{bmatrix}$$

(91) Past Participle Lexical Rule

$$\begin{bmatrix} \textit{d-rule} & \\ \text{INPUT} & \left\langle \boxed{3}\ ,\ \begin{bmatrix} \textit{verb-lxm} & \\ \text{SEM} & \begin{bmatrix}\text{RESTR} & \boxed{A}\end{bmatrix} \\ \text{ARG-ST} & \boxed{B} \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \text{F}_{PSP}(\boxed{3})\ ,\ \begin{bmatrix} \textit{part-lxm} & \\ \text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{FORM} & \text{psp}\end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{RESTR} & \boxed{A} \oplus \ldots.\end{bmatrix} \\ \text{ARG-ST} & \boxed{B} \end{bmatrix} \right\rangle \end{bmatrix}$$

Note that the outputs of these rules belong to the type *participle-lexeme* (*part-lxm*), which is a subtype of *const-lxm* in our grammar. Thus participles undergo no further morphological processes. This is, in essence, an arbitrary fact of English, as participles do undergo inflection in other Indo-European languages, for example in French:

(92) a. Il y est allé.
he there is gone-M.SG
'He went there.'

b. Ils y sont allés.
they there are gone-M.PL
'They(masc.) went there.'

c. Elle y est allée.
she there is gone-F.SG
'She went there.'

d. Elles y sont allées.
they there are gone-F.PL
'They(fem.) went there.'

Such examples show that the lexical rule for past participles in French must be derivational (that is, lexeme-to-lexeme); otherwise, participles could not serve as inputs to the inflectional rules responsible for the agreement suffixes. Our formulation of the English participle rules as derivational minimizes the differences between the grammars of English and French in this regard.[40]

In Chapter 10, we will extend our account of participle-lexemes to include passive participles as well.

8.9 Summary

An important insight, going back at least to Saussure, is that all languages involve arbitrary (that is, unpredictable) information. Most clearly, the association between the forms (sounds) and meanings of words is purely conventional, in the vast majority of cases. A grammar of a language must list these associations somewhere. The original conception of the lexicon in modern linguistics was simply as the repository of such arbitrary information.

This conception did not last long, however. Beginning in the early years of transformational grammar, linguists began enriching their conception of the lexicon to include information that was not idiosyncratic to individual words. This trend continued in a great deal of research carried out within a variety of grammatical frameworks.

In this text, we have to some extent recapitulated this history. We began with context-free grammar in which the lexicon contained only idiosyncratic information, and we gradually enriched our lexical representations, including more and more information – much of it systematic and predictable – about the grammatical and semantic properties of words. Indeed, most of the information needed to determine the well-formedness of sentences is now encoded in our lexical entries.

With the increased expressiveness and concomitant complexity of lexical entries came a need to express succinctly certain generalizations about words. In this chapter, we have examined two formal mechanisms for capturing such generalizations. Structuring the lexicon as a hierarchy of types through which constraints are inherited (an innovation of the mid-1980s) has made it possible to factor out information common to many lexical entries, thereby greatly reducing lexical redundancy. By allowing certain type constraints to be defeasible, we have encoded default values for features, while still allowing for lexical idiosyncrasy. The second mechanism, the lexical rule, is an older idea, going back to work in transformational grammar of the 1970s. We will make considerable use of lexical rules in subsequent chapters. In fact, many of the phenomena that provided the motivation

[40]We know of no evidence strictly from English for choosing between a derivational formulation and an inflectional formulation of the past and present participle rules. Similarly, base forms of verbs could be derived either by derivational or inflectional rule (but some lexical rule is required).

for transformations in the 1950s and 1960s can be reanalyzed in our theory using lexical rules. These include the passive construction – the topic of Chapter 10 – and many of the properties of the English auxiliary verb system, which we treat in Chapter 13.

8.10 Further Reading

An important early paper on lexical rules is Jackendoff 1975. The idea of combining lexical rules with an inheritance hierarchy was first put forward by Flickinger et al. (1985). See also Pollard and Sag 1987, Chapter 8, and Meurers 1999, 2001. Briscoe et al. 1993 is a collection of papers about lexical hierarchies, default inheritance, and related issues. The approach to lexical rules presented here draws heavily on Copestake 1992 and Briscoe and Copestake 1999. A standard reference on lexical classes and subcategorizational alternations is Levin 1993. Goldberg (1995) provides a Construction Grammar analysis of many of the valence alternations discussed at the end of this chapter.

8.11 Problems

Problem 1: *'s* and the SHAC

The name 'Specifier-Head Agreement Constraint' suggests that heads always agree with their specifiers. Examples like *Pat's parents* and *the children's game* look like counterexamples: in both cases, the possessive NP in the DP that functions as the specifier of the noun differs in number from that noun. Explain why these are not really counterexamples, given our formulation of SHAC as a type constraint, together with the analysis of possessives developed in Problem 4 of Chapter 6. *[Hint: The fact that 's is the head of the DP is crucial.]*

⚠ **Problem 2: Plural and Mass NPs Without Specifiers**

There is a problem with our treatment of common nouns. The type *cn-lxm* requires common nouns to have nonempty SPR lists, and this requirement is preserved in the Plural Noun Lexical Rule. Similarly, the type *massn-lxm* inherits the constraint on the SPR, and this constraint is preserved when these nouns undergo the inflectional rules. This treatment makes the wrong predictions: specifiers are optional for plural nouns and mass nouns.

A. Give examples showing, for one plural noun and one mass noun, that the specifier is optional (i.e. permitted but not obligatory).

Two obvious approaches to this problem are the following:

(i) allow empty SPR lists in the lexical entries for plural and mass nouns; or

(ii) introduce a new grammar rule to account for NPs with plural or mass heads and no specifiers.

Alternative (i) would involve modifying the Plural Noun Lexical Rule, as well as the type *massn-lxm* to make the first member of the ARG-ST list optional.[41]

[41] This would require making the constraint on the ARG-ST of *cn-lxm* defeasible.

The rule in alternative (ii) is analogous to the Imperative Rule given in Chapter 7, in that it would have only one constituent on the right hand side, and its function would be to license a constituent without a specifier, although its daughter has a nonempty SPR list.

It turns out that alternative (i) makes incorrect predictions about prenominal modifiers (see Problem 1 of Chapter 5). We want adjectives like *cute* to modify plural nouns even when they don't have specifiers:

(iii) Cute puppies make people happy.

Under alternative (i), in order to generate (iii), we would have to allow adjectives like *cute* to modify NPs (i.e. expressions that are [SPR $\langle\ \rangle$]). If we do that, however, we have no way to block (iv):[42]

(iv)*Cute the puppies make people happy.

Alternative (ii), on the other hand, would allow *cute* to always modify a NOM ([SPR $\langle$ DP $\rangle$]) constituent. A NOM, modified or otherwise, could either be the daughter of the non-branching rule, or the head daughter of the Head-Specifier Rule.

B. Formulate the rule required for alternative (ii).

[*Hint: The trickiest part is formulating the rule so that it applies to both plural count nouns and mass nouns, while not applying to singular count nouns. You will need to include a disjunction in the rule. The SPR list of the head daughter is a good place to state it, since the three types of nouns differ in the requirements they place on their specifiers.*]

Problem 3: *-s*

In most cases, F_{3SG} has the same effect as F_{NPL}, namely, that of suffixing *-s*. In fact, both suffixes have multiple pronunciations, and the conditions under which they are pronounced like *s*, like *z*, or like *iz* are identical. (They depend on phonological properties of the preceding sound.) Nevertheless, these two morphological functions are not identical. Why?

[*Hints: 1. Remember that a function is single-valued, i.e. it specifies only one output for each input. 2. Consider elements that can be used as both nouns and verbs.*]

Problem 4: Coordination and Tense

For the most part, the inflectional rules for verbs stand in a one-to-one relationship with FORM values. The exceptions are the 3rd-Singular, Non-3rd-Singular, and Past-Tense Verb Lexical Rules, all of which produce outputs that are [FORM fin]. The alternative would be to posit a distinct FORM value for each rule: say, '3sg_present', 'non3sg_present' and 'past', or at least two different forms 'present' and 'past'. Making reference to the discussion of FORM and coordination in Section 8.5.2, explain why the decision to use just one FORM value ('fin') is right or wrong. Be sure to consider examples where finite VPs that differ in tense are coordinated.

[42] There are also technical problems with making alternative (i) work with the ARP.

Problem 5: Conjoined Conjunctions

A. Does our grammar license the (ungrammatical) string in (i)? (Assume lexical entries for *and*, *but* and *or* that are all [HEAD *conj*].)
   (i) Kim left and but or or and Sandy stayed.

B. If you answered 'yes' to part (A), draw a tree showing a structure that the grammar licenses for the sentence. (Abbreviated node labels are fine.) If you answered 'no' to part (A), explain how it is ruled out.

Problem 6: Arguments in Japanese

As noted in Chapter 2, Japanese word order differs from English in a number of ways, including the fact that it is a 'Subject-Object-Verb' (SOV) language. Here are a few relevant examples. In the glosses, 'NOM', 'ACC', and 'DAT' stand for nominative, accusative, and dative case, respectively. (Note that Japanese has one more case – dative – than English does. This doesn't have any important effects on the analysis; it merely requires that we posit one more possible value of CASE for Japanese than for English).[43]

(i) Hitorino otoko-ga sono hon-o yonda.
one man-NOM that book-ACC read.PAST
'One man read that book.'
[cf. *Yonda hitorino otoko-ga sono hon-o.
*Hitorino otoko-ga yonda sono hon-o.
*Otoko-ga hitorino sono hon-o yonda.
*Hitorino otoko-ga hon-o sono yonda.
*Hitorino otoko-ni/-o sono hon-o yonda.
*Hitorino otoko-ga sono hon-ga/-ni yonda.]

(ii) Hanako-ga hon-o yonda
Hanako-NOM book-ACC read.PAST
'Hanako read the book(s)'
[cf. *Yonda Hanako-ga hon-o.
*Hanako-ga yonda hon-o.
*Hanako-ni/-o hon-o yonda.
*Hanako-ga hon-ni/-ga yonda.]

(iii) sensei-ga Taroo-ni sono hon-o ageta
teacher-NOM Taroo-DAT that book-ACC gave.PAST
'The teacher(s) gave that book to Taroo'
[cf. *Ageta sensei-ga Taroo-ni sono hon-o.
*Sensei-ga ageta Taroo-ni sono hon-o.
*Sensei-ga Taroo-ni ageta sono hon-o.
*Sensei-o/-ni Taroo-ni sono hon-o ageta.

[43] The examples marked with '*' here are unacceptable with the indicated meanings. Some of these might be well-formed with some other meaning of no direct relevance; others might be well-formed with special intonation that we will ignore for present purposes.

*Sensei-ga Taroo-ga/-o sono hon-o ageta.
*Sensei-ga Taroo-ni sono hon-ga/-ni ageta.]

(iv) Hanako-ga kita
Hanako-NOM arrive.PAST
'Hanako arrived.'
[cf. *Kita Hanako-ga.]

As the contrasting ungrammatical examples show, the verb must appear in final position in Japanese. In addition, we see that verbs select for NPs of a particular case, much as in English. In the following tasks, assume that the nouns and verbs of Japanese are inflected words, derived by lexical rule from the appropriate lexemes.

A. Write Head-Specifier and Head-Complement Rules for Japanese that account for the data illustrated here. How are they different (if at all) from the Head-Specifier and Head-Complement Rules for English?
B. Give the lexical entry for each of the verbs illustrated in (i)–(iv).
[*Notes: Make sure your entries interact with the rules you formulated in part (A) to account for the above data. The data given permit you to specify only some features; leave others unspecified. Assume that there is a Past-Tense Verb Lexical Rule (an i-rule) that relates your lexical entries to the words shown in (i)–(iv). We have not provided a hierarchy of lexeme types for Japanese. You may either give all relevant constraints directly on the lexical entries, or posit and use subtypes of lexeme. In the latter case, you must also provide those types.*]
C. Give the lexical entries for the nouns *Taroo* and *hon.*
[*Note: See notes on part (B).*]
D. Formulate the lexical rule for deriving the inflected forms ending in *-o* from the nominal lexemes.

Problem 7: Japanese Causatives

Crosslinguistically, causative constructions like (i) can be either PERIPHRASTIC or MORPHOLOGICAL. In a periphrastic causative (such as (i)), a separate word (typically a verb) expresses the causation and licenses or selects for the causer argument. In a morphological causative, such as the Japanese example in (iii), the causation is expressed by an affix and the verb's valence is augmented by one.

(i) Kim made Sandy eat the cake.

(ii) Suzuki-ga keeki-o tabeta
Suzuki-NOM cake-ACC eat.PAST
'Suzuki ate the cake.'

(iii) Aoki-ga Suzuki-ni keeki-o tabesaseta
Aoki-NOM Suzuki-DAT cake-ACC eat.CAUSE.PAST
'Aoki made Suzuki eat the cake.'
[cf. *Aoki-ga Suzuki-ni keeki-o tabeta.]

A. What is the case of the CAUSER argument in (iii)?
B. What is the case of the CAUSEE argument in (iii)?

C. Assume that the relevant lexical sequence for *tabeta* in (ii) is as in (iv) and that the semantics of the relevant lexical sequence for *tabesaseta* in (iii) is as in (v).[44] Write a lexical sequence for *tabesaseta* in (iii).

(iv)
$$\left\langle \text{tabeta}\ ,\ \left[\begin{array}{ll}
\textit{word} & \\
\text{SYN} & \left[\begin{array}{ll}
\text{HEAD} & \left[\begin{array}{ll} \textit{verb} & \\ \text{FORM} & \text{fin} \end{array}\right] \\
\text{VAL} & \left[\begin{array}{ll} \text{SPR} & \langle\, \boxed{1}\, \rangle \\ \text{COMPS} & \langle\, \boxed{2}\, \rangle \end{array}\right]
\end{array}\right] \\
\text{ARG-ST} & \langle\, \boxed{1}\text{NP[CASE nom]}_i\ ,\ \boxed{2}\text{NP[CASE acc]}_j\, \rangle \\
\text{SEM} & \left[\begin{array}{ll}
\text{INDEX} & s_1 \\
\text{MODE} & \text{prop} \\
\text{RESTR} & \left\langle \left[\begin{array}{ll} \text{RELN} & \textbf{eat} \\ \text{SIT} & s_1 \\ \text{EATER} & i \\ \text{MEAL} & j \end{array}\right], \ldots \right\rangle
\end{array}\right]
\end{array}\right] \right\rangle$$

(v)
$$\left[\begin{array}{ll}
\text{INDEX} & s_2 \\
\text{MODE} & \text{prop} \\
\text{RESTR} & \left\langle \left[\begin{array}{ll} \text{RELN} & \textbf{eat} \\ \text{SIT} & s_1 \\ \text{EATER} & i \\ \text{MEAL} & j \end{array}\right], \left[\begin{array}{ll} \text{RELN} & \textbf{cause} \\ \text{SIT} & s_2 \\ \text{CAUSER} & k \\ \text{CAUSEE} & i \\ \text{CAUSED-EVENT} & s_1 \end{array}\right], \ldots \right\rangle
\end{array}\right]$$

D. Write a Causative Lexical Rule for Japanese that will derive lexemes like *tabesase-* from lexemes like *tabe-*. [*Notes: Tabesase- and tabe- are the stem forms for tabesaseta and tabeta respectively. That is, they are the forms that are input to the Past-Tense Verb Lexical Rule. Be sure to make your Causative Lexical Rule a derivational rule. Since we haven't defined a hierarchy of lexeme types for Japanese, assume that the second members of the INPUT and OUTPUT of your rule are simply of type lexeme. You'll need to find some other way to restrict the INPUT of the rule to verbal lexemes.*]

---

[44] The '...' in the RESTR lists indicate that there should be something more in these lexical sequences, namely, a representation of the semantics of past tense.

9

Realistic Grammar

9.1 Introduction

In the preceding eight chapters, we have laid out the theory that we will apply to more complex data in the remainder of this book. The theoretical machinery we have developed so far permits us to provide accounts of a rich array of syntactic phenomena that we will examine in Chapters 10-13, specifically, the English passive construction, existential sentences introduced by *there*, subordinate clauses introduced by *that*, a nonreferential use of *it*, the behavior of NPs that are parts of idioms, four types of constructions involving infinitival VPs, sentential negation and reaffirmation, inversion of the auxiliary verb in questions, negative auxiliaries (ending in *-n't*), and elliptical VPs (that is, VPs missing everything but their auxiliary verb). Coverage of these phenomena will require additions to the lexicon, including changes to the lexical type hierarchy, new lexical rules, some new features, and, of course, new lexical entries. But our grammar rules and principles will remain essentially unchanged until Chapter 14, when we address the topic of long-distance dependencies, a complex set of phenomena that will require the addition of a new grammar rule and a new principle, along with a number of modifications to the rules and principles we have seen so far.

Before we proceed, however, it is useful to consolidate the components of our treatment of English grammar and to reflect on the strategy we have adopted for solving syntactic problems – to reflect on the motivation for the design of grammar.

As we noted briefly in Chapter 2, syntacticians rely heavily on considerations of parsimony: the desirability of 'capturing generalizations' is given great weight in choosing between analyses. This concern with providing elegant descriptions is not unique to this field, though it probably figures more prominently in linguistic argumentation than elsewhere. It is natural to ask, however, whether a grammar whose design has been shaped in large measure by concern for parsimony corresponds straightforwardly to the way linguistic knowledge is represented in the minds of language users. We argue in this chapter that the available psycholinguistic evidence fits rather well with the conception of grammar that we have been developing in this book.

First, however, we turn to a summary of our grammar to date. The next section of this chapter gives a formal presentation of everything we have covered so far, including types, lexical entries, grammar rules, the well-formedness definitions (incorporating various principles), and lexical rules.

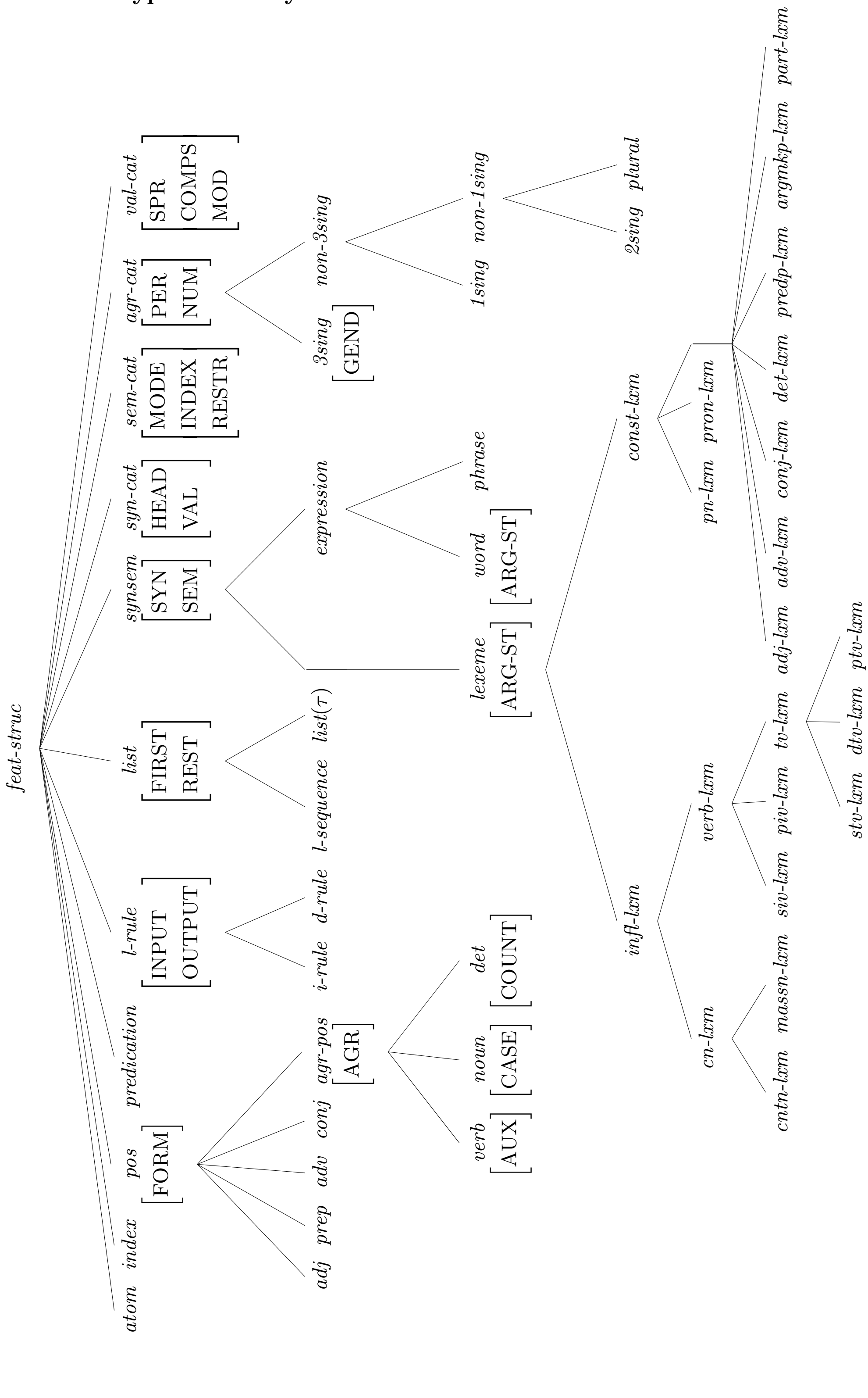
Section 9.2.1 presents the type hierarchy, and Section 9.2.2 gives the feature declaractions and type constraints. Almost all of the types and constraints listed in Section 9.2.2 have been introduced in earlier chapters. We have added little that is new. Section 9.2.3 gives the definitions of the abbreviations we use. These have not changed since Chapter 5. Section 9.2.4 lists our familiar grammar rules from Chapter 5, together with the Imperative Rule introduced in Chapter 7. Section 9.2.5 lists the lexical rules that were presented in Chapter 8. Section 9.2.6 gives some sample lexical entries. It is worth noting that most of what we have to stipulate in our entries is semantic. By virtue of having a richly structured lexicon, we are able to limit the amount of syntactic information that has to be listed in individual entries, thereby greatly reducing redundant stipulation. Section 9.2.7 gives the formal definitions of well-formed tree structure and lexical and phrasal satisfaction, incorporating all of the general principles of grammar we have adopted so far. This version is slightly modified from the one given in Chapter 6, in that the definition of lexical licensing now takes lexical rules into account. In addition, our Binding Theory, the Case Constraint, and the Anaphoric Agreement Principle have been built in.

9.2 The Grammar So Far

The following pages contain a summary of the type hierarchy developed in the preceding chapters:[1]

[1]We use the notation 'list(τ)' to indicate a (possibly empty) list, all of whose members are of type τ.

9.2.1 The Type Hierarchy



9.2.2 Feature Declarations and Type Constraints

GENERAL TYPES		
TYPE	FEATURES/CONSTRAINTS	IST
feat-struc		
atom		*feat-struc*
index		*feat-struc*
l-rule	$\begin{bmatrix} \text{INPUT} & \textit{l-sequence}\left\langle \text{X}\ ,\ \begin{bmatrix}\text{SEM}\ /\ \boxed{2}\end{bmatrix}\right\rangle \\ \text{OUTPUT} & \textit{l-sequence}\left\langle \text{Y}\ ,\ \begin{bmatrix}\text{SEM}\ /\ \boxed{2}\end{bmatrix}\right\rangle \end{bmatrix}$	*feat-struc*
i-rule	$\begin{bmatrix} \text{INPUT} & \left\langle \text{X}\ ,\ \begin{bmatrix}\textit{lexeme} & \\ \text{SYN} & \boxed{3} \\ \text{ARG-ST} & \boxed{\text{A}}\end{bmatrix}\right\rangle \\ \text{OUTPUT} & \left\langle \text{Y}\ ,\ \begin{bmatrix}\textit{word} & \\ \text{SYN} & \boxed{3} \\ \text{ARG-ST} & \boxed{\text{A}}\end{bmatrix}\right\rangle \end{bmatrix}$	*l-rule*
d-rule	$\begin{bmatrix} \text{INPUT} & \left\langle \text{X}\ ,\ \begin{bmatrix}\textit{lexeme} & \\ \text{SYN} & /\ \boxed{3}\end{bmatrix}\right\rangle \\ \text{OUTPUT} & \left\langle \text{Y}\ ,\ \begin{bmatrix}\textit{lexeme} & \\ \text{SYN} & /\ \boxed{3}\end{bmatrix}\right\rangle \end{bmatrix}$	*l-rule*
list		*feat-struc*
list(τ)	$\begin{bmatrix} \text{FIRST} & \tau \\ \text{REST} & \textit{list}(\tau) \end{bmatrix}$	*list*
l-sequence	$\begin{bmatrix} \text{FIRST} & \textit{atom} \\ \text{REST} & \langle \textit{word} \rangle \mid \langle \textit{lexeme} \rangle \end{bmatrix}$	*list*
synsem	$\begin{bmatrix} \text{SYN} & \textit{syn-cat} \\ \text{SEM} & \textit{sem-cat} \end{bmatrix}$	*feat-struc*

GENERAL TYPES (CONTINUED)		
TYPE	FEATURES/CONSTRAINTS	IST
syn-cat	$\begin{bmatrix} \text{HEAD} & \textit{pos} \\ \text{VAL} & \textit{val-cat} \end{bmatrix}$	*feat-struc*
sem-cat	$\begin{bmatrix} \text{MODE} & \left\{\text{prop, ques, dir, ref, ana, none}\right\} \\ \text{INDEX} & \textit{index} \\ \text{RESTR} & \textit{list}(\textit{predication}) \end{bmatrix}$	*feat-struc*
val-cat	$\begin{bmatrix} \text{SPR} & \textit{list}(\textit{expression}) \\ \text{COMPS} & \textit{list}(\textit{expression}) \\ \text{MOD} & \textit{list}(\textit{expression}) \end{bmatrix}$	*feat-struc*
expression		*synsem*
phrase		*expression*
word	$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{SPR} & \boxed{\text{A}} \\ \text{COMPS} & \boxed{\text{B}} \end{bmatrix} \\ \text{ARG-ST} & \boxed{\text{A}} \oplus \boxed{\text{B}} \end{bmatrix}$	*expression*
lexeme	$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{MOD} & / \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \textit{list}(\textit{expression}) \end{bmatrix}$	*synsem*
infl-lxm	$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \rangle \end{bmatrix} \end{bmatrix}$	*lexeme*
const-lxm		*lexeme*
cn-lxm	$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \begin{bmatrix} \text{PER 3rd} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & / \text{ ref} \\ \text{INDEX} & i \end{bmatrix} \\ \text{ARG-ST} & \begin{bmatrix} \text{FIRST} & \text{DP}_i \\ \text{REST} & / \langle\ \rangle \end{bmatrix} \end{bmatrix}$	*infl-lxm*

LEXEME TYPES		
TYPE	FEATURES/CONSTRAINTS	IST
verb-lxm	$\begin{bmatrix} \text{SYN} & \begin{bmatrix}\text{HEAD} & \textit{verb}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{MODE} & \text{prop}\end{bmatrix} \\ \text{ARG-ST} & \langle \text{ NP , ... } \rangle \end{bmatrix}$	*infl-lxm*
cntn-lxm	$\begin{bmatrix}\text{ARG-ST} & \langle \text{ [COUNT +] , ... } \rangle\end{bmatrix}$	*cn-lxm*
massn-lxm	$\begin{bmatrix}\text{ARG-ST} & \langle \text{ [COUNT } - \text{] , ... } \rangle\end{bmatrix}$	*cn-lxm*
siv-lxm	$\begin{bmatrix}\text{ARG-ST} & \langle \text{ X } \rangle\end{bmatrix}$	*verb-lxm*
piv-lxm	$\begin{bmatrix}\text{ARG-ST} & \langle \text{ X , PP } \rangle\end{bmatrix}$	*verb-lxm*
tv-lxm	$\begin{bmatrix}\text{ARG-ST} & \langle \text{ X , NP , ... } \rangle\end{bmatrix}$	*verb-lxm*
stv-lxm	$\begin{bmatrix}\text{ARG-ST} & \langle \text{ X , Y } \rangle\end{bmatrix}$	*tv-lxm*
dtv-lxm	$\begin{bmatrix}\text{ARG-ST} & \langle \text{ X , Y , NP } \rangle\end{bmatrix}$	*tv-lxm*
ptv-lxm	$\begin{bmatrix}\text{ARG-ST} & \langle \text{ X , Y , PP } \rangle\end{bmatrix}$	*tv-lxm*
pn-lxm	$\begin{bmatrix} \text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\textit{noun} & \\ \text{AGR} & \begin{bmatrix}\text{PER} & \text{3rd} \\ \text{NUM} & / \text{ sg}\end{bmatrix}\end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{MODE} & \text{ref}\end{bmatrix} \\ \text{ARG-ST} & / \langle \; \rangle \end{bmatrix}$	*const-lxm*
pron-lxm	$\begin{bmatrix} \text{SYN} & \begin{bmatrix}\text{HEAD} & \textit{noun}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{MODE} & / \text{ ref}\end{bmatrix} \\ \text{ARG-ST} & \langle \; \rangle \end{bmatrix}$	*const-lxm*

LEXEME TYPES (CONTINUED)		
TYPE	FEATURES/CONSTRAINTS	IST
conj-lxm	$\begin{bmatrix} \text{SYN} & [\text{HEAD} \;\; \textit{conj}] \\ \text{SEM} & [\text{MODE} \;\; \text{none}] \\ \text{ARG-ST} & \langle \; \rangle \end{bmatrix}$	*const-lxm*
adj-lxm	$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{adj} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \text{ X } \rangle \\ \text{MOD} & \langle [\text{HEAD} \;\; \textit{noun}] \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & [\text{MODE} \;\; \text{prop}] \\ \text{ARG-ST} & \langle \text{ NP , ... } \rangle \end{bmatrix}$	*const-lxm*
adv-lxm	$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{adv} \\ \text{VAL} & \begin{bmatrix} \text{MOD} & \langle [\text{HEAD} \;\; \textit{verb}] \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & [\text{MODE} \;\; \text{none}] \end{bmatrix}$	*const-lxm*
det-lxm	$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{det} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & / \langle \; \rangle \\ \text{COMPS} & \langle \; \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & [\text{MODE} \;\; \text{none}] \end{bmatrix}$	*const-lxm*
predp-lxm	$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{prep} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \text{ X } \rangle \\ \text{MOD} & \langle \text{ Y } \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \\ \text{RESTR} & \langle \text{ Z } \rangle \end{bmatrix} \\ \text{ARG-ST} & \langle \text{ NP , NP } \rangle \end{bmatrix}$	*const-lxm*
argmkp-lxm	$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{prep} \\ \text{VAL} & [\text{SPR} \;\; \langle \; \rangle] \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \boxed{1} \\ \text{INDEX} & \boxed{2} \\ \text{RESTR} & \langle \; \rangle \end{bmatrix} \\ \text{ARG-ST} & \left\langle \overset{\text{NP}}{\begin{bmatrix} \text{MODE} & \boxed{1} \\ \text{INDEX} & \boxed{2} \end{bmatrix}} \right\rangle \end{bmatrix}$	*const-lxm*
part-lxm		*const-lxm*

OTHER GRAMMATICAL TYPES		
TYPE	FEATURES/CONSTRAINTS	IST
pos	$\left[\text{FORM} \quad \left\{\begin{array}{l}\text{fin, base, prp, psp, pass,}\\ \text{to, nform, aform, ...}\end{array}\right\}\right]$	*feat-struc*
agr-pos	$\left[\text{AGR} \quad \textit{agr-cat}\right]$	*pos*
verb	$\left[\text{AUX} \quad \{+, -\}\right]$	*agr-pos*
noun	$\left[\begin{array}{ll}\text{FORM} & /\ \text{nform}\\ \text{CASE} & \{\text{nom, acc}\}\end{array}\right]$	*agr-pos*
det	$\left[\text{COUNT} \quad \{+, -\}\right]$	*agr-pos*
adj	$\left[\text{FORM} \quad \text{aform}\right]$	*pos*
prep, *adv*, *conj*		*pos*
agr-cat	$\left[\begin{array}{ll}\text{PER} & \{\text{1st, 2nd, 3rd}\}\\ \text{NUM} & \{\text{sg, pl}\}\end{array}\right]$	*feat-struc*
3sing	$\left[\begin{array}{ll}\text{PER} & \text{3rd}\\ \text{NUM} & \text{sg}\\ \text{GEND} & \{\text{fem, masc, neut}\}\end{array}\right]$	*agr-cat*
non-3sing		*agr-cat*
1sing	$\left[\begin{array}{ll}\text{PER} & \text{1st}\\ \text{NUM} & \text{sg}\end{array}\right]$	*non-3sing*
non-1sing		*non-3sing*
2sing	$\left[\begin{array}{ll}\text{PER} & \text{2nd}\\ \text{NUM} & \text{sg}\end{array}\right]$	*non-1sing*
plural	$\left[\text{NUM} \quad \text{pl}\right]$	*non-1sing*
predication	$\left[\begin{array}{ll}\text{RELN} & \{\textbf{love}, \textbf{walk}, \ldots\}\\ \ldots & \end{array}\right]$	*feat-struc*

9.2.3 Abbreviations

$$\text{S} = \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \qquad \text{NP}_i = \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & i \end{bmatrix} \end{bmatrix}$$

$$\text{VP} = \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle \text{X} \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \qquad \text{NOM} = \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle \text{X} \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\text{V} = \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{verb} \end{bmatrix} \end{bmatrix} \qquad \text{N} = \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{noun} \end{bmatrix} \end{bmatrix}$$

$$\text{PP} = \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{prep} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \qquad \text{AP} = \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{adj} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\text{P} = \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{prep} \end{bmatrix} \end{bmatrix} \qquad \text{A} = \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{adj} \end{bmatrix} \end{bmatrix}$$

$$\text{DP} = \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{det} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

9.2.4 The Grammar Rules

(All daughters in our grammar rules are expressions, i.e. of type *word* or *phrase*; never of type *lexeme*).

(1) Head-Specifier Rule

$$\begin{bmatrix} \textit{phrase} & \\ \text{SPR} & \langle\ \rangle \end{bmatrix} \rightarrow \boxed{1} \quad \mathbf{H}\begin{bmatrix} \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix}$$

A phrase can consist of a (lexical or phrasal) head preceded by its specifier.

(2) Head-Complement Rule

$$\begin{bmatrix} \textit{phrase} & \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \rightarrow \mathbf{H}\begin{bmatrix} \textit{word} & \\ \text{COMPS} & \langle \boxed{1}, \ldots, \boxed{n} \rangle \end{bmatrix} \boxed{1} \ldots \boxed{n}$$

A phrase can consist of a lexical head followed by all its complements.

(3) Head-Modifier Rule

$$[\textit{phrase}] \rightarrow \mathbf{H}\boxed{1}\begin{bmatrix}\text{COMPS} & \langle\ \rangle\end{bmatrix} \begin{bmatrix}\text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\boxed{1}\rangle\end{bmatrix}$$

A phrase can consist of a (lexical or phrasal) head followed by a compatible modifier.

(4) Coordination Rule

$$\begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0}\end{bmatrix} \\ \text{SEM} & [\text{IND} \;\; s_0]\end{bmatrix} \rightarrow \begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0}\end{bmatrix} \\ \text{SEM} & [\text{IND} \;\; s_1]\end{bmatrix} \ldots \begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0}\end{bmatrix} \\ \text{SEM} & [\text{IND} \;\; s_{n-1}]\end{bmatrix}$$

$$\begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{HEAD} \; \textit{conj}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{IND} & s_0 \\ \text{RESTR} & \langle[\text{ARGS} \; \langle s_1, \ldots, s_n\rangle]\rangle\end{bmatrix}\end{bmatrix} \begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0}\end{bmatrix} \\ \text{SEM} & [\text{IND} \;\; s_n]\end{bmatrix}$$

Any number of elements with matching VAL and FORM specifications can form a coordinate phrase with identical VAL and FORM specifications.

(5) Imperative Rule

$$\begin{bmatrix}\textit{phrase} & \\ \text{HEAD} & \begin{bmatrix}\textit{verb}\end{bmatrix} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \rangle\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{MODE} & \text{dir} \\ \text{INDEX} & s\end{bmatrix}\end{bmatrix} \rightarrow \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\textit{verb} & \\ \text{FORM} & \text{base}\end{bmatrix} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \left\langle \text{NP}\begin{bmatrix}\text{PER} & \text{2nd}\end{bmatrix}\right\rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s\end{bmatrix}\end{bmatrix}$$

An imperative phrase can consist of a (lexical or phrasal) VP whose FORM value is base and whose unexpressed subject is 2nd person.

9.2.5 Lexical Rules

The following lexical rules interact with the constraints provided earlier for feature structures of type *i-rule* and *d-rule*:

(6) Singular Noun Lexical Rule

$$\begin{bmatrix}\textit{i-rule} & \\ \text{INPUT} & \left\langle \boxed{1}\,,\ \textit{cn-lxm}\right\rangle \\ \text{OUTPUT} & \left\langle \boxed{1}\,,\ \begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{AGR} & \begin{bmatrix}\text{NUM} \; \text{sg}\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix}\right\rangle\end{bmatrix}$$

(7) Plural Noun Lexical Rule

$$\left[\begin{array}{ll} \textit{i-rule} & \\ \text{INPUT} & \left\langle \boxed{1}\ , \textit{cntn-lxm} \right\rangle \\ \text{OUTPUT} & \left\langle \mathrm{F}_{NPL}(\boxed{1})\ , \left[\begin{array}{ll}\text{SYN} & \left[\begin{array}{ll}\text{HEAD} & \left[\begin{array}{ll}\text{AGR} & \left[\begin{array}{ll}\text{NUM} & \text{pl}\end{array}\right]\end{array}\right]\end{array}\right]\end{array}\right]\right\rangle \end{array}\right]$$

(8) 3rd-Singular Verb Lexical Rule

$$\left[\begin{array}{ll} \textit{i-rule} & \\ \text{INPUT} & \left\langle \boxed{3}\ , \left[\begin{array}{ll}\textit{verb-lxm} & \\ \text{SEM} & \left[\begin{array}{ll}\text{RESTR} & \boxed{A}\end{array}\right]\end{array}\right] \right\rangle \\ \text{OUTPUT} & \left\langle \mathrm{F}_{\mathit{3SG}}(\boxed{3})\ , \left[\begin{array}{ll}\text{SYN} & \left[\begin{array}{ll}\text{HEAD} & \left[\begin{array}{ll}\text{FORM} & \text{fin} \\ \text{AGR} & \textit{3sing}\end{array}\right]\end{array}\right] \\ \text{SEM} & \left[\begin{array}{ll}\text{RESTR} & \boxed{A} \oplus ...\end{array}\right] \\ \text{ARG-ST} & \left\langle \left[\begin{array}{ll}\text{CASE} & \text{nom}\end{array}\right], ... \right\rangle \end{array}\right]\right\rangle \end{array}\right]$$

(9) Non-3rd-Singular Verb Lexical Rule

$$\left[\begin{array}{ll} \textit{i-rule} & \\ \text{INPUT} & \left\langle \boxed{1}\ , \left[\begin{array}{ll}\textit{verb-lxm} & \\ \text{SEM} & \left[\begin{array}{ll}\text{RESTR} & \boxed{A}\end{array}\right]\end{array}\right] \right\rangle \\ \text{OUTPUT} & \left\langle \boxed{1}\ , \left[\begin{array}{ll}\text{SYN} & \left[\begin{array}{ll}\text{HEAD} & \left[\begin{array}{ll}\text{FORM} & \text{fin} \\ \text{AGR} & \textit{non-3sing}\end{array}\right]\end{array}\right] \\ \text{SEM} & \left[\begin{array}{ll}\text{RESTR} & \boxed{A} \oplus ...\end{array}\right] \\ \text{ARG-ST} & \left\langle \left[\begin{array}{ll}\text{CASE} & \text{nom}\end{array}\right], ... \right\rangle \end{array}\right]\right\rangle \end{array}\right]$$

(10) Past-Tense Verb Lexical Rule

$$\begin{bmatrix} \textit{i-rule} & \\ \text{INPUT} & \left\langle \boxed{3}\,, \begin{bmatrix} \textit{verb-lxm} & \\ \text{SEM} & \begin{bmatrix} \text{RESTR} & \boxed{\text{A}} \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \text{F}_{PAST}(\boxed{3})\,, \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{fin} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{RESTR} & \boxed{\text{A}} \oplus \ldots. \end{bmatrix} \\ \text{ARG-ST} & \left\langle \begin{bmatrix} \text{CASE} & \text{nom} \end{bmatrix}, \ldots \right\rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

(11) Base Form Lexical Rule

$$\begin{bmatrix} \textit{i-rule} & \\ \text{INPUT} & \left\langle \boxed{1}\,, \textit{verb-lxm} \right\rangle \\ \text{OUTPUT} & \left\langle \boxed{1}\,, \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{base} \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

(12) Constant Lexeme Lexical Rule

$$\begin{bmatrix} \textit{i-rule} & \\ \text{INPUT} & \left\langle \boxed{1}\,, \textit{const-lxm} \right\rangle \\ \text{OUTPUT} & \begin{bmatrix} \text{FIRST} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

(13) Present Participle Lexical Rule

$$\begin{bmatrix} \textit{d-rule} & \\ \text{INPUT} & \left\langle \boxed{3}\,, \begin{bmatrix} \textit{verb-lxm} & \\ \text{SEM} & \begin{bmatrix} \text{RESTR} & \boxed{\text{A}} \end{bmatrix} \\ \text{ARG-ST} & \boxed{\text{B}} \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \text{F}_{PRP}(\boxed{3})\,, \begin{bmatrix} \textit{part-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{prp} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{RESTR} & \boxed{\text{A}} \oplus \ldots. \end{bmatrix} \\ \text{ARG-ST} & \boxed{\text{B}} \end{bmatrix} \right\rangle \end{bmatrix}$$

(14) Past Participle Lexical Rule

$$\begin{bmatrix} \textit{d-rule} & \\ \text{INPUT} & \left\langle \boxed{3}\,, \begin{bmatrix} \textit{verb-lxm} & \\ \text{SEM} & \begin{bmatrix}\text{RESTR} & \boxed{A}\end{bmatrix} \\ \text{ARG-ST} & \boxed{B} \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \text{F}_{PSP}(\boxed{3})\,, \begin{bmatrix} \textit{part-lxm} & \\ \text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{FORM} & \text{psp}\end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{RESTR} & \boxed{A} \oplus ....\end{bmatrix} \\ \text{ARG-ST} & \boxed{B}\end{bmatrix}\right\rangle \end{bmatrix}$$

(15) Agent Nominalization Lexical Rule

$$\begin{bmatrix} \textit{d-rule} & \\ \text{INPUT} & \left\langle \boxed{2}\,, \begin{bmatrix} \textit{verb-lxm} & \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s\end{bmatrix} \\ \text{ARG-ST} & \langle\, \text{X}_i\,,\ \text{NP}_j\, \rangle \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \text{F}_{-er}(\boxed{2})\,, \begin{bmatrix} \textit{cntn-lxm} & \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & i\end{bmatrix} \\ \text{ARG-ST} & \left\langle \text{Y} \left(, \begin{matrix}\text{PP}_j \\ \begin{bmatrix}\text{FORM} & \text{of}\end{bmatrix}\end{matrix} \right) \right\rangle \end{bmatrix}\right\rangle \end{bmatrix}$$

9.2.6 The Basic Lexicon

Here are some sample lexical entries that are part of the basic lexicon. Each entry is a pair consisting of (1) a description of a phonological form and (2) a description satisfiable by feature structures of (some maximal subtype) of *lexeme*. Lexical entries include only information that is not inherited from other types. As before, the notation '...' indicates things we haven't dealt with but which a complete grammar would have to.

Nouns

(16)

$$\left\langle \text{she}\ , \begin{bmatrix} \textit{pron-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{CASE} & \text{nom} \\ \text{AGR} & \begin{bmatrix} \textit{3sing} & \\ \text{GEND} & \text{fem} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{female} \\ \text{INST} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(17)

$$\left\langle \text{him}\ , \begin{bmatrix} \textit{pron-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{CASE} & \text{acc} \\ \text{AGR} & \begin{bmatrix} \textit{3sing} & \\ \text{GEND} & \text{masc} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{male} \\ \text{INST} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(18)

$$\left\langle \text{themselves}\ , \begin{bmatrix} \textit{pron-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{CASE} & \text{acc} \\ \text{AGR} & \begin{bmatrix} \textit{plural} & \\ \text{PER} & \text{3rd} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ana} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{group} \\ \text{INST} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(19)

$$\left\langle \text{Kim}\ , \begin{bmatrix} \textit{pn-lxm} & \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{Kim} \\ \text{NAMED} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(20)

$$\left\langle \text{book} , \begin{bmatrix} \textit{cntn-lxm} & \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{book} \\ \text{INST} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Verbs

(21)

$$\left\langle \text{die} , \begin{bmatrix} \textit{siv-lxm} & \\ \text{ARG-ST} & \langle \text{X}_i \rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{die} \\ \text{SIT} & s \\ \text{CORPSE} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(22)

$$\left\langle \text{love} , \begin{bmatrix} \textit{stv-lxm} & \\ \text{ARG-ST} & \langle \text{X}_i , \text{Y}_j \rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{love} \\ \text{SIT} & s \\ \text{LOVER} & i \\ \text{LOVED} & j \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(23)

$$\left\langle \text{give} , \begin{bmatrix} \textit{dtv-lxm} & \\ \text{ARG-ST} & \langle \text{X}_i , \text{Y}_j , \text{Z}_k \rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{give} \\ \text{SIT} & s \\ \text{GIVER} & i \\ \text{GIVEN} & j \\ \text{GIFT} & k \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(24)

$$\left\langle \text{give}\ ,\ \left[\begin{array}{ll} \textit{ptv-lxm} & \\ \text{ARG-ST} & \langle\, \text{X}_i\ ,\ \text{Y}_k\ ,\ \text{Z}_j\left[\text{FORM}\ \ \text{to}\right] \rangle \\ \text{SEM} & \left[\begin{array}{ll} \text{INDEX} & s \\ \text{RESTR} & \left\langle \left[\begin{array}{ll} \text{RELN} & \textbf{give} \\ \text{SIT} & s \\ \text{GIVER} & i \\ \text{GIVEN} & j \\ \text{GIFT} & k \end{array}\right] \right\rangle \end{array}\right] \end{array}\right] \right\rangle$$

Miscellaneous

(25)

$$\left\langle \text{the}\ ,\ \left[\begin{array}{ll} \textit{det-lxm} & \\ \text{SEM} & \left[\begin{array}{ll} \text{INDEX} & i \\ \text{RESTR} & \left\langle \left[\begin{array}{ll} \text{RELN} & \textbf{the} \\ \text{BV} & i \end{array}\right] \right\rangle \end{array}\right] \end{array}\right] \right\rangle$$

(26)

$$\left\langle \text{few}\ ,\ \left[\begin{array}{ll} \textit{det-lxm} & \\ \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll} \text{AGR} & \left[\text{NUM pl}\right] \\ \text{COUNT} & + \end{array}\right] \end{array}\right] \\ \text{SEM} & \left[\begin{array}{ll} \text{INDEX} & i \\ \text{RESTR} & \left\langle \left[\begin{array}{ll} \text{RELN} & \textbf{few} \\ \text{BV} & i \end{array}\right] \right\rangle \end{array}\right] \end{array}\right] \right\rangle$$

(27)

$$\left\langle \text{'s}\ ,\ \left[\begin{array}{ll} \textit{det-lxm} & \\ \text{SYN} & \left[\text{VAL}\ \ \left[\text{SPR}\ \ \langle\ \text{NP}\ \rangle\right]\right] \\ \text{SEM} & \left[\begin{array}{ll} \text{INDEX} & i \\ \text{RESTR} & \left\langle \left[\begin{array}{ll} \text{RELN} & \textbf{the} \\ \text{BV} & i \end{array}\right], \ldots \right\rangle \end{array}\right] \end{array}\right] \right\rangle$$

(28)

$$\left\langle \text{to}\ ,\ \left[\begin{array}{ll} \textit{argmkp-lxm} & \\ \text{SYN} & \left[\text{HEAD}\ \ \left[\text{FORM}\ \ \text{to}\right]\right] \end{array}\right] \right\rangle$$

(29)
$$\left\langle \text{in}\ , \begin{bmatrix} \textit{predp-lxm} & \\ \text{ARG-ST} & \langle\ \text{NP}_i\ ,\ \text{NP}_j\ \rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{in} \\ \text{SIT} & s \\ \text{CONTAINER} & j \\ \text{CONTAINED} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(30)
$$\left\langle \text{and}\ , \begin{bmatrix} \textit{conj-lxm} & \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{and} \\ \text{SIT} & s \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(31)
$$\left\langle \text{today}\ , \begin{bmatrix} \textit{adv-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \left\langle \begin{matrix} \text{VP} \\ \begin{bmatrix} \text{INDEX} & s \end{bmatrix} \end{matrix} \right\rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{today} \\ \text{ARG} & s \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

9.2.7 Well-Formed Structures

In this section, we lay out more precisely the constructs of the theory whose effects we have been illustrating in the preceding chapters. As noted in Chapter 6, the definitions presented in section 36 below, should be sufficient for most readers.

Preliminaries

According to our approach, a grammar G is defined by the following components:

- a finite set of features: $\mathcal{F} = \{\text{SYN, SEM, HEAD, AGR, } \ldots\}$,
- a finite set of primitive items:
  $\mathcal{A}_{atom} = \mathcal{A}_{pol} \cup \mathcal{A}_{gr.atom} \cup \mathcal{A}_{mode} \cup \mathcal{A}_{reln}$, where:
  1. $\mathcal{A}_{pol} = \{+, -\}$,
  2. (a set of ground atoms) $\mathcal{A}_{gr.atom} = \{\text{1st}, \text{2nd}, \text{3rd}, \text{sg}, \text{pl}, \ldots, \text{run}, \text{dog}, \ldots\}$,
  3. $\mathcal{A}_{mode} = \{\text{prop}, \text{ques}, \text{dir}, \text{ref}, \text{none}\}$, and
  4. $\mathcal{A}_{reln} = \{\textbf{walk}, \textbf{love}, \textbf{person}, \ldots\}$,
- a denumerably infinite set of primitive items: $\mathcal{A}_{index} = \mathcal{A}_{ind} \cup \mathcal{A}_{sit}$, where:
  1. $\mathcal{A}_{ind} = \{i, j, \ldots\}$ and
  2. $\mathcal{A}_{sit} = \{s_1, s_2, \ldots\}$,
- the distinguished element *elist* (*empty-list*), discussed below,
- a finite set of types: $\mathcal{T} = \{$*noun*, *agr-pos*, *plural*, *expression*, ...$\}$,
- a type hierarchy with a tree structure associated with constraint inheritance (for instance, the type hierarchy represented by the tree and table in Section 9.2.1 and 9.2.2),
- a set $\mathcal{LT} \subset \mathcal{T}$ called the *leaf type* (a type τ is a *leaf type* if it is associated with a leaf in the type hierarchy tree, i.e. if τ is one of the most specific types),
- a set of list types (if τ is a type, then *list*(τ) is a list type),
- a set of grammar rules (see Section 9.2.4),
- a set of principles,
- a lexicon (which is a finite set of lexical entries like those in Section 9.2.6), and
- a set of lexical rules (like those in Section 9.2.5).

Thus a grammar G comes with various primitives grouped into two sets: $\mathcal{A}_{atom}$ ($\mathcal{A}_{pol}$, $\mathcal{A}_{gr.atom}$, $\mathcal{A}_{mode}$, $\mathcal{A}_{reln}$) and $\mathcal{A}_{index}$ ($\mathcal{A}_{ind}$, and $\mathcal{A}_{sit}$). G assigns the type *atom* to all elements of $\mathcal{A}_{atom}$. The elements of $\mathcal{A}_{index}$ are used by the grammar for describing individual objects and situations; they are associated with the leaf type *index*. We assume that no items in these sets of primitives can be further analyzed via grammatical features.

Our grammar appeals to several ancillary notions which we now explicate: *feature structure description*, *feature structure*, *satisfaction of a description*, and *tree structure*.

Feature Structure Descriptions

For expressing the constraints associated with the grammar rules, principles, types, and lexical entries, we introduce the notion of a feature structure description. The feature structure descriptions are given as attribute-value matrices, augmented with the connective '|', set descriptors ($\{\ldots\}$), list descriptions ($\langle\ldots\rangle$, attribute-value matrices with FIRST/REST, or two list descriptions connected by $\oplus$), and a set *Tags* of tags (labels represented by boxed integers or letters).

Feature Structures

The set of the feature structures $\mathcal{FS}$ is given by the following recursive definition:

(32) $\phi \in \mathcal{FS}$ (i.e. ϕ is a feature structure) iff

a. $\phi \in \mathcal{A}_{atom} \cup \mathcal{A}_{index}$, or

b. ϕ is a function from features to feature structures, $\phi : \mathcal{F} \longrightarrow \mathcal{FS}$ satisfying the following conditions

1. ϕ is of a leaf type τ;
2. $DOM(\phi) = \{F \mid \text{G declares } F \text{ appropriate for } \tau\} \cup$

$$\{F' \mid \exists\tau' \text{ such that } \tau' \text{ is a supertype of } \tau \text{ and}$$
$$\text{G declares } F' \text{ appropriate for } \tau'\},$$

i.e. ϕ is defined for any feature that is declared appropriate for τ or for any of τ's supertypes;

3. for each $F \in DOM(\phi)$, G defines the type of the value $\phi(F)$ (we call the value $\phi(F)$ of the function ϕ on F the value of the feature F); and
4. ϕ obeys all further constraints ('type constraints') that G associates with type τ (including those inherited by default from the supertypes τ' of τ), or

c. ϕ is of type *list*(τ), for some type τ, in which case either:

1. ϕ is the distinguished element *elist*, or else:
2. A. *DOM*(ϕ) is $\{\text{FIRST}, \text{REST}\}$,
   B. the type of ϕ(FIRST) is τ, and
   C. the type of ϕ(REST) is *list*(τ).

Satisfaction

We explain how feature structures satisfy descriptions indirectly – in terms of denotation, which we define as follows:

Denotation of Feature Structure Descriptions

The denotation of a feature structure description is specified in terms of a structure $\mathcal{M}$:

(33) $\mathcal{M} = \langle \mathcal{A}, \mathcal{F}, \mathcal{T}, \mathbf{Type}, I \rangle$, where:

1. $\mathcal{A} = \mathcal{A}_{atom} \cup \mathcal{A}_{index} \cup \{elist\}$,
2. $\mathcal{F}$ is a finite set of features,
3. $\mathcal{T}$ is a finite set of types,

4. **Type** is a function mapping feature structures to types –
   Type : $\mathcal{FS} \longrightarrow \mathcal{LT}$, where $\mathcal{LT}$ is the set of the leaf types, and
5. I is a function mapping feature names and atomic descriptors to features and atoms of the appropriate sort:

$$I = I_{\widetilde{\mathcal{F}}} \cup I_{\widetilde{\mathcal{A}}_{atom}} \cup I_{\widetilde{\mathcal{A}}_{ind}} \cup I_{\widetilde{\mathcal{A}}_{sit}} \cup \{\langle elist, elist\rangle\},$$

   where
   $I_{\widetilde{\mathcal{F}}} \in \mathcal{F}^{\widetilde{\mathcal{F}}}$, $I_{\widetilde{\mathcal{A}}_{atom}} \in \mathcal{A}_{atom}{}^{\widetilde{\mathcal{A}}_{atom}}$, $I_{\widetilde{\mathcal{A}}_{ind}} \in \mathcal{A}_{ind}{}^{\widetilde{\mathcal{A}}_{ind}}$, $I_{\widetilde{\mathcal{A}}_{sit}} \in \mathcal{A}_{sit}{}^{\widetilde{\mathcal{A}}_{sit}}$,

   and $\widetilde{X}$ denotes the set of expressions that have denotations in the set X.[2]

The function I is called an *interpretation function.* An *assignment function* is a function

$$g : Tags \longrightarrow \mathcal{FS}.$$

We say that a feature structure ϕ is of a type $\tau \in \mathcal{T}$ iff there is a (unique) leaf type $\tau' \in \mathcal{LT}$ such that:

(34)
1. τ' is a subtype of τ, and
2. **Type**$(\phi) = \tau'$.

Given $\mathcal{M}$, the interpretation $[\![d]\!]^{\mathcal{M},g}$ of a feature structure description d with respect to an assignment function g is defined recursively as follows:

(35)
1. if $v \in \widetilde{\mathcal{F}} \cup \widetilde{\mathcal{A}}_{atom} \cup \widetilde{\mathcal{A}}_{index} \cup \{elist\}$, *then* $[\![v]\!]^{\mathcal{M},g} = I(v)$;
2. if τ is a type, i.e. $\tau \in \mathcal{T}$, then $[\![\tau]\!]^{\mathcal{M},g} = \{\phi \in \mathcal{FS} : \phi \text{ is of type } \tau\}$;
3. if $F \in \widetilde{\mathcal{F}}$, and d is a feature structure description, then $[\![[F\ d]]\!]^{\mathcal{M},g} =$
   $\{\phi \in \mathcal{FS} : \text{there is some } \phi' \text{ such that } \phi' \in [\![d]\!]^{\mathcal{M},g} \text{ and } \langle I(F), \phi'\rangle \in \phi\}$;[3]
4. if $d = \begin{bmatrix} d_1 \\ \dots \\ d_n \end{bmatrix}$
   where $n \geq 1$, and $d_1, \dots, d_n$ are feature structure descriptions, then

$$[\![d]\!]^{\mathcal{M},g} = \bigcap_{i=1}^{n} [\![d_i]\!]^{\mathcal{M},g};$$

5. if d is a set descriptor $\{d_1, \dots, d_n\}$, then

$$[\![d]\!]^{\mathcal{M},g} = \bigcup_{i=1}^{n} [\![d_i]\!]^{\mathcal{M},g}$$

   $([\![\{\ \}]\!]^{\mathcal{M},g} = \emptyset)$;
6. $[\![d_1 \mid d_2]\!]^{\mathcal{M},g} = [\![d_1]\!]^{\mathcal{M},g} \cup [\![d_2]\!]^{\mathcal{M},g}$;
7. if $d \in Tags$, then $[\![d]\!]^{\mathcal{M},g} = g(d)$;
8. if $d \in Tags$ and d' is a feature structure description, then
   $[\![d\ d']\!]^{\mathcal{M},g} = \{\phi \in \mathcal{FS} : g(d) = \phi \text{ and } \phi \in [\![d']\!]^{\mathcal{M},g}\}$;
   (Note that tagging narrows the interpretation down to a singleton set.)

[2] Y^X is the standard notation for the set of all functions $f : X \to Y$.

[3] Note that the definition of a feature structure in (32), taken together with this clause, ensures that each element ϕ of the set $[\![[F\ d]]\!]^{\mathcal{M},g}$ is a proper feature structure.

9. List Addition:[4]

a. $[\![elist \oplus d]\!]^{\mathcal{M},g} = [\![d]\!]^{\mathcal{M},g}$,

b. if $d = \begin{bmatrix} \text{FIRST} & d_1 \\ \text{REST} & d_2 \end{bmatrix} \oplus d_3$,

then $[\![d]\!]^{\mathcal{M},g} =$
$\{\phi \in \mathcal{FS} : \phi(\text{FIRST}) \in [\![d_1]\!]^{\mathcal{M},g}$ and $\phi(\text{REST}) \in [\![d_2 \oplus d_3]\!]^{\mathcal{M},g}\}$.

Satisfaction of Feature Structure Descriptions[5]

A feature structure $\phi \in \mathcal{FS}$ satisfies a feature structure description d iff there is some assignment function g such that $\phi \in [\![d]\!]^{\mathcal{M},g}$.

For examples of feature structures that satisfy particular descriptions, see Section 6.3.4 of Chapter 6.

Tree Structures

Finally, we assume a notion of tree structure described informally as follows:

(36) A tree structure is a directed graph that satisfies a number of conditions:[6]
1. it has a unique root node,
2. each non-root node has exactly one mother,
3. sister nodes are ordered with respect to each other,
4. it has no crossing branches,
5. each nonterminal node is labeled by a feature structure, and
6. each terminal node is labeled by a phonological form (an atom).

Structures Defined by the Grammar

(37) **Well-Formed Tree Structure**:

Φ is a Well-Formed Tree Structure according to G if and only if:
1. Φ is a tree structure,
2. the label of Φ's root node satisfies the constraint:

$$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{fin} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}, \text{ and}$$

3. each local subtree within Φ is either phrasally licensed or lexically licensed.

[4] Where no confusion should arise, we use 'FIRST', 'SYN', etc. to refer either to the appropriate feature (an element of $\mathcal{F}$) or to its name (an element of $\widetilde{\mathcal{F}}$).

[5] We make no attempt here to extend this definition to include the satisfaction of defeasible constraints. For a logic of typed feature structures with defeasible constraints, see Lascarides and Copestake 1999, whose feature structures embody a distinction between defeasible and indefeasible information. Alternatively, one might view the inheritance hierarchy more syntactically, as a means for enriching the constraints on leaf types via the inheritance of compatible constraints from superordinate types. As noted in Chapter 8, such an approach would draw a distinction between 'initial descriptions' and 'enriched descriptions' of linguistic entities. Assuming then that the constraints associated with individual lexemes, words, and lexical rules would all be indefeasible, this syntactic approach to constraint inheritance would not require any revision of the satisfaction definition provided in the text.

[6] Again, we assume familiarity with notions such as *root*, *mother*, *terminal node*, *non-terminal node*, and *branches*. See footnote 16 of Chapter 6.

Lexical Licensing is defined in terms of lexical sequences that are legitimate outputs of lexical rules. The instances of the type *lexical-sequence* are defined as follows:

(38) **Lexical Sequences**:
$\langle \omega, \phi \rangle$ is a lexical sequence if and only if ω is a phonological form (an atom), ϕ is a feature structure, and either:

1. G contains some lexical entry $\langle d_1, d_2 \rangle$ such that ω satisfies d_1 and ϕ satisfies d_2, or
2. there is some lexical rule instantiation licensed by G (a feature structure of type *l-rule*) whose OUTPUT value is $\langle \omega, \phi \rangle$.

(39) **Lexical Licensing**:
A word structure of the form:

$$\begin{array}{c} \phi \\ | \\ \omega \end{array}$$

is licensed if and only if:

1. $\langle \omega, \phi \rangle$ is a lexical sequence, where ϕ is of type *word*,
2. (Case Constraint:) An outranked NP is [CASE acc], and
3. ϕ satisfies the Binding Theory.

(40) **The Binding Theory**:

Principle A: A [MODE ana] expression must be outranked by a coindexed element.
Principle B: A [MODE ref] expression must not be outranked by a coindexed element;
where:

(i) If a node is coindexed with its daughter, their feature structures are of equal rank.
(ii) If there is an ARG-ST list on which A precedes B, then A outranks B.

(41) **Phrasal Licensing**:
A grammar rule $\rho = d_0 \rightarrow \; d_1 \ldots \; d_n$ licenses a local subtree:

$$\Phi = \begin{array}{ccc} & \phi_0 & \\ \swarrow & & \searrow \\ \phi_1 & \ldots & \phi_n \end{array}$$

if and only if:

1. for each $i, 0 \leq i \leq n$, ϕ_i is of type *expression*,
2. there is some assignment function g under which the sequence $\langle \phi_0, \phi_1, ..., \phi_n \rangle$ satisfies the description sequence $\langle d_0, \, d_1, \ldots, \, d_n \rangle$,
3. Φ satisfies the Semantic Compositionality Principle and the Anaphoric Agreement Principle, and
4. if ρ is a headed rule, then Φ satisfies the Head Feature Principle, the Valence Principle and the Semantic Inheritance Principle, with respect to ρ.

(42) Φ satisfies the Semantic Compositionality Principle with respect to a grammar rule ρ if and only if Φ satisfies:

$$\begin{array}{ccc} & \left[\text{RESTR} \quad \boxed{A_1} \oplus ... \oplus \boxed{A_n}\right] & \\ \swarrow & \downarrow & \searrow \\ [\text{RESTR} \ \boxed{A_1}] & \dots & [\text{RESTR} \ \boxed{A_n}] \end{array}$$

(43) Anaphoric Agreement Principle:
Coindexed NPs agree (i.e. their AGR values are identical).

(44) Φ satisfies the Head Feature Principle with respect to a headed rule ρ if and only if Φ satisfies:

$$\begin{array}{ccc} & [\text{HEAD} \ \boxed{1}] & \\ \swarrow & \downarrow & \searrow \\ \dots & \begin{array}{c}\phi_h \\ \left[\text{HEAD} \quad \boxed{1}\right]\end{array} & \dots \end{array}$$

where ϕ_h is the head daughter of Φ.

(45) Φ satisfies the Valence Principle with respect to a headed rule ρ if and only if, for any VAL feature F, Φ satisfies:

$$\begin{array}{ccc} & \left[\text{F} \quad \boxed{A}\right] & \\ \swarrow & \downarrow & \searrow \\ \dots & \begin{array}{c}\phi_h \\ \left[\text{F} \quad \boxed{A}\right]\end{array} & \dots \end{array}$$

where ϕ_h is the head daughter of Φ and ρ does not specify incompatible F values for ϕ_h and ϕ_0.

(46) Φ satisfies the Semantic Inheritance Principle with respect to a headed rule ρ if and only if Φ satisfies:

$$\begin{array}{ccc} & \begin{bmatrix}\text{MODE} & \boxed{4} \\ \text{INDEX} & \boxed{5}\end{bmatrix} & \\ \swarrow & \downarrow & \searrow \\ \dots & \begin{array}{c}\phi_h \\ \begin{bmatrix}\text{MODE} & \boxed{4} \\ \text{INDEX} & \boxed{5}\end{bmatrix}\end{array} & \dots \end{array}$$

where ϕ_h is the head daughter of Φ.

9.3 Constraint-Based Lexicalism

We turn now to some reflections on the relationship between the sort of grammatical descriptions in this text and what is known about the mental processes underlying human language comprehension and production. Adopting the familiar terminology of Chomsky (1965), we distinguish between speakers' knowledge of their language – what Chomsky called their 'competence' – and the ways in which that knowledge is put to use in speaking and understanding – what Chomsky called 'performance'.

The way we speak and understand is clearly influenced by many things other than our linguistic knowledge. For example, we all make speech errors on occasion, reversing words or garbling our utterances in other ways; and we also sometimes misunderstand what was said. These sorts of errors are more likely to occur under certain conditions (such as a drunk speaker or a noisy environment) that have nothing to do with the interlocutors' knowledge of the language.

There are also subtler aspects of the competence/performance distinction. For example, memory limitations prevent anyone from being able to produce or understand a sentence a million words long. But we do not say that all such examples are ungrammatical, because the memory limitations that make such sentences unusable are not intrinsic to our knowledge of language. (If a speaker were to come along who could produce and understand million-word sentences of English, we would not say that that person spoke a different language from our own). Many other aspects of language use, including what people find easy and hard to understand, are generally included under the rubric of performance.

Psycholinguists are concerned with developing models of people's actual use of language. They try to figure out what sequences of (largely unconscious) steps people go through in producing and understanding utterances. They are, therefore, concerned with the types of errors people make, with what people find easy and difficult, and with how nonlinguistic factors influence language use. In short, psycholinguists study performance.

Chomsky (1965:15) wrote: 'In general, it seems that the study of performance models incorporating generative grammars may be a fruitful study; furthermore, it is difficult to imagine any other basis on which a theory of performance might develop.' We agree wholeheartedly with the idea of incorporating competence grammars into models of performance. However, at the time Chomsky wrote this, only one theory of generative grammar had been given serious consideration for modeling natural language. Since that time, a wide range of alternatives have been explored. One obvious basis for comparing these alternatives is to see how well they comport with what is known about performance. That is, theories of linguistic competence should be able to serve as a basis for testable models of linguistic performance.

We believe not only that grammatical theorists should be interested in performance modeling, but also that empirical facts about various aspects of performance can and should inform the theory of linguistic competence. That is, compatibility with performance models should bear on the design of grammars. As we will show later in this chapter, there is now a considerable body of psycholinguistic results that suggest properties that a competence theory should have, if it is to be embedded within an account of human linguistic performance. And we will argue that the theory we have been developing does well on this criterion.[7]

Let us start with three basic observations about the grammar we have been developing:

1. It is SURFACE ORIENTED. Our grammar (like standard context-free grammars) provides a reasonably simple structure that is directly associated with the string of words that constitute each sentence. The ancillary structure that has to be computed to ascertain whether a given sentence is grammatical expresses information that is straightforwardly derivable from properties of the words in the string. No additional abstract structures are posited. In particular, our theory has no need for the sequences of phrase structures that constitute the derivations of sentences in transformational grammar.
2. It is CONSTRAINT-BASED. There are no operations that destructively modify any representations. The principles of the theory, the grammar rules, and the lexical entries are all just constraints that interact so as to define a set of phrase structures – those that simultaneously satisfy the relevant constraints of our grammar. Once generated, phrase structures are not rearranged, trimmed, or otherwise modified via transformational rules.
3. It is STRONGLY LEXICALIST. We have localized most grammatical and semantic information within lexical entries. These lexical entries furthermore correspond directly to the words present in the sentence, which can be viewed as the key elements that drive the construction of the syntactic and semantic structure of the sentence. As will become evident in the next few chapters, many of the relationships that transformational grammarians have analyzed using rules relating sentence types are handled in our theory via lexical rules.

Any theory that has these three design properties exemplifies a viewpoint that we will refer to as CONSTRAINT-BASED LEXICALISM (CBL).

9.4 Modeling Performance

Available evidence on how people produce and comprehend utterances provides some general guidelines as to the nature of an adequate performance model. Some of that evidence is readily available to anyone who pays attention to language use. Other evidence has come out of controlled laboratory experiments, in some cases requiring sophisticated methods and equipment. The two most striking facts about language processing are the following:

- Language processing is incremental: Utterances are sequences of sounds. At any point in the production or comprehension of an utterance, language users are working on what has just been said and what is about to be said. Speakers do not wait until they have their utterances fully planned to begin speaking; and listeners do not wait until the end of an utterance to begin trying to figure out what the speaker means to say.
- Language processing is rapid: producing and understanding three words per second is no problem.

[7] Jackendoff (2002:Chapter 7) makes a similar argument. He takes a different stand on the question of modularity, discussed in Section 9.4.3, but on the whole his conclusions and ours are quite similar.

9.4.1 Incremental Processing

We don't have to venture into a psycholinguistic laboratory to convince ourselves that language processing is highly incremental. We saw this already in Chapter 1, when we considered examples like (47):

(47) After finding the book on the atom, Sandy went into class, confident that there would be no further obstacles to getting that term paper done.

When we hear such a sentence, we process it as it comes – more or less word by word – building structure and partial interpretation incrementally, using what nonlinguistic information we can to make the right decisions at certain points. For example, when we encounter the PP *on the atom*, we have to decide whether it modifies VP or NOM; this is a kind of ambiguity resolution, i.e. deciding which of two currently available analyses is the one intended. We make this decision 'on-line' it seems, using a plausibility assessment of the meaning that would result from each structure. Information that can resolve such a local parsing ambiguity may appear later in the sentence. If the processor makes a decision about how to resolve a local ambiguity, but information later in the sentence shows that the decision was the wrong one, we would expect processing to be disrupted.

And indeed, psycholinguists have shown us that sentence processing sometimes does go astray. Garden path examples like (48a,b) are as remarkable today as they were when they were first brought to the attention of language researchers.[8]

(48) a. The horse raced past the barn fell.
b. The boat floated down the river sank.

On first encountering such examples, almost all English speakers judge them to be totally ungrammatical. However, after seeing them juxtaposed to fully well-formed examples like (49), speakers recognize that examples like (48) are grammatical sentences, though very hard to process.

(49) a. The horse that was raced past the barn fell.
b. The horse taken to the hospital died.
c. The boat that was floated down the river sank.
d. The boat seen down the river sank.

Experimental researchers thought at first that these garden paths showed that certain purely linguistic processing strategies (like trying to build an S out of the NP *the horse* and a VP beginning with *raced past*) were automatic - virtually impossible to turn off. But modern psycholinguistics has a very different story to tell.

First, note that in the right context, one can eliminate the garden path effect even with the sentences in (48). The right context can even make the NOM-modifying interpretation of *raced past the barn* the most natural one:[9]

(50) The horse that they raced around the track held up fine. The horse that was raced down the road faltered a bit. And the horse raced past the barn fell.

[8] By Bever (1970).
[9] This kind of effect is discussed by Crain and Steedman (1985).

The context here highlights the need to identify one horse among many, which in turn favors the meaning of the NOM-modifying structure of (48a).

Moreover, if we keep the same potential for ambiguity, but change the words, we can eliminate the garden path effect even without an elaborate preceding context. Consider examples like (51a,b).

(51) a. The evidence assembled by the prosecution convinced the jury.
 b. The thief seized by the police turned out to be our cousin.

As shown in a number of studies,[10] examples like these present no more processing difficulty than their unambiguous counterparts in (52):

(52) a. The evidence that was assembled by the prosecution convinced the jury.
 b. The thief who was seized by the police turned out to be our cousin.

That is, the examples in (51), even in the absence of a prior biasing context, do not cause garden path effects.

The explanation for this difference lies in the relevant nonlinguistic information. Evidence (or, say, a particular piece of evidence) can't assemble itself (or anything else), and the sentence built out of a subject NP *the evidence* and a VP headed by *assembled* would require some such implausible interpretation. (Similarly, intransitive uses of *seize* normally take some sort of mechanical device as their subject, making a thief an unlikely subject for *seized* in (51b)). That is, it is a fact about the world that only animate things (like people, animals, and perhaps some kinds of machines or organizations) assemble, and since evidence is inanimate, that hypothesis about the interpretation of the sentence is implausible. The fact that the decision to reject that interpretation (and hence the associated sentential structure) is made so quickly as to be imperceptible (i.e. so as to produce no noticeable garden path effect) is evidence that language comprehension is working in a highly integrative and incremental fashion. Linguistic and nonlinguistic constraints on the interpretation are interleaved in real time.

9.4.2 Rapid Processing

Just how rapidly people integrate available information in processing language has become evident since the early 1990s, thanks largely to technological advances that have made possible sophisticated new methods for investigating language use.[11] Of particular interest in the present context are head-mounted eye trackers, whose application to psycholinguistic research was pioneered by Michael Tanenhaus of the University of Rochester. These devices show investigators exactly where a participant's gaze is directed at any given moment. By following listeners' eye movements during speech, it is possible to draw inferences about their mental processes on a syllable-by-syllable basis.

The evidence from a great many experiments using this technique can be summed up concisely as follows: listeners use whatever information is available to them, as soon as it becomes available to them, to infer the speaker's intentions. In other words, language processing rapidly draws on all available types of linguistic and non-linguistic information as such information is needed.

[10] See, for example, Trueswell et al. 1992, Pearlmutter and MacDonald l992, and Tabossi et al. 1994.

[11] However, earlier work had made similar points. See, for example, Marslen-Wilson and Tyler 1987.

In one study, for example, participants viewed a grid with several objects on it, e.g. a box, a wallet, a fork, etc. Two of the objects would normally be described with words whose initial portions sound the same, for example, a *candle* and a *candy*; such pairs are called 'competitors'. Participants received instructions to pick up an object and to place it somewhere else on the grid. For example, they might be told, 'Pick up the candle. Now put it above the fork'. In some cases, the object they were told to pick up had a competitor on the grid (e.g. in the example just given, a candy might be present). Comparing cases in which a competitor was present to cases without a competitor provided evidence regarding the processes of word recognition and comprehension. Participants eye movements to the objects they picked up were significantly faster in cases when no competitor was present (445 milliseconds vs. 530 milliseconds). Tanenhaus et al. (1996:466) concluded that the timing of eye movements 'provides clear evidence that retrieval of lexical information begins before the end of a word.'

Another study (also described by Tanenhaus et al. (1996)) involved sets of blocks that could differ in marking, color, and shape, so that uniquely identifying one with a verbal description would require a multi-word phrase. The stimuli were manipulated so that the target objects could be uniquely identified early, midway, or late in the production of the description. Listeners' gaze again moved to the target object as soon as the information necessary for unique identification was uttered. What this information was depended not only on the words used, but also on what was in the visual display.

When one word in a description is contrastively accented (e.g. *the* LARGE *blue triangle*), the conditions for unique identification are different, since there must be another object present satisfying all but the contrasting word in the description (e.g. a small blue triangle). In some cases, this allows earlier resolution of the reference of a phrase. Eye-tracking shows that listeners use such accentual information in determining reference (Tanenhaus et al. 1996).

Similar results have been obtained under many different conditions. For example, eye movements show that resolution of prepositional phrase attachment ambiguities (*Put the apple on the towel in the box*) takes place as soon as listeners have the information needed for disambiguation, and this likewise depends on both linguistic factors and the visual display (see Tanenhaus et al. 1995).

Recent eye-tracking studies (Arnold et al. 2002) show that even disfluencies in speech are used by listeners to help them interpret speakers' intentions. In particular, when a disfluency such as *um* or *uh* occurs early in a description, listeners tend to look at objects that have not yet been mentioned in the discourse. This makes sense, since descriptions of new referents are likely to be more complex, and hence to contain more disfluencies, than descriptions of objects previously referred to. Once again, the eye movements show the listeners using the information as soon as it becomes available in identifying (or, in this case, predicting the identification of) the objects that speakers are referring to.

It is easy to come up with many more examples showing that language comprehension proceeds rapidly and incrementally, with different types of information utilized as they are needed and available. The same is true of language production. One type of evidence for this again comes from disfluencies (see, for example, Clark and Wasow 1998 and Clark and Fox Tree 2002). The high rate of disfluencies in spontaneous speech shows that people start their utterances before they have finished planning exactly what they are going to say and how they want to say it. And different types of disfluencies are symptoms of different kinds of production problems. For example, speakers tend to pause longer when they say *um* than when they say *uh*, suggesting that *um* marks more serious production problems. Correspondingly, *um* tends to occur more frequently at the beginnings of utterances,[12] when more planning is required, and its frequency relative to *uh* decreases later in utterances. The locations and frequencies of various types of disfluencies show that people are sensitive to a wide variety of linguistic and nonlinguistic factors in language production, just as they are in comprehension.

9.4.3 The Question of Modularity

The processing evidence cited so far also brings out the fact that people use all kinds of information – including nonlinguistic information – in processing language. Although this may strike some readers as unsurprising, it has been a highly controversial issue. Chomsky has long argued that the human language faculty is made up of numerous largely autonomous modules (see, for example, Chomsky 1981:135). Jerry Fodor's influential 1983 book *The Modularity of Mind* elaborated on this idea, arguing that the human mind comprised a number of distinct modules that are 'informationally encapsulated', in the sense that they have access only to one another's outputs, not to their internal workings.

The appeal of the modularity hypothesis stems primarily from two sources. The first is the analogy with physical organs: since various bodily functions are carried out by specialized organs (liver, kidney, pancreas, etc.), it seems plausible to posit similarly specialized mental organs to carry out distinct cognitive functions (vision, reasoning, language processing, etc.).[13] Second, it is generally good practice to break complex problems down into simpler, more tractable parts. This is common in building computer systems, and computational metaphors have been very influential in recent theorizing about the human mind. It was natural, therefore, to postulate that the mind has parts, each of which performs some specialized function. Fodor's version of the modularity hypothesis is not only that these mental organs exist, but that they function largely independently of each other.

According to this view, there should be severe limitations on how people combine information of different types in cognitive activities. Many psycholinguists would claim that the field has simply failed to detect such limitations, even when they use methods that can provide very precise information about timing (like the head-mounted eye tracker). These researchers would argue that linguistic processing appears to be opportunistic from start to finish, drawing on any kind of linguistic or nonlinguistic information that might be helpful in figuring out what is being communicated. Others working within the field would counter that the modularity hypothesis is not refuted by the existence of rapid information integration in sentence comprehension. Modularity can be reconciled with these results, it is argued, by assuming that informationally encapsulated language modules

[12] More precisely, at the beginnings of intonation units.

[13] The advocates of modularity are not entirely clear about whether they consider the language faculty a single mental organ or a collection of them. This is analogous to the vagueness of the notion of a physical organ: is the alimentary canal a single organ or a collection of them?

work at a finer grain than previously believed, producing partial results of a particular kind without consulting other modules. The outputs of these processors could then be integrated with other kinds of information relevant to comprehension quite rapidly. The controversy continues, hampered perhaps by a lack of general agreement about what counts as a module and what the space of hypotheses looks like in between Fodor's original strong formulation of the modularity hypothesis and the complete denial of it embodied in, for example, connectionist networks.

9.5 A Performance-Plausible Competence Grammar

Describing one of their eye-tracking experiments, Tanenhaus et al. write:

> [T]he instruction was interpreted incrementally, taking into account the set of relevant referents present in the visual work space....That information from another modality influences the early moments of language processing is consistent with constraint-based models of language processing, but problematic for models holding that initial linguistic processing is encapsulated. (1996:466)

More generally, language understanding appears to be a process of constraint satisfaction. Competing interpretations exist in parallel, but are active to varying degrees. A particular alternative interpretation is active to the extent that evidence is available to support it as the correct interpretation of the utterance being processed. Note, by the way, that frequency can also play a significant role here. One reason the *horse raced past the barn* example is such a strong garden path is that *raced* occurs much more frequently as a finite verb form than as the passive participle of the transitive use of *race*, which is precisely what the NOM-modifying reading requires. Ambiguity resolution is a continuous process, where inherent degrees of activation (e.g. those correlating with gross frequency) fluctuate as further evidence for particular interpretations become available. Such evidence may in principle stem from any aspect of the sentence input or the local or discourse context. A garden-path sentence is one that has an interpretation strongly supported by initial evidence that later turns out to be incorrect.

The next three subsections argue that the three defining properties of Constraint-Based Lexicalism, introduced in Section 9.3, receive support from available evidence about how people process language.

9.5.1 Surface-Orientation

Our grammar associates structures directly with the string of words that the listener hears, in the form (and order) that the listener hears them. This design feature of our grammar is crucial in accounting for the word-by-word (or even syllable-by-syllable) fashion in which sentence processing proceeds. We have seen that in utterances, hearers use their knowledge of language to build partial hypotheses about the intended meaning. These hypotheses become more or less active, depending on how plausible they are, that is, depending on how well their meaning squares with the hearers' understanding of what's going on in the discourse.

Sometimes the process even takes short-cuts. We have all had the experience of completing someone else's utterance (a phenomenon that is, incidentally, far more common than one might imagine, as shown, e.g. by Wilkes-Gibbs (1986)) or of having to wait for someone to finish an utterance whose completion had already been made obvious by context. One striking example of this is 'echo questions', as illustrated in the following kind of dialogue:

(53) [Speaker A:] Señora Maria Consuelo Bustamante y Bacigalupo is coming to dinner tomorrow night.
[Speaker B:] WHO did you say is coming to dinner tomorrow night?
****

In a dialogue like this, it is quite likely that Speaker A may comprehend the intent of Speaker B's utterance well before it is complete, somewhere in the region indicated by the asterisks. Presumably, this is possible precisely because Speaker A can recognize that the remainder of B's utterance is a repetition of A's own utterance and can graft that bit of content onto the partial analysis A has performed through word-by-word processing of B's utterance. What examples like this show is that a partial linguistic analysis (e.g. the partial linguistic analysis of *who did you*, *who did you say* or *who did you say is*) is constructed incrementally, assigned a (partial) interpretation, and integrated with information from the context to produce an interpretation of a complete utterance even before the utterance is complete. Amazing, if you think about it!

So if a grammar is to be realistic, that is, if it is to be directly embedded in a model of this kind of incremental and integrative language processing, then it needs to characterize linguistic knowledge in a way that allows for the efficient incremental computation of partial analyses. Moreover, the partial grammatical analyses have to be keyed in to partial linguistic meanings, because these are what interacts with other factors in processing.

The kind of grammar we are developing seems quite compatible with these performance-driven design criteria. The representation our grammar associates with each word provides information about the structure of the sentence directly, that is, about the phrases that the words are part of and about the neighboring phrases that they combine with syntactically. In addition, the words of our grammar provide partial information about the meaning of those phrases, and hence, since all phrases are built up directly from the component words and phrases in a context-free manner, there is useful partial semantic information that can be constructed incrementally, using our surface-oriented grammar.

It is not clear how to reconcile the incremental processing of utterances with transformational grammar, in which the surface ordering of elements depends on a sequence of structures and operations on them. If only the surface structures are involved in the processing model, then the transformational derivations are evidently irrelevant to performance. On the other hand, a full derivation cannot be available incrementally, because it necessarily involves all elements in the sentence.

Of course we have not actually spelled out the details of a performance model based on a grammar like ours, but the context-free-like architecture of the theory and the hybrid syntactic-semantic nature of the lexical data structures are very suggestive. Incremental computation of partial semantic structures, the key to modeling integrative sentence processing, seems to fit in well with our grammar.

9.5.2 Constraint-Based Grammar

Our grammar consists of a set of constraints that apply simultaneously to define which structures are well-formed. When this abstract model of language is applied (in a computational system, or in a model of human language processing), this simultaneity is cashed out as order independence: it doesn't matter which order the constraints are consulted in, they will always give the same collective result.

As noted above, the order of presentation of the words in an utterance largely determines the order of the mental operations listeners perform in comprehending it. However, words are associated with many different kinds of information, and the architecture of the theory does not impose any fixed order on which kind is used first. For example, it is not the case that syntactic information (e.g. agreement information that might rule out a particular parse) is always consulted before semantic information (e.g. semantic incompatibility that would favor or disfavor some potential interpretation of an utterance). In fact, it is possible to make an even stronger claim. In examples like (54), early accessing of morphological information allows the number of sheep under discussion to be determined incrementally, and well before the nonlinguistic knowledge necessary to select the 'fenced enclosure' sense of *pen*, rather than its 'writing implement' sense.

(54) The sheep that was sleeping in the pen stood up.

In (55), on the other hand, the relevant information about the world – that sheep might fit inside a fenced enclosure, but not inside a writing implement – seems to be accessed well before the relevant morphological information constraining the number of sheep:[14]

(55) The sheep in the pen had been sleeping and were about to wake up.

So the information accessed in on-line language processing is typically made available in an order determined by the input stream, not by the constructs of grammatical theory. In comprehending these sentences, for example, a hearer accesses morphological information earlier in (54) and later in (55) precisely because the order of access is tied fairly directly to the order of the words being processed. A theory positing a fixed order of access – for example, one that said all strictly linguistic processing must be completed before nonlinguistic knowledge could be brought to bear on utterance interpretation – would not be able to account for the contrast between (54) and (55).

Such a theory would also be incompatible with the evidence from the head-mounted eye-tracking studies cited earlier. Those studies show that listeners use both linguistic and visual information to determine a speaker's intended meaning, and they use it as soon as the information is available and helpful to them. Hence, a theory of linguistic comprehension must allow the order of access to information to remain flexible.

Finally, we know that for the most part linguistic information functions fairly uniformly in many diverse kinds of processing activity, including comprehension, production, translation, playing language games, and the like. By 'fairly uniformly' we mean that the set of sentences reliably producible[15] by a given speaker-hearer is similar – in fact bears a natural relation (presumably proper inclusion) – to the set of sentences that that speaker-hearer can comprehend. This might well have been otherwise. That there is so close and

[14]This pair of examples is due to Martin Kay.

[15]That is, sentences short enough to utter in a real language-use situation. We also intend to rule out production errors.

predictable a relation between the production activity and the comprehension activity of any given speaker of a natural language militates strongly against any theory on which the production grammar is independent from the comprehension grammar, for instance. This simple observation suggests rather that the differences between, say, comprehension and production should be explained by a theory that posits distinct processing regimes making use of a single language description. And that description should therefore be a process-neutral grammar of the language, which can serve each kind of process that plays a role in on-line linguistic activity.[16] Since production involves going from a meaning to an utterance and comprehension involves going from an utterance to a meaning, a grammar that is used in both processes should not favor one order over the other.

Grammars whose constructs are truly process-neutral, then, hold the most promise for the development of processing models. Transformational grammars aren't process-neutral, because transformational derivations have a directionality – that is, an ordering of operations – built into them. To interpret a transformational grammar as a model of linguistic knowledge, then, it is necessary to abstract away from its inherent directionality, obscuring the relationship between the grammar and its role in processing. This problem can be avoided by formulating a grammar as a declarative system of constraints. Such systems of constraints fit well into models of processing precisely because they are process-neutral.

What these observations add up to is a view of grammar as a set of constraints, each expressing partial information about linguistic structures, rather than a system employing destructive operations of any kind. Moreover, we have also seen that these constraints should exhibit certain further properties, such as order-independence, if performance-compatibility is to be achieved. The grammar we've been developing has just these design properties – all the constructs of the grammar (lexical entries, grammar rules, even lexical rules and our general principles) are nothing more than constraints that produce equivalent results no matter what order they are applied in.

9.5.3 Strong Lexicalism

Our theory partitions grammatical information into a number of components whose interaction determines the well-formedness of particular examples. By far the richest locus of such information, however, is the lexicon. Our grammar rules are simple in their formulation and general in their application, as are such aspects of our formal theory as the Head Feature Principle and the Valence Principle. Most of the details we need in order to analyze individual sentences are codified in the lexical entries (though much of it need not be stipulated, thanks to lexical rules and inheritance through the type hierarchy).

However, other divisions of grammatical labor are conceivable. Indeed, a number of theories with highly articulated rule systems and relatively impoverished lexicons have been developed in considerable detail (e.g. early transformational grammar and Generalized Phrase Structure Grammar, both of which are described briefly in Appendix B).

[16]The fact that comprehension extends beyond systematic production can be explained in terms of differences of process – not differences of grammar. Speakers that stray far from the grammar of their language run a serious risk of not being understood; yet hearers that allow grammatical principles to relax when necessary will understand more than those that don't. There is thus a deep functional motivation for the two kinds of processing to differ as they appear to.

We have argued for strong lexicalism on the basis of linguistic adequacy (along with general considerations of elegance and parsimony). It turns out that the psycholinguistic evidence on language processing points in the same direction. Investigations of syntactic ambiguity resolution in general and garden path effects in particular have shown that the choice of words can make a big difference. That is, the difficulty listeners exhibit in resolving such ambiguities (including overcoming garden paths) is influenced by factors other than the structure of the tree. Processing is critically affected by semantic compatibility and pragmatic plausibility, type and valence of the words involved, and the frequencies with which individual words occur in particular constructions. Our earlier discussion of eye-tracking studies describes some of the evidence to this effect, and there is considerably more (see Tanenhaus and Trueswell 1995 for a survey of relevant results).

To give another kind of example, a sentence beginning with the sequence NP_1–V–NP_2 can be continued in a number of ways. NP_2 could be the object of the verb, or it could be the subject of a complement sentence. This is illustrated in (56a), which can be continued as in (56b) or (56c):

(56) a. Lou forgot the umbrella ...
b. Lou forgot the umbrella was broken.
c. Lou forgot the umbrella in the closet.

Hence a listener or reader encountering (56a) must either postpone the decision about whether to attach the NP *the umbrella* to the VP, or decide prematurely and then potentially have to reanalyze it later. Either way, this places a burden on the parser in at least some cases. Various experimental paradigms have been used to verify the existence of this parsing difficulty, including measuring reading times and tracking the eye movements of readers.

However, not all verbs that could appear in place of *forgot* in (56a) can appear in both of the contexts in (56b) and (56c). This is illustrated in (57):

(57) a. Lou hoped the umbrella was broken.
b.*Lou hoped the umbrella in the closet.
c.*Lou put the umbrella was broken.
d. Lou put the umbrella in the closet.

The increased parsing load in (56a) is reduced greatly when the valence of the verb allows for no ambiguity, as in (57). This has been demonstrated via the methods used to establish the complexity of the ambiguity in the first place (see Trueswell et al. 1993). This provides strong evidence that people use valence information associated with words incrementally as they process sentences.

Similarly, listeners use semantic and pragmatic information about the verb and the following NP to choose between possible attachment sites for the NP. For example, though *learn* may take either an NP object or a sentential complement, as illustrated in (58),

(58) a. Dana learned the umbrella was broken.
b. Dana learned a new theorem in class.

when the immediately following NP is not the sort of thing one can learn, people do not exhibit the level of complexity effects in parsing that show up in (56).

The same sort of effect of lexical meaning on parsing shows up with PP attachment ambiguities, like those in (59):

(59) a. The artist drew the child with a pencil.
b. Lynn likes the hat on the shelf.

In (59a), the pencil could be either the artist's instrument or something in the child's possession; in (59b), *on the shelf* could identify either Lynn's preferred location for the hat, or which hat it is that Lynn likes. The structural ambiguity of such sentences causes parsing complexity, but this is substantially mitigated when the semantics or pragmatics of the verb and/or noun strongly favors one interpretation, as in (60):

(60) a. The artist drew the child with a bicycle.
b. Lynn bought the hat on the shelf.

In short, lexical choices have a substantial influence on processing. Moreover, the information that we have been led to posit in our lexical entries has independently been found to play a role in language processing. After reviewing a number of studies on the factors that influence syntactic ambiguity resolution, MacDonald et al. (1994) discuss what information they believe needs to be lexically specified to account for the psycholinguistic results. Their list includes:

- valence;
- 'coarse-grained semantic information' (i.e. the sort of information about who did what to whom that is given in our SEM feature); and
- 'grammatically relevant features' such as 'tense..., finiteness..., voice (active or passive), number..., person..., and gender...'.

They also mention grammatical category, which we represent in our lexical entries by means of types (specifically, the subtypes of *pos*). In short, the elements in the MacDonald et al. list correspond remarkably well to the information that we list in our lexical entries.

9.5.4 Summary

In this section we have seen how the design features of our grammar are supported by evidence from language processing. A grammar must be SURFACE-ORIENTED to account for the incremental and integrative nature of human language processing. The fact that different kinds of linguistic information and even non-linguistic information are accessed in any order, as convenient for the processor, suggests a CONSTRAINT-BASED design of grammar. This is further motivated by the process-neutrality of knowledge of language. Finally, STRONG LEXICALISM and the particular kinds of information associated with words in our lexical entries are supported by psycholinguistic evidence from garden paths, eye-tracking experiments, and tests of parsing complexity.

9.6 Universal Grammar: A Mental Organ?

In the preceding section we have argued that the design features of our grammatical theory comport well with existing evidence about how people process language. There is yet another psycholinguistic consideration that has played a central role in much work in generative grammar, namely, learnability. In this section, we briefly address the question of evaluating our theory by this criterion.

As noted in Chapter 1, Chomsky has argued that the most remarkable fact about human language – and the one he thinks linguists should be primarily concerned with explaining – is that virtually all children become fluent speakers of a language, with little apparent effort or instruction. The puzzle, as Chomsky sees it, is how people can come to know so much about language so quickly and easily. His solution in a nutshell is that people's knowledge of language is for the most part innate, not learned. This entails that much linguistic structure – namely, those aspects that are innate – must be common to all languages. Consequently, a central goal of much work in modern syntactic theory has been to develop a conception of universal grammar rich enough to permit the descriptions of particular languages to be as simple as possible.

Chomsky's strong claims about the role of innate knowledge in language acquisition are by no means uncontroversial among developmental psycholinguists. In particular, many scholars disagree with his position that the human language faculty is highly task-specific – that is, that people are born with a 'mental organ' for language which is distinct in its organization and functioning from other cognitive abilities (see, for example, Bates and MacWhinney 1989, Tomasello 1992 and Elman et al. 1996 for arguments against Chomsky's position; but see also Hauser et al. 2002).

There can be little doubt that biology is crucial to the human capacity for language; if it were not, family pets would acquire the same linguistic competence as the children they are raised with. There is no doubt that humans are quite special, biologically, though the details of just what is special remain to be worked out. It is far less clear, for example, that the human capacity for language is as independent of other systems of knowledge as has sometimes suggested. A range of views on this issue are possible. At one end of the spectrum is the idea that the language faculty is a fully autonomous module, unrelated to general cognitive capacity. At the other end is the idea that there are no specifically linguistic abilities – that our capacity to learn language arises essentially as a side-effect of our general intelligence or of other abilities. Chomsky's view is close to the former;[17] Tomasello (1992) argues for something close to the latter. Other scholars have defended views somewhere in between.

The participants in this debate often seem to be talking past one another. Opponents of task-specificity tend to take a simplistic view of linguistic structure, emphasizing basic communicative functions while ignoring the intricacies of syntax that are the bread and butter of generative grammar. On the other hand, proponents of task-specificity have a tendency to leap from the complexity of their analyses to the conclusion that the knowledge involved must be innate and unique to language.

We find much of the argumentation on both sides of this controversy unconvincing, and hence we take no position in this book. Nevertheless, the theory presented here can contribute to its resolution. Explicit syntactic and semantic analyses can facilitate more precise formulations of what is at issue in the debate over task-specificity. Moreover, formal representations of data structures and their interactions makes it possible to see more clearly where there could be analogues in other cognitive domains. Our position is that the grammatical constructs we have been developing in this text are well suited to a theory of universal grammar, whether or not that theory turns out to be highly task-specific, and that the explicitness of our proposals can be helpful in resolving the

[17] But see Hauser et al. 2002 for what seems to be a striking switch in Chomsky's position.

task-specificity question.

To justify this claim, we will consider various components of our theory, namely: the phrase structure rules, the features and their values, the type hierarchy with its feature declarations and constraints, the definition of phrasal licensing (incorporating the Head Feature Principle, the Valence Principle, and the two semantic principles), the Binding Theory, and the lexical rules. We will find that most of these have elements that are very likely universal, and that our formulations do not prejudge the issue of task-specificity.

Phrase Structure Rules Our grammar rules (with the exception of the Imperative Rule) are sufficiently general that, aside from their linear ordering of the constituents, they are natural candidates for universality. It would not be hard to factor out the ordering, so that versions of these rules could be posited as part of universal grammar.

The sort of hierarchical structure induced by the rules, which we represent with trees, is arguably not unique to language: it also seems appropriate, for example, to aspects of mathematical reasoning. On the other hand, the concepts of 'head', 'complement', 'specifier', and 'modifier', which are crucial to our formulation of the rules, appear to be specialized to language. If it should turn out, however, that they can be shown to be instances of some more generally applicable cognitive relations, this would in no way undermine our analysis.

Features and Values Most of the features we have posited have obvious cross-linguistic application. It seems at least plausible that a more fully worked out version of the theory presented here could include an inventory of features from which the feature structures of all languages must be constructed. In later chapters, we will identify the values of some features with particular English words, a practice inconsistent with saying that the set of possible feature values is part of universal grammar. It might be possible, however, to restrict feature values to come from either the set of morphological forms of the language or a universally specifiable set.

Some features (e.g. PER, GEND, COUNT) clearly reflect properties of the world or of human thought, whereas others (e.g. CASE, FORM) seem specifically linguistic. Our treatment is neutral on the question of whether grammatical features will ultimately be reducible to more general aspects of cognition, though the general data type of features with values certainly has applications beyond linguistics.

Types and the Type Hierarchy The types we have proposed could arguably be drawn as well from a fixed universal inventory. The feature declarations associated with the types are likewise probably quite similar across languages. The constraints introduced by some types (such as SHAC), on the other hand, appear to be more specific to the particular language. Some of the (subtype and supertype) relations in the type hierarchy (e.g. that *siv-lxm* is a subtype of *verb-lxm*) are surely universal, whereas others (e.g. the hierarchy of subtypes of *agr-cat*) may vary across languages.

Our types are arranged in a default inheritance hierarchy, a kind of structure that very likely plays an important role in how people organize many kinds of information. Indeed, the use of such hierarchies in linguistics was inspired by earlier work in artificial intelligence, which suggested this sort of structure for taxonomies of concepts. The particular types we have posited appear task-specifically linguistic, though we leave open the possibility that some of them may be more general.

Phrasal Licensing Our definition of phrasal licensing involves both universal and English-specific elements. As noted earlier, the Argument Realization Principle may well differ across languages. And clearly, the Case Constraint as we have formulated it applies only to English. On the other hand, the Head Feature Principle and the two semantic principles are intended to apply to all languages.

Some parts of the phrasal licensing definition make reference to specifically linguistic constructs (such as grammar rules, heads, and particular features), but the idea of unifying information from diverse sources into a single structure has nonlinguistic applications as well.

Binding Theory All languages evidently have some binding principles, and they are quite similar. Characteristically, there is one type of element that must be bound within a local domain and another type that cannot be locally bound. But there is cross-language variation in just what counts as 'local' and in what can serve as the antecedents for particular elements. Our particular Binding Theory is thus not part of universal grammar. Ideally, a grammatical theory would delineate the range of possible binding principles, of which the ones presented in Chapter 7 would be instances.

While these principles appear to be quite language-specific, it is conceivable that they might be explained in terms of more general cognitive principles governing identity of reference.

Lexical Rules The lexical rules presented in the previous chapter are clearly parochial to English. However, our characterizations of derivational, inflectional, and post-inflectional lexical rules seem like plausible candidates for universality. More generally, our formulation of lexical rules as feature structures lays the groundwork for developing a more articulated inheritance hierarchy of types of lexical rules. Although formulating a general theory of what kinds of lexical rules are possible is beyond the scope of this book, our grammatical framework has a way of expressing generalizations about lexical rules that are not language-particular.

The contents of these rules are quite specific to language, but their general form is one that one might expect to find in many domains: if a database contains an object of form X, then it also contains one of form Y.

To sum up this superficial survey of the components of our theory: it contains many elements (the grammar rules, the definition of Well-Formed Tree Structure, the features and types) that are plausible candidates for playing a role in a theory of universal grammar. Moreover, some elements (the binding principles, some lexical rules) probably have close analogues in many other languages. Although our central purpose in this book is to present a precise framework for the development of descriptively adequate grammars for human languages, rather than to account for the puzzle of language learnability through the development of a theory of universal grammar, the framework we have presented here is nevertheless quite compatible with the latter goal.

Further, our grammatical theory suggests a number of parallels between the kinds of information structures needed to account for linguistic competence and those employed in other cognitive domains. However, we need not commit ourselves on the question of task-specificity; rather, we offer the hope that increasingly precise linguistic descriptions like those that are possible within the framework developed here will help to clarify the nature of this controversy and its resolution.

9.7 Summary

Chomsky's famous distinction between knowledge of language ('competence') and use of language ('performance') has allowed syntacticians to concentrate on relatively tractable problems, by abstracting away from many features of the way people actually speak. But most generative grammarians agree that an optimal theory of competence will play a role in explaining many features of linguistic performance. To the extent that a theory of grammar attains this ideal, we call it 'realistic'.

We have argued in this chapter that the theory we are developing in this book does well by this criterion. Our theory, by virtue of being surface-oriented, constraint-based, and strongly lexicalist, has properties that fit well with what we know about how people process utterances and extract meaning from them. Our understanding of the mechanisms that underlie linguistic performance is incomplete at present, and many of the points discussed in this chapter remain controversial. Nevertheless, a preliminary examination of what is known about processing provides grounds for optimism about our approach to syntactic theory. Considerations of learnability also support such a favorable assessment.

9.8 Further Reading

Many of the issues raised in this chapter are discussed at a relatively elementary level in the essays in Gleitman and Liberman 1995. Important discussions of issues raised in this chapter can be found in the following works: Chomsky 1965, Bever 1970, Bates and MacWhinney 1989, Tomasello 1992, MacDonald et al. 1994, Pinker 1994, Tanenhaus and Trueswell 1995, Elman et al. 1996, Marcus 2001, Jackendoff 2002, Hauser et al. 2002, and Marcus 2004.

9.9 Problems

Problem 1: Inflectional Lexical Rules With No Morphological Effect
The Singular Noun Lexical Rule, the Non-3rd-Singular Verb Lexical Rule, and the Base Form Lexical Rule are all inflectional lexical rules (that is, rules of type *i-rule*) which have no effect on the shape (i.e. the phonology) of the word.

A. Explain why we need these rules anyway.
B. Each of these rules have lexical exceptions, in the sense that there are lexemes that idiosyncratically don't undergo them. Thus, there are some nouns without singular forms, verbs without non-third-person singular present tense forms, and verbs without base forms. List any you can think of. [*Hint: The nouns without singular forms are ones that must always be plural; these aren't too hard to think of. The exceptional verbs are much harder to come up with; we only know of two (fairly obscure) exceptions to the Non-3rd-Singular Verb Lexical Rule and a small (though frequently used) class of exceptions to the Base Form Lexical Rule. In short, parts of this problem are hard.*]

10

The Passive Construction

10.1 Introduction

Perhaps the most extensively discussed syntactic phenomenon in generative grammar is the English passive construction. The active/passive alternation provided one of the most intuitive motivations for early transformational grammar, and it has played a role in the development of almost all subsequent theories of grammar.

In this chapter, we present an account of the English passive using the formal mechanisms we have developed in this text. Given the strongly lexical orientation of our theory, it should come as no surprise that we treat the active/passive relationship primarily as a relationship between two verb forms, and that we use a lexical rule to capture the generality of that relationship.

We begin with some data to exemplify the phenomenon in question. We then formulate our rule and explain how it works. Finally, we turn to the question of the status of the forms of the verb *be* that characteristically occur in passive sentences.

10.2 Basic Data

Consider sets of sentences (and nonsentences) like the following:

(1) a. The dog bit the cat.
b. The cat was bitten (by the dog).
c.*The cat was bitten the mouse (by the dog).

(2) a. Pat handed Chris a note.
b. Chris was handed a note (by Pat).
c.*Chris was handed Sandy a note (by Pat).

(3) a. TV puts dumb ideas in children's heads.
b. Dumb ideas are put in children's heads (by TV).
c.*Dumb ideas are put notions in children's heads (by TV).

The b-sentences in (1)–(3) are what are standardly called 'passive'; the a-sentences are referred to as their 'active' counterparts. There is clearly a close semantic relationship between active and passive pairs. In particular, the semantic roles of the arguments are the same – in (1), the dog is the biter, and the cat is the one being bitten. To put it informally, in an active sentence and its passive counterpart, 'who does what to whom' is the same. The crucial difference between active and passive sentences is that the subject of the passive corresponds to the object of the active. The participant denoted by the subject of the active, if expressed at all in the passive, is referred to by the object of the preposition *by*. Consequently, the verb in a passive sentence always has one less object (that is, NP complement) than the verb in its active counterpart. This is illustrated in the c-sentences of (1)–(3). It follows that sentences with intransitive verbs, like (4a), normally do not have passive counterparts, as in (4b):

(4) a. The patient died.
b.*The patient was died (by the doctor).
c.*The doctor died the patient.

Moreover, aside from this one difference, active verbs and their corresponding passives have identical valence requirements. This is illustrated in (5), where the absence of an obligatory complement renders both the active and passive examples ungrammatical:

(5) a. Pat handed Chris *(a note).
b. Chris was handed *(a note) (by Pat).
c. TV puts dumb ideas *(into their heads).
d. Dumb ideas are put *(into their heads) (by TV).

10.3 The Passive Lexical Rule

It would not be hard to formulate lexical entries for passive forms of verbs. To capture the generalizations stated informally above, however, we need to formulate a rule that can relate actives and passives. As was the case with the rules discussed in Chapter 8, our passive rule is motivated by more than just parsimony. Faced with novel transitive verbs – either new coinages like *email* or rare words like *cark* – English speakers can (and often do) immediately use them correctly in passive sentences. Hence a rule-governed treatment of the active/passive alternation will be psychologically more realistic than a mere listing of the passive forms for all transitive verbs.

Intuitively, then, we want a rule that does the following:

- turns the first NP complement into the subject;
- allows the subject either to turn into the object of a PP headed by *by* or to be omitted altogether;
- leaves the valence features otherwise unchanged;
- leaves the semantics unchanged; and
- makes the appropriate morphological change in the form of the verb.

This last item is one we have not mentioned until this point. A moment's reflection should reveal that the morphology of the passive form of a verb (or 'passive participle', as it is commonly called) is always identical to that of the past participle; this is especially clear if we consider verbs with exceptional past participles, such as *do* (*done*), *sink* (*sunk*) and *cut* (*cut*). This generalization is captured easily in our framework by invoking the same morphological function, F_{PSP}, for both the Past Participle Lexical Rule and the Passive Lexical Rule.

Before writing the Passive Lexical Rule, we need to decide what type of *l-rule* it is. The morphology of English passives is inconclusive on this point: no further affixes attach to passives. As far as the morphology is concerned, the rule could be either an *i-rule* or a *d-rule*. However, the syntactic aspects of passive are only consistent with the constraints on *d-rules*. Recall from Chapter 8 that the constraints on inflectional rules (*i-rule*s) and derivational rules (*d-rules*) are as in (6) and (7), respectively.

(6)
$$
\textit{i-rule} : \begin{bmatrix} \text{INPUT} & \left\langle \text{X} \;, \begin{bmatrix} \textit{lexeme} & \\ \text{SYN} & \boxed{3} \\ \text{ARG-ST} & \boxed{\text{A}} \end{bmatrix} \right\rangle \\ \\ \text{OUTPUT} & \left\langle \text{Y} \;, \begin{bmatrix} \textit{word} & \\ \text{SYN} & \boxed{3} \\ \text{ARG-ST} & \boxed{\text{A}} \end{bmatrix} \right\rangle \end{bmatrix}
$$

(7)
$$
\textit{d-rule} : \begin{bmatrix} \text{INPUT} & \left\langle \text{X} \;, \begin{bmatrix} \textit{lexeme} & \\ \text{SYN} & /\boxed{3} \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \text{Y} \;, \begin{bmatrix} \textit{lexeme} & \\ \text{SYN} & /\boxed{3} \end{bmatrix} \right\rangle \end{bmatrix}
$$

In order to change the subject and complements, the passive rule must specify either different SPR and COMPS values or different ARG-ST values on the INPUT and OUTPUT. The passive rule given immediately below specifies different ARG-ST values, but either strategy would be inconsistent with the constraints on *i-rule*. Therefore, given our theory of inflectional and derivational rules, passive must be a derivational rule.[1]

The following is a lexical rule that satisfies the desiderata given above:

(8) Passive Lexical Rule
$$
\begin{bmatrix} \textit{d-rule} & \\ \text{INPUT} & \left\langle \boxed{1} \;, \begin{bmatrix} \textit{tv-lxm} & \\ \text{ARG-ST} & \langle \, [\text{INDEX } i] \, \rangle \oplus \boxed{\text{A}} \end{bmatrix} \right\rangle \\ \text{OUPUT} & \left\langle \text{F}_{PSP}(\boxed{1}) \;, \begin{bmatrix} \textit{part-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & [\text{FORM} \;\; \text{pass}] \end{bmatrix} \\ \text{ARG-ST} & \boxed{\text{A}} \oplus \left\langle \left(\begin{matrix} \text{PP} \\ \begin{bmatrix} \text{FORM} & \text{by} \\ \text{INDEX} & i \end{bmatrix} \end{matrix} \right) \right\rangle \end{bmatrix} \right\rangle \end{bmatrix}
$$

There are several points of explanation that need to be made here.

[1] French again confirms this conclusion: There are four inflected forms of any given passive participle, the choice depending on the number and gender of the participle's subject NP. This indicates that the passivization rule in French feeds into various inflectional rules, and hence must be derivational.

First, like the present and past participle lexical rules, the OUTPUT of this rule is of type *part(iciple)-lxm*. This is a subtype of *const-lxm*, so passive participles, like other participles, undergo the Constant Lexeme Lexical Rule. The only effect of the Constant Lexeme Lexical Rule is to change the type of the second member of the lexical sequence to *word*. The type *word*, however, is constrained to satisfy the Argument Realization Principle. As such, OUTPUTs of the Constant Lexeme Lexical Rule will be subject to the Argument Realization Principle (Chapter 7).

Second, notice that most of the effects of the rule (which applies to any lexeme belonging to a subtype of *tv-lxm*) are in the ARG-ST. At a coarse level of description, what the rule does is rearrange the elements of the ARG-ST list. Because of the ARP, these rearrangements also affect the values of the valence features. Specifically, (8) makes the second element (corresponding to the direct object) of the input ARG-ST list be the first element (corresponding to the subject) of the output's ARG-ST list. Whatever follows the second element in the input also moves up in the list. (8) also adds a PP to the end of the ARG-ST list. The specification [FORM by] on this PP indicates that the PP must be headed by the preposition *by*. We will abbreviate 'PP[FORM by]' as 'PP[by]' (and similarly with other values of FORM). Hence a verbal lexeme with an argument structure like (9a) will give rise to a passive lexeme whose argument structure is (9b):

(9) a. $\begin{bmatrix}\text{ARG-ST} & \langle\ \text{NP}_i\ ,\ \text{NP}_j\ ,\ \text{PP[to]}\ \rangle\end{bmatrix}$ (*send*, *give*, *fax*...)

b. $\begin{bmatrix}\text{ARG-ST} & \langle\ \text{NP}_j\ ,\ \text{PP[to]}\ (,\ \text{PP[by]}_i)\ \rangle\end{bmatrix}$ (*sent*, *given*, *faxed*...)

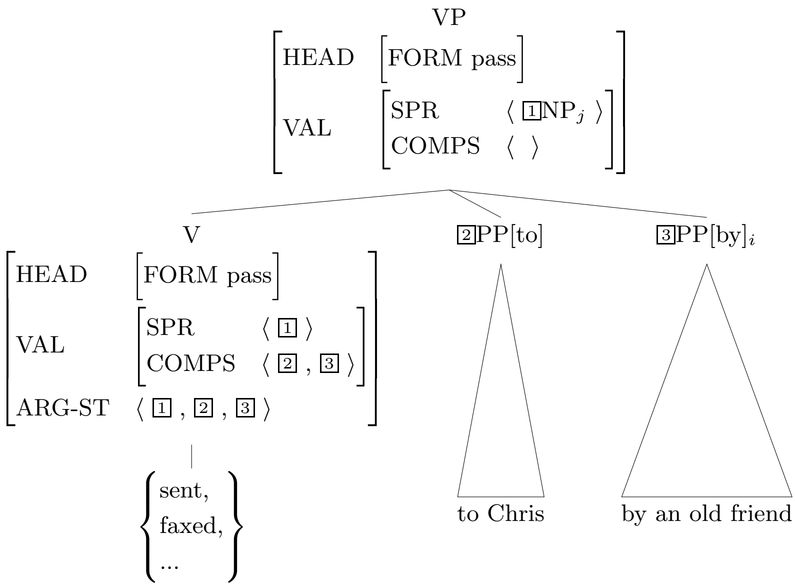
After going through the Constant Lexeme Lexical Rule, (9b) licenses two basic kinds of word structure, both constrained by the ARP. These are shown in (10):

(10) a. $\begin{bmatrix}\text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \boxed{1}\ \rangle \\ \text{COMPS} & \langle\ \boxed{2}\ \rangle\end{bmatrix} \\ \text{ARG-ST} & \langle\ \boxed{1}\text{NP}_j\ ,\ \boxed{2}\text{PP[to]}\ \rangle\end{bmatrix}$ (*sent*, *given*, *faxed*...)

b. $\begin{bmatrix}\text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \boxed{1}\ \rangle \\ \text{COMPS} & \langle\ \boxed{2}\ ,\ \boxed{3}\ \rangle\end{bmatrix} \\ \text{ARG-ST} & \langle\ \boxed{1}\text{NP}_j\ ,\ \boxed{2}\text{PP[to]}\ ,\ \boxed{3}\text{PP[by]}_i\ \rangle\end{bmatrix}$ (*sent*, *given*, *faxed*...)

Hence passive words will automatically give rise to passive VPs like (11), thanks to the Head-Complement Rule (and the HFP and the Valence Principle):

(11)



In other words, once our lexicon has passive words, our grammar already guarantees that we will have the appropriate passive VPs. These VPs can be selected as a complement by a few verbs, most notably *be*:

(12) A message [was [sent to Chris by an old friend]].

A third noteworthy property of the Passive Lexical Rule concerns indices. Recall that subscripts indicate values of the feature INDEX; so (8) says that the optional PP[by] in the rule output has an index that is coindexed with the subject of the lexical rule input. This means that whatever semantic role the verbal lexeme assigns to its subject will be assigned to the INDEX value of the PP[by] of the passive word, and hence (since *by* is an argument-marking preposition) to the prepositional object within the PP[by] (see below). Likewise, since the verbal lexeme's object – the first element in the list $\boxed{A}$ – is identified with the subject of the passive word, it follows that the index of the subject of the passive word is the same as that of the verbal lexeme's direct object. Therefore, since the semantics remains unchanged by this lexical rule (because the rule says nothing to override the effect of the defeasible identity constraint), the semantic role of the active object will be the same as that of the passive subject. The overall result of this rule, then, is to shift the role assignments from subject to PP[by] and from object to subject.

Fourth, note that the passive rule does not mention case at all. Verbal lexemes do not specify CASE values for any of their arguments (in English); hence, though the lexeme's object NP becomes the subject of the corresponding passive participle, there is no need to 'unassign' an accusative case specification. All nonsubject arguments of verbs must be accusative, but the constraint that guarantees this (namely, the Case Constraint – see Chapter 8, Section 8.4.5) applies to lexical trees (word structures), not to lexemes. (See the definition of lexical licensing in Chapter 9, Section 9.2.7.) Nor does the passive rule assign nominative case to the first argument of the rule output, as one might expect on the basis of examples like (13):

(13) a. He was arrested by the police.
b.*Him was arrested by the police.

The nominative case of the subject in examples like (13) is determined by the auxiliary verb *was*, whose SPR value is identified with that of the passive VP, as discussed in the next section. There are in fact instances of passive verbs whose subjects are not nominative, as in (14).

(14) $\left\{\begin{array}{c}\text{Him}\\ \text{*He}\end{array}\right\}$ being arrested by the police upset many people.

Our passive rule achieves the desired effect in such instances by leaving the subject of the passive word unspecified for CASE. Hence, whatever case requirements the particular grammatical context imposes will determine the CASE value of a passive verb's subject.[2]

Fifth, the rule says that passive verbs are constrained to be [FORM pass].[3] The justification for having a separate value 'pass' for the FORM of passive verbs has not yet been provided; this will be addressed in the next section.

Returning to the use of the FORM feature on the PP in (8), recall that FORM has so far been used primarily for distinguishing among verb forms. But in the Agent Nominalization Lexical Rule presented in Chapter 8, we already made use of the FORM feature on PPs: a PP specified as [FORM of] was meant to be one that could only be headed by the preposition *of*. In fact, we want to employ the feature FORM more generally, to mark the choice of preposition in other contexts as well. Since the set of prepositions in English is a relatively small, closed set, we might (in the limiting case) have a separate value of FORM for each preposition. In this book, we'll use only the following FORM values for prepositions:

(15) of, by, to, at, in, about, on, for

Having FORM values for prepositions allows us, for example, to represent the fact that the verb *rely* requires a PP complement headed by either *on* or *upon*. The FORM value of the lexical preposition will be shared by the entire PP (since FORM is a head feature and hence is governed by the Head Feature Principle), as shown in the tree for a *by*-phrase sketched in (16):

[2] Verbal gerunds like *being* in (14), for example, might lexically specify the case of their subject (which is identified with the subject of the passive participle in (14)).

[3] Note that the passive rule, like other lexical rules applying to verbs, isn't changing the FORM value, but rather further specifying it, as verbal lexemes are generally underspecified for FORM.

(16)

$$\text{PP}\left[\begin{array}{ll}\text{SYN} & \left[\begin{array}{ll}\text{HEAD} & \boxed{2}\left[\text{FORM} \quad \text{by}\right] \\ \text{VAL} & \left[\text{COMPS} \quad \langle\ \rangle\right]\end{array}\right] \\ \text{SEM} & \left[\begin{array}{ll}\text{MODE} & \boxed{3} \\ \text{INDEX} & \boxed{4}\end{array}\right]\end{array}\right]$$

$$\text{P}\left[\begin{array}{ll}\text{SYN} & \left[\begin{array}{ll}\text{HEAD} & \boxed{2} \\ \text{VAL} & \left[\text{COMPS} \quad \langle \boxed{1} \rangle\right]\end{array}\right] \\ \text{SEM} & \left[\begin{array}{ll}\text{MODE} & \boxed{3} \\ \text{INDEX} & \boxed{4}\end{array}\right]\end{array}\right] \qquad \boxed{1}\text{NP}\left[\begin{array}{ll}\text{MODE} & \boxed{3} \\ \text{INDEX} & \boxed{4}\end{array}\right]$$

by themselves/them

Crucially, we assume *by* is an argument-marking preposition whose INDEX and MODE values are identified with those of its NP object. Thus whatever index the passive participle assigns to the PP[by] complement will be identified with the index of the NP object within that PP.

The effect of the Passive Lexical Rule, then, is to map lexemes like (17) into lexemes like (18):[4]

(17)

$$\left\langle \text{love}\ ,\ \left[\begin{array}{ll}\multicolumn{2}{l}{\textit{stv-lxm}} \\ \text{SYN} & \left[\begin{array}{ll}\text{HEAD} & \left[\begin{array}{ll}\multicolumn{2}{l}{\textit{verb}} \\ \text{AGR} & \boxed{1}\end{array}\right] \\ \text{VAL} & \left[\text{SPR} \quad \langle \left[\text{AGR}\ \boxed{1}\right] \rangle\right]\end{array}\right] \\ \text{ARG-ST} & \langle\ \text{NP}_i\ ,\ \text{NP}_j\ \rangle \\ \text{SEM} & \left[\begin{array}{ll}\text{INDEX} & s \\ \text{RESTR} & \left\langle\left[\begin{array}{ll}\text{RELN} & \textbf{love} \\ \text{SIT} & s \\ \text{LOVER} & i \\ \text{LOVED} & j\end{array}\right]\right\rangle\end{array}\right]\end{array}\right]\right\rangle$$

[4](17)–(19) represent families of lexical sequences, each of which contains more information than is shown. The optionality of the PP in (18) and (19) is just another kind of underspecification in the description. Each of the fully resolved lexical sequences that make up these families will have a fully resolved value for ARG-ST. Some will have ARG-ST values with the PP and some will have ARG-ST values without it.

(18)

$$\left\langle \text{loved}\ ,\ \begin{bmatrix} \textit{part-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \boxed{1} \\ \text{FORM} & \text{pass} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \text{NP}_j \left(, \begin{matrix} \text{PP} \\ \begin{bmatrix} \text{FORM} & \text{by} \\ \text{INDEX} & i \end{bmatrix} \end{matrix} \right) \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{love} \\ \text{SIT} & s \\ \text{LOVER} & i \\ \text{LOVED} & j \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

The Constant Lexeme Lexical Rule then maps lexemes like (18) into words like (19):

(19)

$$\left\langle \text{loved}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \boxed{1} \\ \text{FORM} & \text{pass} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \boxed{2}[\text{AGR}\ \boxed{1}] \rangle \\ \text{COMPS} & \boxed{B} \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \boxed{2}\text{NP}_j \right\rangle \oplus \boxed{B} \left\langle \left(, \begin{matrix} \text{PP} \\ \begin{bmatrix} \text{FORM} & \text{by} \\ \text{INDEX} & i \end{bmatrix} \end{matrix} \right) \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{love} \\ \text{SIT} & s \\ \text{LOVER} & i \\ \text{LOVED} & j \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Note that the effect of the ARP is seen in (19), since these lexical sequences involve *word*s.

10.4 The Verb *Be* in Passive Sentences

What about the forms of *be*, which in all of our examples (so far) immediately precede the passive participle? The first thing to observe is that passive participles can also occur in environments that lack any form of *be*. Some examples are given in (20):

(20) a. The cat got bitten (by the dog).
b. Liked by many people but respected by few, Jean will have to run an aggressive reelection campaign.
c. Anyone handed a note will be watched closely.

Hence, though some form of *be* is typical in passive sentences, it would have been a mistake to try to build it into the rule introducing the passive form of verbs. Rather, we need to provide an analysis of the relevant lexical entry for *be* that links its occurrence to the presence of a passive participle.[5]

More precisely, our analysis needs to say that the passive *be* takes a complement that is a VP[FORM pass] like the one shown in (11) above. This means that the ARG-ST list of the lexeme *be* contains both an NP subject and a VP[FORM pass]. A few points are worth noting here. First, this is the first time we have considered VP arguments/complements in detail, though our Head-Complement Rule permits them, as we saw earlier (see Section 8.5.1 of Chapter 8). We will see many more examples of VP complements soon. Second, since FORM is a head feature, a verb's FORM value will show up on its mother VP node. Hence if a verb like *be* selects a VP[FORM pass] complement, that is sufficient to guarantee that the complement's head daughter will be a V[FORM pass].

The trickiest and most important aspect of our analysis of *be* in passives is how we deal with the subject (i.e. with the value of SPR). In a sentence like (1b), repeated here as (21a), the agreement indicates that *the cat* should be treated as the subject (that is, the SPR) of *was*:

(21) a. The cat was bitten by the dog.
b.*The cat were bitten by the dog.

This is further supported by the unacceptability of (21b). But in our discussion of passive participles in the previous section, we discussed *the cat* as the subject of *bitten*. This was necessary for semantic reasons (i.e. to ensure that the cat functions semantically as the thing bitten, rather than as the biter), and to capture the correspondence between the valence values of the active and passive forms.

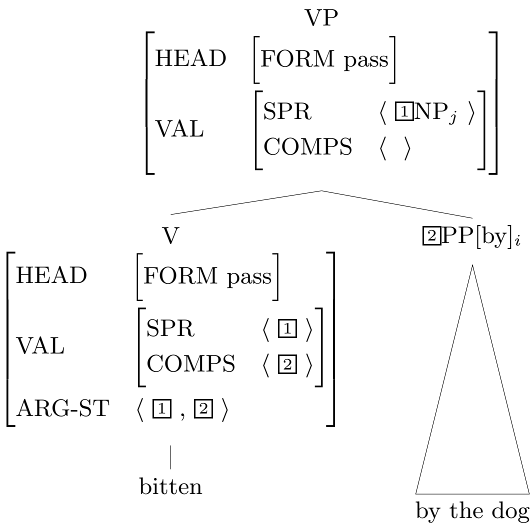
Our analysis provides a unified account of both these observations by identifying the subject of *be* with the subject of the passive verb. That is, there is only one subject NP in the sentence, but it is identified with the first member of the ARG-ST list of both *be* and the passive verb. As the subject of *be*, it is required to satisfy the agreement constraints imposed by the relevant inflected form of *be*, i.e. *was* in (21a). As the subject of the passive verb, it will also be assigned the semantic role that the object NP would take in an active sentence (the BITTEN role, rather than the BITER role that an active

[5]We'll return to the issue of whether we can analyze other uses of *be* in terms of this same lexical entry in Chapter 11.

form of *bite* would assign to its subject).

How exactly do we identify the subject of *was* with the subject of the passive verb *bitten*? First of all, it is important to see that half the job has already been accomplished by the Valence Principle, which requires that in a structure like (22), the SPR value of the passive verb is identical with that of the passive VP:

(22)



To represent the fact that *be* and its passive VP complement share the same subject, we need only add a constraint (using the familiar device of tagging) which specifies that the first argument of *be* (its subject) is identical to the SPR value of its VP[FORM pass] argument. We can now formulate the lexical entry for the passive *be* as follows:

(23)

$$\left\langle \text{be} , \begin{bmatrix} \textit{be-lxm} \\ \text{ARG-ST} \left\langle \boxed{1} , \begin{bmatrix} \text{SYN} \begin{bmatrix} \text{HEAD} \begin{bmatrix} \textit{verb} \\ \text{FORM} \;\; \text{pass} \end{bmatrix} \\ \text{VAL} \begin{bmatrix} \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle \; \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} \begin{bmatrix} \text{INDEX} \;\; s \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{SEM} \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \langle \; \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

What this entry says is that *be* belongs to a new type *be-lxm* (a subtype of *verb-lxm* whose properties do not yet concern us) and takes a VP argument specified as [FORM pass]. In addition, this *be* says that its subject must be the same as its complement's subject. This means that the subject of the sentence will also serve as the subject of the verb that heads the complement VP, according to the Valence Principle. And because *be* adds nothing to the meaning except the information that the complement's INDEX value is the same as that of *be*, (23) also guarantees that the semantics of the verb phrase headed by *be* is identical to the semantics of *be*'s VP complement. (Note that *be-lxm* inherits the constraint [MODE prop] from the type *verb-lxm*.)

We will see in the next two chapters that the idea of having a verb and its argument share a subject is extremely useful in describing a number of phenomena. In Chapter 13, we will see in addition how using lexical types can simplify lexical entries such as these.

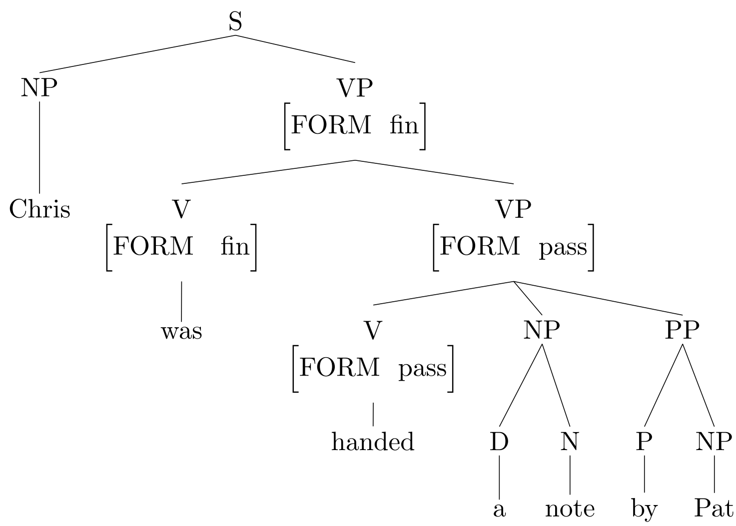
Exercise 1: Shared Subjects
Why doesn't the lexical entry in (23) license sentences like (i)?

(i)*A cat was a cat bitten by the dog.

10.5 An Example

We conclude this chapter with a detailed analysis of example (2b). The phrase structure we need to license is the following:

(24)



In this phrase structure, the word *was* is part of a family of lexical sequences constrained as shown in (25):

(25)

$$\left\langle \text{was}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \boxed{5} \\ \text{FORM} & \text{fin} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \boxed{1}[\text{AGR}\ \boxed{5}]\ \rangle \\ \text{COMPS} & \langle\ \boxed{2}\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \boxed{1}\begin{bmatrix} \text{AGR} & \textit{3sing} \\ \text{CASE} & \text{nom} \end{bmatrix} ,\ \boxed{2}\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{pass} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \boxed{1}\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \\ \text{INDEX} & s \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s \\ \text{RESTR} & \langle\ \ldots\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

This is the same as (23), except that it includes constraints contributed by the Past-Tense Verb Lexical Rule. In particular (25) ensures that *was* is finite (i.e. [FORM fin]) and that it has past-tense semantics (suppressed here) and a third-person singular subject.[6] Note that the subject in (25) is identical to the complement's subject (as was the case in (23)). Further, the verb's SPR value is constrained to be identical to the first member of the ARG-ST list. This, together with the COMPS value, is the result of the ARP, which (25) must obey.

So now let us consider more closely the VP[pass], whose head is the passive participle *handed*. The lexical entry for *hand* is the following:

(26)

$$\left\langle \text{hand}\ ,\ \begin{bmatrix} \textit{dtv-lxm} & \\ \text{ARG-ST} & \langle\ X_i\ ,\ Y_j\ ,\ Z_k\ \rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{hand} \\ \text{SIT} & s \\ \text{HANDER} & i \\ \text{RECIPIENT} & j \\ \text{HANDED} & k \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

[6]The verb *be* is unique among English verbs in distinguishing different forms (*was* and *were*) in the past tense. See note 34 of Chapter 8.

The lexical sequences satisfying this lexical entry all obey (27):

(27)
$$\left\langle \text{hand}\ , \begin{bmatrix} \textit{dtv-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \boxed{6} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ [\text{AGR}\ \ \boxed{6}]\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \langle\ \text{NP}_i\ ,\ \text{NP}_j\ ,\ \text{NP}_k\ \rangle \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{hand} \\ \text{SIT} & s \\ \text{HANDER} & i \\ \text{RECIPIENT} & j \\ \text{HANDED} & k \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

In addition, they may undergo the Passive Lexical Rule, yielding lexical sequences like the following:

(28)
$$\left\langle \text{handed}\ , \begin{bmatrix} \textit{part-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \boxed{6} \\ \text{FORM} & \text{pass} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ [\text{AGR}\ \ \boxed{6}]\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \text{NP}_j\ ,\ \text{NP}_k \left(,\ \begin{matrix}\text{PP}\\ \begin{bmatrix} \text{FORM} & \text{by} \\ \text{INDEX} & i \end{bmatrix}\end{matrix} \right) \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{hand} \\ \text{SIT} & s \\ \text{HANDER} & i \\ \text{RECIPIENT} & j \\ \text{HANDED} & k \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

And these may undergo the Constant Lexeme Lexical Rule to give sequences like (29): (Note that as words, these are subject to the ARP.)

(29)

$$
\left\langle \text{handed}\ ,\ \begin{bmatrix}
\textit{word} & \\
\text{SYN} & \begin{bmatrix}
\text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \boxed{6} \\ \text{FORM} & \text{pass} \end{bmatrix} \\
\text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \boxed{1}[\text{AGR}\ \boxed{6}]\ \rangle \\ \text{COMPS} & \boxed{B} \end{bmatrix}
\end{bmatrix} \\
\text{ARG-ST} & \left\langle \boxed{1}\text{NP}_j \right\rangle \oplus\ \boxed{B} \left\langle \text{NP}_k \left(,\ \begin{bmatrix} \text{PP} & \\ \text{FORM} & \text{by} \\ \text{INDEX} & i \end{bmatrix} \right) \right\rangle \\
\text{SEM} & \begin{bmatrix}
\text{MODE} & \text{prop} \\
\text{INDEX} & s \\
\text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{hand} \\ \text{SIT} & s \\ \text{HANDER} & i \\ \text{RECIPIENT} & j \\ \text{HANDED} & k \end{bmatrix} \right\rangle
\end{bmatrix}
\end{bmatrix} \right\rangle
$$

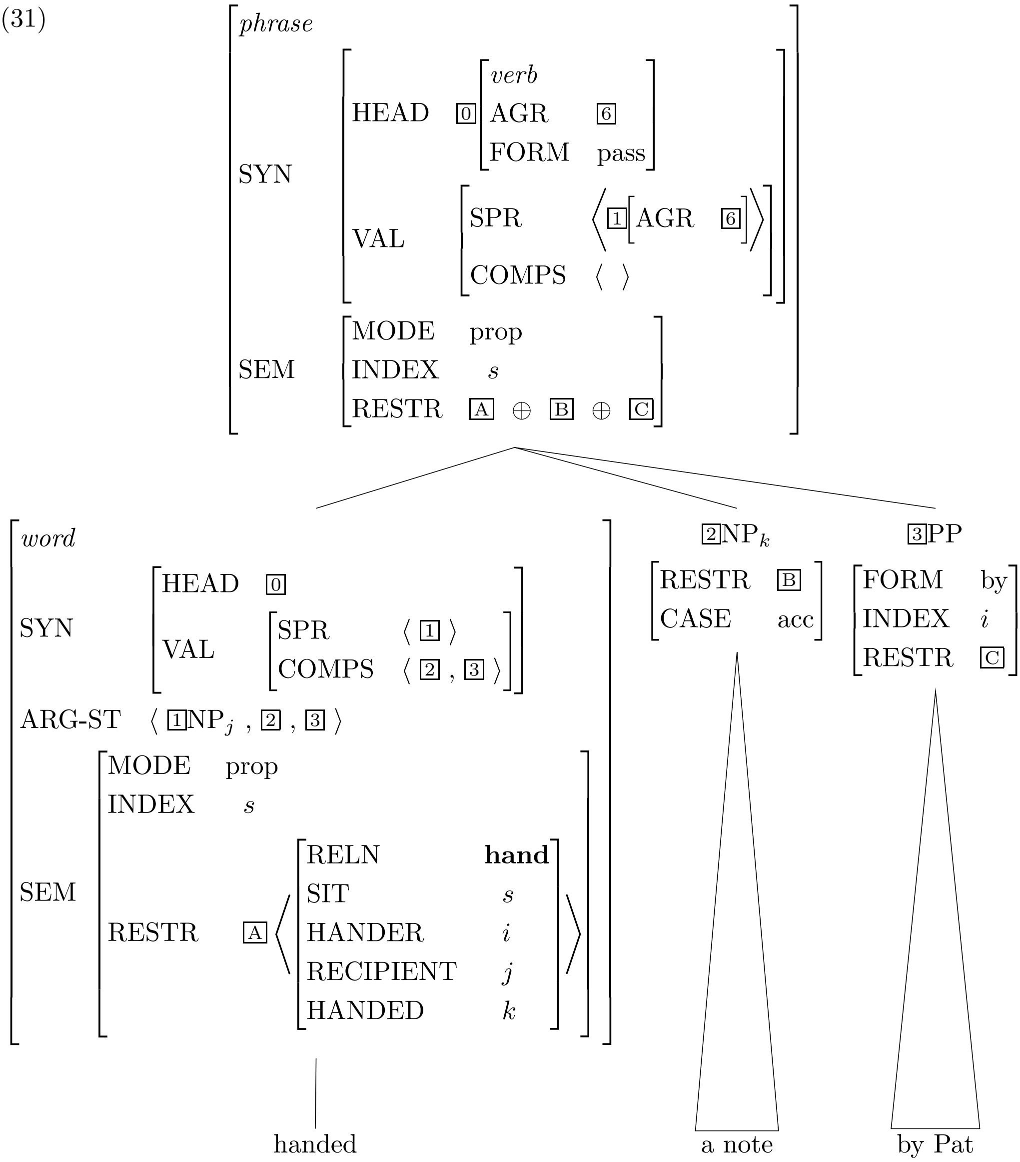
Lexical sequences like (29) form the basis for word structures like (30), where the optionality of the PP is resolved, and the Case Constraint and the Binding Theory come into play:

(30)

$$
\begin{array}{c}
\begin{bmatrix}
\textit{word} & \\
\text{SYN} & \begin{bmatrix}
\text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \boxed{6} \\ \text{FORM} & \text{pass} \end{bmatrix} \\
\text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \boxed{1}[\text{AGR}\ \boxed{6}]\ \rangle \\ \text{COMPS} & \langle\ \boxed{2}\ ,\ \boxed{3}\ \rangle \end{bmatrix}
\end{bmatrix} \\
\text{ARG-ST} & \left\langle \boxed{1}\text{NP}_j\ ,\ \boxed{2}\text{NP}_k[\text{acc}]\ ,\ \begin{bmatrix} \boxed{3}\text{PP} & \\ \text{FORM} & \text{by} \\ \text{INDEX} & i \end{bmatrix} \right\rangle \\
\text{SEM} & \begin{bmatrix}
\text{MODE} & \text{prop} \\
\text{INDEX} & s \\
\text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{hand} \\ \text{SIT} & s \\ \text{HANDER} & i \\ \text{RECIPIENT} & j \\ \text{HANDED} & k \end{bmatrix} \right\rangle
\end{bmatrix}
\end{bmatrix} \\
| \\
\text{handed}
\end{array}
$$

This is consistent with the use of *handed* in (24). (30) fits into the larger tree corresponding to the VP[pass] shown in (31):

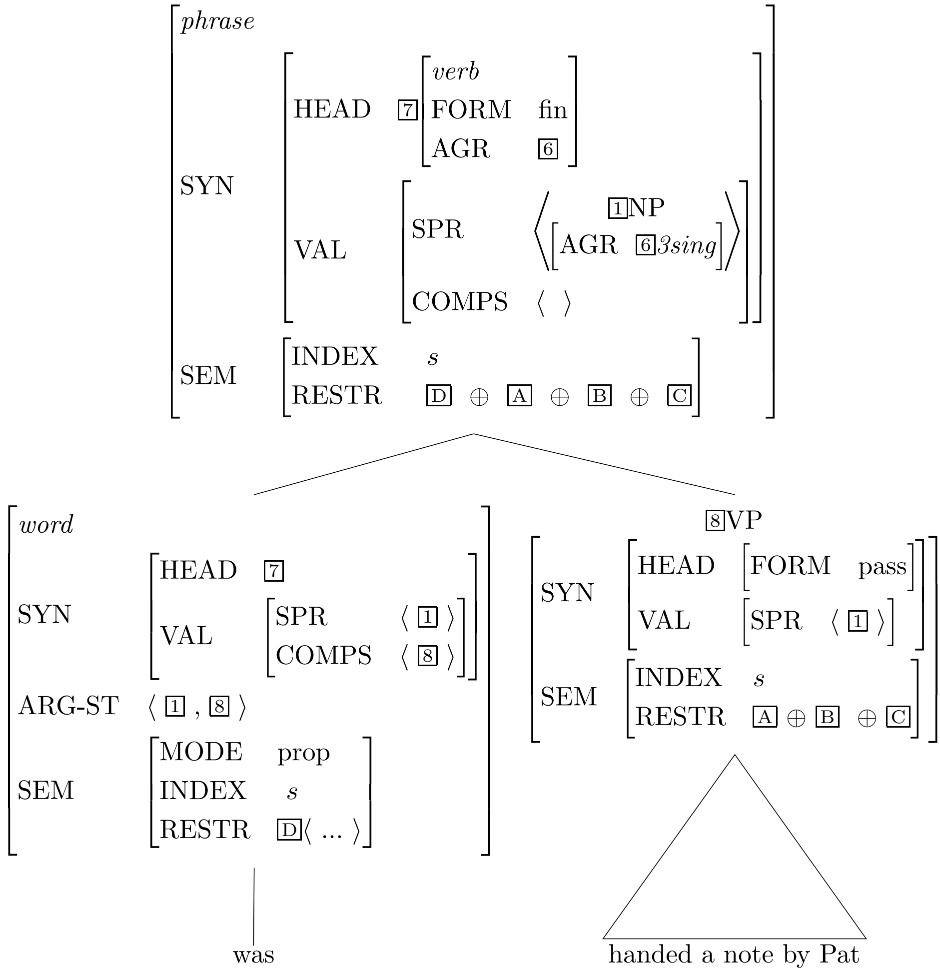
(31)



As usual, the HEAD, SPR, and INDEX values of the mother are the same as those of the head daughter (courtesy of the HFP, the Valence Principle, and the Semantic Inheritance Principle, respectively), and the mother's RESTR value is the sum of the daughters' RESTR values (courtesy of the Semantic Compositionality Principle).

This VP[pass] combines with a word structure licensed by the lexical sequence in (25) to form the VP[fin] in (24), which is shown in more detail in (32):

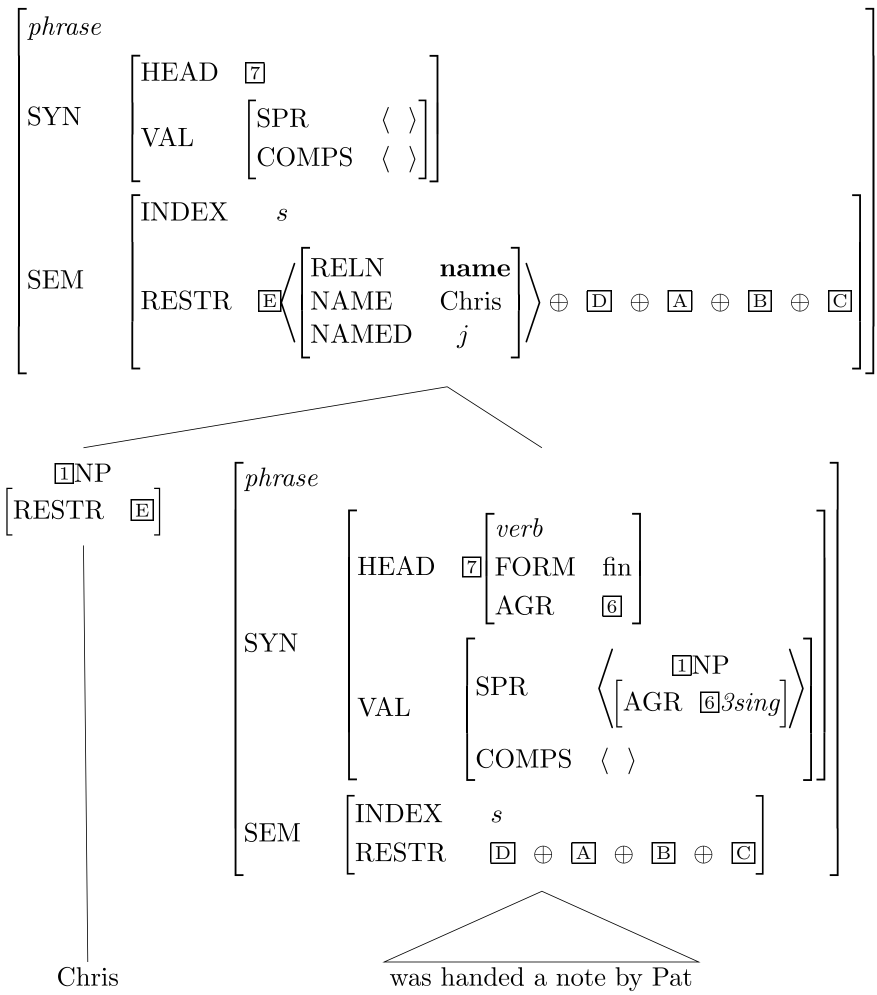
(32)



Again note the effect of the HFP, the Valence Principle, the Semantic Compositionality Principle, and the Semantic Inheritance Principle.

And finally, this VP combines with the subject NP, as shown in (33):

(33)



Since the NP dominating *Chris* is singular, it is consistent with the SPR specification in (33). Because of the identity of subjects established in *be-lxm*, *Chris* (more precisely the NP dominating *Chris*) is the subject of both *was* and *handed*. This assigns the correct semantic interpretation to the sentence: *Chris* plays the recipient role of the handing relation. The other two roles are straightforwardly determined by the indexing shown in (31).

10.6 Summary

Our treatment of the active/passive alternation in English is based on a relationship between verb forms. We formalize this with a derivational lexical rule that modifies the lexeme type, the morphology, the argument structure, and some details of the HEAD values. Passive participles usually follow a form of *be*; this chapter introduced a lexical entry for this use of *be*. Passive participles and the form of *be* that precedes them share the same subject. Our lexical entry for *be* encodes this fact, anticipating a central topic of Chapter 12.

10.7 Changes to the Grammar

In this chapter, we added the following lexical rule to the grammar:

Passive Lexical Rule

$$\begin{bmatrix} \textit{d-rule} & \\ \text{INPUT} & \left\langle \boxed{1}\,, \begin{bmatrix} \textit{tv-lxm} & \\ \text{ARG-ST} & \langle\, [\text{INDEX } i] \,\rangle \oplus \boxed{\text{A}} \end{bmatrix} \right\rangle \\ \text{OUPUT} & \left\langle \text{F}_{PSP}(\boxed{1})\,, \begin{bmatrix} \textit{part-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & [\text{FORM} \;\; \text{pass}\,] \end{bmatrix} \\ \text{ARG-ST} & \boxed{\text{A}} \oplus \left\langle \left(\begin{matrix} \text{PP} \\ \begin{bmatrix} \text{FORM} & \text{by} \\ \text{INDEX} & i \end{bmatrix} \end{matrix} \right) \right\rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

We also added a lexeme *be*, which is distinguished from other verb lexemes we've seen so far in that it identifies the first member of its ARG-ST list with the SPR of the second member:

$$\left\langle \text{be}\,, \begin{bmatrix} \textit{be-lxm} & \\ \text{ARG-ST} & \left\langle \boxed{1}\,, \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{pass} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\, \boxed{1} \,\rangle \\ \text{COMPS} & \langle\;\rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \langle\;\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

The constraints in (33) will be revised somewhat in Chapters 11 and 13, but this key property will remain constant.

10.8 Further Reading

The English passive has been analyzed and reanalyzed throughout the history of generative grammar. Among the most influential works on the subject are: Chomsky 1957, 1965, and 1970; Perlmutter and Postal 1977; Wasow 1977; Bresnan 1982c; Burzio 1986; and Postal 1986.

10.9 Problems

Problem 1: Passive and Binding Theory

The analysis of passive just sketched makes some predictions about binding possibilities in passive sentences. Consider the following data:[7]

(i) She$_i$ was introduced to herself$_i$ (by the doctor).
(ii)*She$_i$ was introduced to her$_i$ (by the doctor).
(iii) The barber$_i$ was shaved (only) by himself$_i$.
(iv)*The barber$_i$ was shaved (only) by him$_i$.
(v) The students$_i$ were introduced to each other$_i$ (by Leslie).
(vi)*The students$_i$ were introduced to them$_i$ (by Leslie).
(vii) Kim was introduced to Larry$_i$ by himself$_i$.
(viii)*Kim was introduced to himself$_i$ by Larry$_i$.

Assuming that *to* and *by* in these examples are uniformly treated as argument-marking prepositions, does the treatment of passives sketched in the text correctly predict the judgements in (i)–(viii)? If so, explain why; if not, discuss the inadequacy of the analysis in precise terms.

An ideal answer should examine each one of the eight sentences and determine if it follows the binding principles. That is, the analysis of passive presented in this chapter associates a particular ARG-ST list with the passive verb form in each example and these lists interact with the binding principles of Chapter 7 to make predictions. Check to see if the predictions made by our Binding Theory match the grammaticality judgements given.

Problem 2: Pseudopassives

Consider the following passive sentences:

(i) Dominique was laughed at by the younger kids.
(ii) This bed was slept in by the ambassador to Dalmatia.
(iii) This problem is talked about in every home.

A. Explain why our current passive rule does not allow sentences like (i)–(iii) to be generated.

[7] It may require a little imagination to construct contexts where such examples have a plausible meaning, e.g. a doctor dealing with an amnesia victim. Being able to construct such contexts is an essential part of being able to understand what conclusions to draw from the fact that some sentence you are interested in doesn't sound completely acceptable.

We know of cases where grammatical deviance has not been separated with sufficient care from semantic implausibility. For example, examples like ?*I smell funny to myself* have on occasion been cited as ungrammatical. However, a bit of reflection will reveal, we think, that what is strange about such examples is the message they convey, not their grammar. If one needed to convey that one's own olfactory self-impression was strange (in whatever odd context such a need might arise), then *I smell funny to myself* is probably the most straightforward way the grammar of English has for allowing such a meaning to be expressed.

B. Give the ARG-ST and RESTR values for one of the passive participles in (i)–(iii), along with the ARG-ST and RESTR values of the corresponding active form.

C. Propose an additional lexical rule that will produce appropriate lexical sequences for the passive participles in these sentences.

[*Hints: Your new rule should be similar to our existing Passive Lexical Rule. Assume that the prepositions involved in examples of this sort are all argument-marking prepositions – that is, they all share INDEX and MODE values with their object NPs. Your rule will need to use these INDEX values (and the FORM values of the prepositions) in producing the passive lexemes needed to license examples like (i)–(iii).*]

D. Explain how your lexical rule relates the ARG-ST values you gave in (B) to each other.

E. Assuming the lexical entry in (iv), does the rule you formulated in (C) predict that both (iii) and (v) are grammatical?

(iv) $\left\langle \text{talk}\ ,\ \begin{bmatrix} \textit{new-v-lxm} \\ \text{ARG-ST} \quad \langle\ \text{NP (, PP[to]) (, PP[about])}\ \rangle \end{bmatrix} \right\rangle$

(v) This person was talked to by every teacher.

Explain your answer.

Problem 3: The Dative Alternation

In Chapter 8, we mentioned the possibility of formulating a lexical rule describing the 'dative alternation' – that is, a class of verbs that appear in both of the valence patterns exemplified in (i) and (ii):

(i) Dale $\left\{ \begin{array}{l} \text{gave} \\ \text{handed} \\ \text{sold} \\ \text{loaned} \\ \text{mailed} \end{array} \right\}$ Merle a book.

(ii) Dale $\left\{ \begin{array}{l} \text{gave} \\ \text{handed} \\ \text{sold} \\ \text{loaned} \\ \text{mailed} \end{array} \right\}$ a book to Merle.

A. Is this alternation productive? Justify your answer with at least two examples.

[*Hint: See the discussion of productive lexical rules at the end of Section 8.1 of Chapter 8.*]

B. Formulate a lexical rule for the dative alternation.

[*Hint: Consider which kind of l-rule (i-rule or d-rule) this should be, based on the kind of constraints you need to write. You can choose either of the valences* *illustrated in (i) and (ii) as the input and the other as the output. It should not be easier one way than the other.*]

C. Show how your rule interacts with the Passive Lexical Rule to make possible the generation of both (iii) and (iv). Your answer should include ARG-ST values showing the effect of applying the rules. [*Hint: First consider which order the rules apply in, based on the types of the INPUT and OUTPUT values of each rule.*]

   (iii) Merle was handed a book by Dale.

   (iv) A book was handed to Merle by Dale.

D. Explain why your rule correctly fails to license (v) (or, more precisely, fails to license (v) with the sensible meaning that the book was the thing handed to Merle).

   (v) ?*A book was handed Merle by Dale.

11

Nominal Types: Dummies and Idioms

11.1 Introduction

In the last chapter, we presented a lexical entry for the verb *be* as it occurs in passive sentences. We begin this chapter with a consideration of how to generalize the formulation of this lexical entry to cover other uses of *be* as well. This will lead us to the use of forms of *be* in combination with the subject *there* as a way of presenting an entity or asserting its existence, as in (1):

(1) a. There are storm clouds gathering.
    b. There is a monster in Loch Ness.

This, in turn, will take us to an examination of other NPs that seem to have very restricted distributions, and whose semantic contributions cannot readily be isolated from the meanings of the larger constructions in which they occur. Examples are the use of *it* in sentences like (2a) and *tabs* in (2b):

(2) a. It is obvious that Pat is lying.
    b. The FBI is keeping close tabs on Pat.

11.2 *Be* Revisited

The lexical entry for *be* presented in the last chapter demanded a VP[FORM pass] complement, but of course forms of *be* occur with a variety of other types of complements:

(3) a. Pat is on the roof.
    b. Pat is the captain of the team.
    c. Pat is fond of Chris.
    d. Pat is singing the blues.

Such examples show that the possible complements of *be* include, besides VP[FORM pass], at least PP, NP, AP, and VP[FORM prp]. At first glance, one might think that this could be handled simply by removing the FORM feature (and hence, implicitly, the part of speech information) from the second element of the ARG-ST list in the lexical entry for passive *be* – that is, by allowing any type of phrase (of the appropriate valence) as a complement. However, the distribution of *be* is not quite this free.

(4) a. *Pat is likes Chris.
    b. *Pat is hate Chris.
    c. *Pat is mere.

These examples show that only some verb forms can head a VP complement of *be* and that not all adjectives can head AP complements of *be*. The traditional name for the kind of phrase that can appear after *be* is 'predicative'. We will introduce a binary feature PRED to encode the distinction between predicative and non-predicative phrases. So *fond* is [PRED +], while *mere* is [PRED −], though both have HEAD values of type *adj*. Likewise, passive and present participles are [PRED +], and all other verb forms are [PRED −]. We can state these constraints on verb forms most generally by making [PRED −] a constraint on *verb-lxm*, and having the lexical rules which create passive and present participles change this specification to [PRED +]. The inflectional rules for verbs won't affect the PRED specification.

Using the feature PRED, we can reformulate the lexical entry for *be* to handle not only passive VP complements, but also complements like those in (3). The new formulation[1] is the following:

(5)
$$\left\langle \text{be}\ , \begin{bmatrix} \textit{be-lxm} & \\ \text{ARG-ST} & \left\langle \boxed{1}\ , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & [\text{PRED}\ \ +] \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & [\text{INDEX}\ \ s] \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

As before, the semantic index of the verb *be* (s) is just the index of its predicative complement – the verb contributes nothing to the semantics of the sentences; it is just a syntactic placeholder. Indeed, in many languages (including some dialects of English – see Chapter 15) the meanings like those expressed by the sentences in (3) would normally be expressed without any verb at all, as in the following examples:

(6) a. Ona xorošij vrač.
she good doctor
'She is a good doctor.'
(Russian)

b. A magyar zászló piros-fehér-zőld.
the Hungarian flag red-white-green
'The Hungarian flag is red, white, and green.'
(Hungarian)

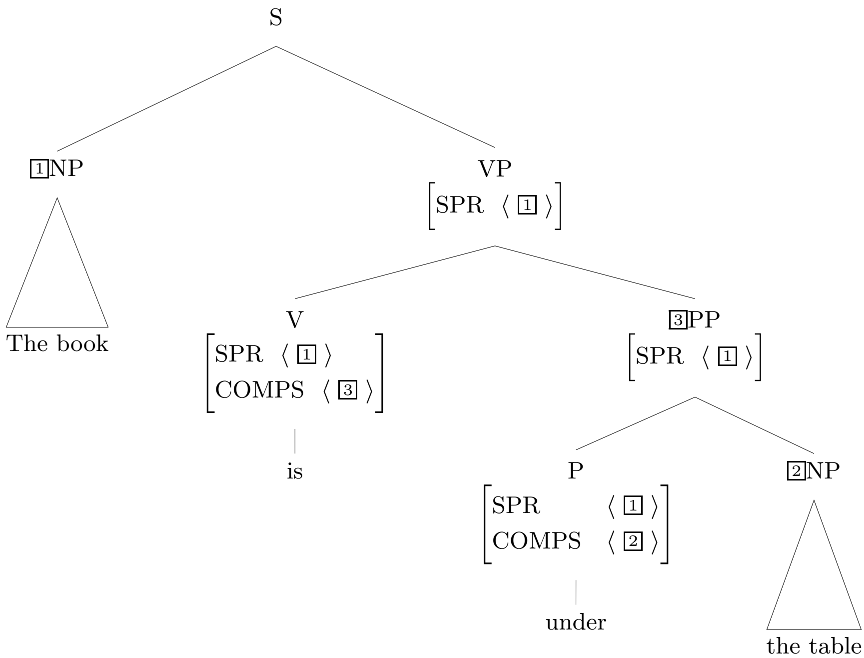
As discussed in Chapter 10, Section 10.4, the first argument of *be* is identified with the SPR requirement of its second argument. This means that all complements of *be* must

[1]We will incorporate this entry (in slightly revised form) into our lexical type hierarchy in Chapter 13, Section 13.2.2.

have non-empty SPR values.[2] For example, predicative prepositions like *under* in (7) must have two arguments, one on the SPR list and one on the COMPS list.

(7) The book is under the table.

(8)



The syntactic arguments correspond to the two semantic arguments that the preposition takes: the **under** relation holds between the book and the table.

11.3 The Existential *There*

Consider another sort of sentence that involves *be*:

(9) a. There is a unicorn in the garden.
b. There were many people fond of Pat.
c. There are people looking through the window.
d. There was a felon elected to the city council.

---

[2]There is a bit more to be said in the case of predicative NPs. In order to account for examples like (i), such NPs must be [SPR ⟨ NP ⟩].

(i) Pat is a scholar.

Since NPs normally have empty SPR values, our account is incomplete.

One possible solution is a non-branching phrase structure rule, whose mother is a NOM and whose daughter is an NP. We will not develop this solution further here. Observe, however, that this syntactic distinction between predicative and nonpredicative NPs reflects a semantic difference between two uses of certain NPs: one involving properties; the other individuals. Thus, the NP *a scholar* in (i) is used to predicate a property of Pat (scholarliness) and hence its semantic mode is actually 'prop', whereas the same string of words in (ii) is used simply to make reference to an individual, i.e. its semantic mode is 'ref'.

(ii) A scholar arrived.

These involve a nonreferential subject, *there* (often called the 'dummy' *there*), an NP following *be*, and a [PRED +] phrase following the NP. We can see that there are in fact two complements and not just one complex one (that is, an NP with some kind of modifying phrase attached to it) on the basis of sentences like (10).

(10) a. There is a seat available.
b.*A seat available was in the last row.
c.*Pat took a seat available.
d.*I looked for a seat available.

If *a seat available* in (10a) were a single NP, we would expect it to be able to appear in other typical NP positions, such as those in (10b-d). So *a seat* and *available* must be two separate arguments of *be*. But if this use of *be* takes a subject and two more arguments, then it cannot be subsumed under (5), whose ARG-ST list contains only two elements. Hence, we will need a separate lexical entry for this lexeme, which we will call the 'existential *be*'.

Stating the restrictions on the existential *be*'s complements is not difficult,[3] but restricting the subject to the word *there* is not entirely trivial. This is the first case we have seen in which a verb requires that a particular word be its subject. We have, however, previously encountered verbs that select PP complements that are headed by a specific word. This was true, for example, in the passive construction discussed in Chapter 10: the passive form of a verb always allows a PP headed by *by* to express the argument of the passive that corresponds semantically to the subject of the active. Similar selections are involved with other verb-preposition pairs, such as *rely* and *on*. Indeed, the argument-marking prepositions discussed in Chapter 7 are often selected by verbs, sometimes quite idiosyncratically.

Recall that to deal with selection of prepositions by verbs, we introduced specific values of the feature FORM (previously used primarily for distinguishing verbal inflections) for prepositions, adding new FORM values such as 'by', 'to', etc. And in Chapter 8, the value 'nform' was used as the default value for nouns of all kinds. We can now introduce specific values of FORM for exceptional nominals that need to be grammatically regulated. For example, we can put the feature specification [FORM there] in the lexical entry for the existential *there*, and require that the subject of the existential *be* must be [FORM there], as shown in (11):[4]

[3]This is an oversimplification (as is almost any claim that some aspect of grammar is easy). Examples like (i) and (ii) are markedly worse than sentences like (9):

(i) ?*There is each unicorn in the garden.
(ii) ?There was the felon elected to the city council.

It is often claimed that the postverbal NP in existential sentences must be indefinite, but this is too strong: examples like (ii) are acceptable if interpreted as part of a listing of exemplars of something, and sentences like *There is the cutest puppy outside* are commonplace (in certain styles, at least). We will not pursue the problem of characterizing the so-called definiteness restriction on the NPs in existential sentences, on the assumption that the restriction is actually a semantic one.

[4]Our use of FORM values may seem somewhat promiscuous. In actual practice, however, we believe that the number of words entering into such morphologically-sensitive co-occurrence relations in any language is quite manageable.

(11)
$$\left\langle \text{be}\ ,\ \begin{bmatrix} \textit{exist-be-lxm} & \\ \text{ARG-ST} & \left\langle \begin{matrix}\text{NP}\\ \begin{bmatrix}\text{FORM} & \text{there}\end{bmatrix}\end{matrix},\ \boxed{2}\ ,\ \begin{bmatrix} \text{PRED} & + \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \boxed{2}\ \rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix} \\ \text{SEM} & [\text{INDEX}\ \ s\ ] \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s \\ \text{RESTR} & \langle\ \rangle\end{bmatrix} \end{bmatrix} \right\rangle$$

Notice that the existential *be* contributes nothing to the meaning of the sentence, except the identification of its index with that of its predicative complement. Moreover, since the NP argument is identified with the SPR of the predicative complement, the semantics of these two will be combined within the VP in the same way as they would be in a simple subject-predicate sentence: The index of the NP ends up associated with the same semantic role in the verb's predication, and RESTR lists are merged by the Semantic Compositionality Principle. Since existential *be* itself contributes no predications (nor does *there*, see below), the RESTR of an existential sentence ends up being the same as the RESTR of a corresponding non-existential sentence. Thus, the sentences in (9) are analyzed as paraphrases of those in (12).[5]

(12) a. A unicorn is in the garden.
b. Many people were fond of Pat.
c. People are looking through the window.
d. A felon was elected to the city council.

We turn now to the lexical entry for the existential *there*. Its key property is being the only word in the English language that is specified as [FORM there]. Hence, the SPR value of (11) picks out this word as the only compatible subject. Non-dummy NPs (proper nouns, pronouns, and phrases headed by common nouns alike) continue to be specified as [FORM nform]. (Recall that this is the result of the defeasible constraint on the type *noun* that was introduced in Chapter 8.) A few other special nouns (including those discussed later in this chapter) will also have distinguished values for FORM that override the default. The existential *there* is exceptional in that, although a pronoun, it has no referential function, and under our analysis (as noted above) it does not contribute to the meaning of the sentences in which it occurs (but see footnote 5). The lexical entry for existential *there* is thus the following:

[5]This account of the semantics of the existential *there* construction is only a first approximation. For one thing, the use of *there* seems to involve an explicit assertion of existence not associated with sentences like (12). In addition, the [PRED +] phrase in the *there* construction must denote a potentially transient property of the referent of the NP, whereas this is not required in the analogous examples without *there*. This is illustrated in (i)–(iv):

(i) A vase is blue.
(ii) *There is a vase blue.
(iii) A unicorn was the winner of the Kentucky Derby.
(iv) *There was a unicorn the winner of the Kentucky Derby.

We will not pursue these interesting (and subtle) semantic issues here.

(13)
$$\left\langle \text{there}\ , \begin{bmatrix} \textit{pron-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{there} \\ \text{AGR} & \begin{bmatrix}\text{PER} & \text{3rd}\end{bmatrix}\end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{none} \\ \text{INDEX} & \text{none} \\ \text{RESTR} & \langle\ \rangle \end{bmatrix}\end{bmatrix}\right\rangle$$

The lexeme in (13) inherits from the type *pron-lxm* the constraints [HEAD *noun*] and [ARG-ST ⟨ ⟩]. Observe that the AGR specification in (13) is unspecified for number; this is because *there* can be plural, as in (9b,c). Note in addition that the empty list specification for the feature RESTR guarantees that *there* will not contribute to the RESTR list of phrases that contain it. And finally, the 'none' values that we have introduced for the features MODE and INDEX reflect the fact that *there* has no referential potential and no referential index.

This last fact is particularly significant, as it allows us to account for the restricted distribution of existential *there*. Each of the verbs we have considered thus far (except for *be*) has a lexical entry which identifies the INDEX value of each of its arguments with the value of some semantic role (e.g. LOVER, GIVEN) in its predication. However, the semantic roles require values of a certain type (namely, *index*). The value of the feature INDEX in (13), 'none', is incompatible with this type. Intuitively, since *there* doesn't have an index, any attempt to combine *there* with a role-assigning verb will produce a conflict. Thus from the semantic vacuity of existential *there*, it follows immediately that examples like the following are ungrammatical:

(14) a. *There loved Sandy.
b. *Merle gave there a book.
c. *We talked to them about there.

In this section, we have seen our first example of a semantically empty noun: the dummy *there* of existential constructions. In the following sections, we will explore two more kinds of dummy NPs.

11.4 Extraposition

This section considers a second semantically empty noun, the dummy *it* of extraposition. Extraposition[6] is illustrated in the following pairs of sentences:

(15) a. That the Giants had lost (really) mattered.
b. It (really) mattered that the Giants had lost.

(16) a. That dogs bark annoys people.
b. It annoys people that dogs bark.

(17) a. That Chris knew the answer (never) occurred to Pat.
b. It (never) occurred to Pat that Chris knew the answer.

[6]In using this terminology, we follow the renowned Danish grammarian Otto Jespersen (1860–1943).

(18) a. That the Cardinal won the game gave Sandy a thrill.
b. It gave Sandy a thrill that the Cardinal won the game.

This seems to be a systematic alternation that we would like to account for. Moreover, it is productive: an English speaker unfamiliar with the verb *discomfit* who heard (19a) would know that (19b) is also well formed:

(19) a. That the media discuss celebrities' sex lives discomfits many Americans.
b. It discomfits many Americans that the media discuss celebrities' sex lives.

And speakers who use verbs like *suck* or *bite* in the sense of 'be bad' should find both members of the following pairs to be well formed:

(20) a. That the Giants lost the series (really) sucks.
b. It (really) sucks that the Giants lost the series.

(21) a. That the Giants lost the series (really) bites.
b. It (really) bites that the Giants lost the series.

Thus the alternation illustrated in (15)–(18) appears to have some psycholinguistic reality.

The b-sentences in (15)–(21) all have a nonreferential pronoun *it* as their subject and a *that*-clause at the end. This nonreferential – or 'dummy' – pronoun is in fact quite similar to the expletive *there* discussed in the previous section. Like existential *there*, the dummy *it* is very restricted in its distribution. This may not be evident, but in examples like (22)–(23), which do not fit the pattern of (16)–(21), the uses of *it* are referential:

(22) a.*That Pat is innocent proves.
b. It proves that Pat is innocent.

(23) a.*That Sandy had lied suggested.
b. It suggested that Sandy had lied.

That is, the *it* that occurs in each of these examples is a referential pronoun, analyzed in terms of a lexical entry distinct from the dummy *it*.

Following the treatment of the existential *there*, then, we are led to posit lexical sequences for the verbs in the b-sentences of (17)–(21) that specify that their subjects must be the nonreferential *it*. We can do this as we did with *there* by positing a FORM value 'it', which uniquely identifies the dummy *it*. The lexical entry for the dummy *it* is therefore the following:

(24)
$$\left\langle \text{it}, \begin{bmatrix} \textit{pron-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{it} \\ \text{AGR} & \textit{3sing} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{none} \\ \text{INDEX} & \text{none} \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Note that the dummies *it* and *there* have slightly different AGR values: unlike *there*, *it* is always singular, as shown by the following contrast:

(25) It annoys/*annoy people that dogs bark.

Consequently, where the entry for *there* has the AGR value [PER 3rd], the entry for *it* has the more restrictive AGR value *3sing*.

Like the dummy existential *there*, and for exactly the same reasons, dummy *it* can never appear in a role-assigned position:

(26) a. *It loved Sandy.
b. *I gave it to Pat.

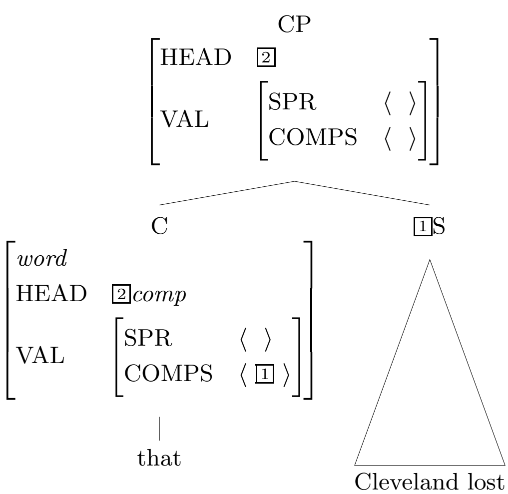
Such examples are fully grammatical, of course, if we interpret *it* as the personal pronoun *it* (i.e. as a pronoun referring to something in the context), in which case we are dealing with the homophonous referential pronoun, rather than the dummy *it*.

To capture the regularity of the alternation illustrated in (15)–(21), we will want to posit a lexical rule whose output is the version of the verbs taking the dummy subject *it*. But before we can do this, we need to consider how to analyze the *that*-clauses that occur in the examples in question.

11.4.1 Complementizers and *That*-Clauses

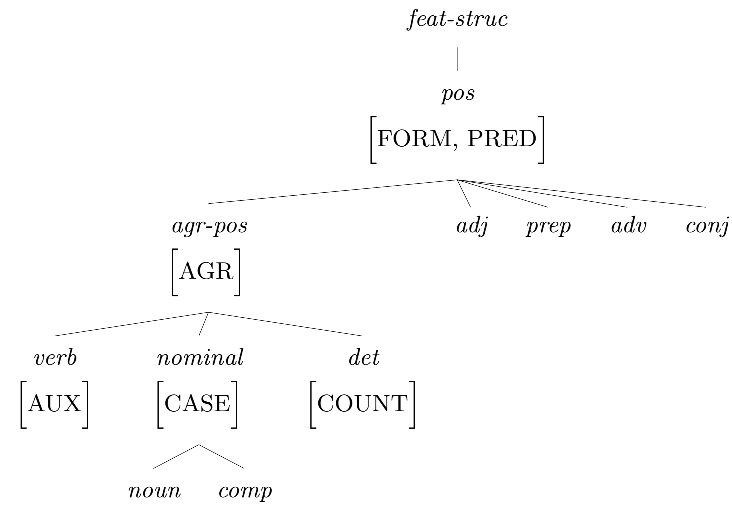
The part after *that* is just a finite S (i.e. a phrase headed by a finite verb, with empty COMPS and SPR specifications – as noted in Chapter 4, we sometimes call such a phrase 'saturated'). It is less obvious how to deal with *that*, which might be thought of as simply 'marking' the sentence that follows it. We treat *that* as a head, taking a finite S as its only argument (note that in this respect, *that* is similar to the argument-marking prepositions such as *to* and *of* discussed in Chapter 7). In order to handle words like *that*, however, we will have to introduce a new part of speech type: *comp* (for 'complementizer'). *That*-clauses, then, are complementizer phrases (CPs, for short) whose structure is as shown in (27):

(27)



We'll see that the type *comp* is most like the type *noun* in terms of which features are appropriate for it. Therefore, we will fit *comp* into our part-of-speech hierarchy in terms of a supertype *nominal*, as shown in (28):

(28)



The type *comp* will be subject to the constraint in (29), where 'cform' is a FORM analogous to 'nform', 'aform', etc. (see Section 8.5.2 of Chapter 8):

(29) $\textit{comp} : \begin{bmatrix}\text{FORM} & \text{cform}\end{bmatrix}$

In Chapter 8, we proposed a constraint on the type *verb-lxm* requiring that the first member of the ARG-ST list be an NP. This constraint needs to be revised in light of the CP subjects we see in the a-examples of (15)–(21). We can use the type *nominal* to restate the constraint: verbs have argument structures that start with a [HEAD *nominal*], saturated phrase. The lexical entries for some verbs will constrain this further, but others will leave it underspecified. Since (finite forms of) such verbs will assign nominative case to their subjects regardless of whether they are NPs or CPs, the feature CASE must be appropriate to the type *comp*. The hierarchy in (28) ensures that it is.[7]

Just as many verbs can take either NPs or *that*-clauses as subjects, many transitive verbs also allow *that*-clauses as their first complement:

(30) a. Cohen proved the independence of the continuum hypothesis.
b. Cohen proved that the continuum hypothesis was independent.
c. We forgot our invitations.
d. We forgot that we needed invitations.
e. Nobody saw Pat.
f. Nobody saw that Pat had arrived.

Such cases can be accommodated without changing the lexical entries of the verbs in question, if we change the constraint on the type *tv-lxm* from (31) to (32):

[7]For uniformity, we will also generalize the Case Constraint (introduced in Chapter 8) so that it requires that CPs, as well as NPs, be [CASE acc] when they are noninitial in an ARG-ST.

(31) $\left[\text{ARG-ST} \quad \langle \text{ X , NP , ... } \rangle\right]$

(32)
$$\left[\text{ARG-ST} \quad \left\langle \text{X} , \begin{bmatrix} \text{HEAD} & \textit{nominal} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix}, ... \right\rangle \right]$$

Of course, not all transitive verbs take *that*-clause complements, but those that don't (such as *devour, pinch, elude*, etc.) can have additional constraints in their lexical entries. Similarly, there are verbs (such as *hope*) which can take CP but not NP complements, and these can be treated in an analogous fashion. Alternatively, it might plausibly be argued that these selectional restrictions are semantic in nature, so that this constraint need not be specified in their ARG-ST values.[8]

The next issue to address before formulating our lexical rule is the semantic role played by the *that*-clauses in both the a- and b-sentences of (15)–(21). So far, the values we've posited for the feature RESTR have been lists of simple predications, that is, predications where the semantic role features (LOVER, INST, etc.) take indices as their arguments. These indices in general correspond to individuals that are referred to by NPs within the sentence. One important exception to this has to do with modification. In Chapter 5, we allowed situational indices to be the value of the feature ARG(UMENT) that appeared in certain predications introduced by adverbial modifiers, as in (33):

(33)
$$\begin{bmatrix} \text{MODE} & \text{none} \\ \text{INDEX} & s_1 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{today} \\ \text{SIT} & s_1 \\ \text{ARG} & s_2 \end{bmatrix} \right\rangle \end{bmatrix}$$

This in fact is the general technique we will use for semantic embedding – for making one semantic complex the argument of another. That is, we will not in general embed one feature structure within another inside the value of SEM, as is done in (34):

(34) **Not** how we represent semantic embedding:

$$\begin{bmatrix} \text{MODE} & \text{none} \\ \text{INDEX} & s_1 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{today} \\ \text{SIT} & s_1 \\ \text{ARG} & \begin{bmatrix} \text{RELN} & ... \\ \text{SIT} & s_2 \\ ... & \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

[8]A further complication in these complementation patterns is that most verbs which can take CP complements can also take S complements. This matter is taken up in Problem 5 below.

Instead, we will use sameness of situational indices to get the same semantic effect. We will use various (hopefully intuitive) feature names to designate the roles whose value is an embedded proposition. In this way, we can express meanings that involve arbitrarily deep semantic embeddings, but we can keep the RESTR lists inside our semantic structures 'flat'.[9]

On this view, we will be able to deal with the semantics of subordinate clauses in terms of index identity, using the kind of semantic analysis we have already developed. For example, we can make the reasonable assumption that the semantics of *that Fido barks* in (35a) is the same as that of the stand-alone sentence (35b), namely, (35c):

(35) a. That Fido barks annoys me.

b. Fido barks.

c.
$$\begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix}\text{RELN} & \textbf{name}\\ \text{NAME} & \text{Fido} \\ \text{NAMED} & i\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{bark}\\ \text{SIT} & s\\ \text{BARKER} & i\end{bmatrix}\right\rangle\end{bmatrix}$$

How do we ensure that this will be the semantics for the CP *that Fido barks*?

The complementizer *that* belongs to a new type of lexeme associated with the constraints in (36):[10]

(36)
$$\textit{comp-lxm} : \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{comp} & \\ \text{AGR} & \textit{3sing} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \begin{matrix} \text{S} \\ \begin{bmatrix} \text{INDEX} & s \end{bmatrix} \end{matrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \end{bmatrix}$$

(36) says that all instances of this type of lexeme share the semantic index of their (only) argument, and contribute no predications (i.e. have an empty RESTR list). Further, it requires that all complementizers be specified as 3rd singular and that they have empty SPR lists. These are the common properties that *that* shares with other complementizers, e.g. *whether*, *if* and *for*, whose analysis would take us too far from this chapter's concerns.

With these type constraints in place, the lexical entry for *that* need say nothing more than what is shown in (37):

[9] We are simplifying here in not providing any apparatus for distinguishing embedded propositions from embedded questions, exclamations, etc., although the machinery developed here can be extended to include such distinctions.

[10] In the grammar we are developing for a fragment of English, the type *comp-lxm* is a subtype of *const-lxm*. Some varieties of Dutch and certain other Germanic languages show what appear to be inflected forms of complementizers.

(37)

$$\left\langle \text{that}\ ,\ \begin{bmatrix} \textit{comp-lxm} & \\ \text{ARG-ST} & \langle \begin{bmatrix}\text{FORM fin}\end{bmatrix} \rangle \\ \text{SEM} & \begin{bmatrix}\text{MODE prop}\end{bmatrix} \end{bmatrix} \right\rangle$$

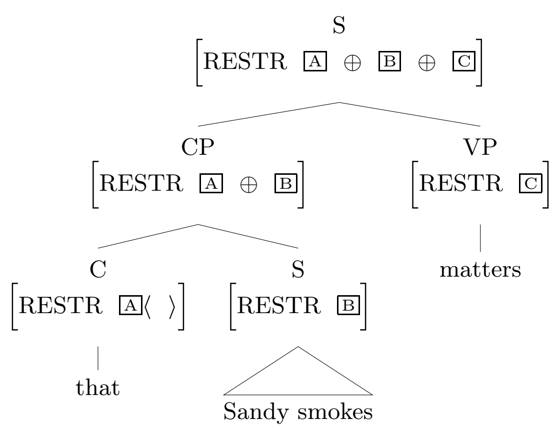
The constraints passed on through type-based constraint inheritance thus interact with those that are lexically specified to ensure that the complementizer *that* has the INDEX value of its only argument, which in turn must be a saturated finite clause. With these constraints in place, the lexical entry for *that* in (37) gives us lexical sequences like (38):

(38)

$$\left\langle \text{that}\ ,\ \begin{bmatrix} \textit{comp-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{comp} & \\ \text{FORM} & \text{cform} \\ \text{AGR} & \textit{3sing} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \overset{\text{S}}{\begin{bmatrix} \text{FORM} & \text{fin} \\ \text{INDEX} & s \end{bmatrix}} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Given (38) and its interaction with the semantics principles of Chapter 5, it follows that the semantics of *that*-clauses is identical to the semantics of the clause that *that* takes as its complement. A clause like *That Sandy smokes matters* will then have the structure shown in (39):

(39)



And the RESTR value of this sentence ($\boxed{\text{A}} \oplus \boxed{\text{B}} \oplus \boxed{\text{C}} = \boxed{\text{B}} \oplus \boxed{\text{C}}$) is shown in (40):

(40)
$$\begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s_1 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{Sandy} \\ \text{NAMED} & i \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{smoke} \\ \text{SIT} & s_2 \\ \text{SMOKER} & i \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{matter} \\ \text{SIT} & s_1 \\ \text{MATTERING} & s_2 \end{bmatrix}, \ldots \right\rangle \end{bmatrix}$$

Importantly, the index of the smoking situation (s_2) is identified with the value of the MATTERING role in the *matter* predication shown in (40). This is achieved through a cascade of identities: *matter* identifies the index of its subject with the MATTERING role, as shown in (41):

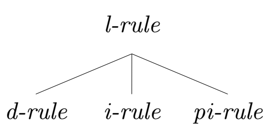
(41)
$$\left\langle \text{matter}, \begin{bmatrix} \textit{siv-lxm} & \\ \text{ARG-ST} & \left\langle \begin{bmatrix} \text{SEM } [\text{INDEX } \boxed{1}] \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{matter} \\ \text{SIT} & s \\ \text{MATTERING} & \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

In (39), this subject turns out to be a CP. The INDEX value of the CP is identified with the INDEX of its head (*that*) by the Semantic Inheritance Principle. The INDEX of *that* is identified with the INDEX of its complement *Sandy smokes*, as required by the constraints *that* inherits from *comp-lxm*. The INDEX of the S *Sandy smokes* is identified with the INDEX of the head of the S (the V *smokes*), again by the Semantic Inheritance Principle. Finally, the lexical entry for *smoke* identifies the INDEX with the SIT value of the *smoke* relation.

11.4.2 The Extraposition Lexical Rule

We are now almost ready to state our extraposition rule. We want the rule to take as input a word whose first argument is a CP and produce as output a word with the CP at the end of its ARG-ST list and an NP[FORM it] at the beginning.

In previous chapters, we have seen derivational lexical rules (*d-rule*s) which map lexemes to lexemes and inflectional lexical rules (*i-rule*s) which map lexemes to words. Extraposition is the first example of a new type: post-inflectional lexical rules (*pi-rule*s), which map words to words. The type *pi-rule* is a sister of *i-rule* and *d-rule*, as shown in (42):

(42)


It is subject to the constraint shown in (43):

(43)

$$
\textit{pi-rule}: \begin{bmatrix}
\text{INPUT} & \left\langle /\,\boxed{0}\,, \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & /\,\boxed{1} \\ \text{VAL} & \begin{bmatrix} \text{MOD} & \boxed{A} \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle \\
\text{OUTPUT} & \left\langle /\,\boxed{0}\,, \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & /\,\boxed{1} \\ \text{VAL} & \begin{bmatrix} \text{MOD} & \boxed{A} \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle
\end{bmatrix}
$$

pi-rule also inherits the defeasible identity constraint on SEM from *l-rule*.

Now that we have the type *pi-rule*, we can state the Extraposition Lexical Rule:

(44) Extraposition Lexical Rule

$$
\begin{bmatrix}
\textit{pi-rule} & \\
\text{INPUT} & \left\langle \text{X}\,, \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\, \boxed{2}\text{CP}\, \rangle \\ \text{COMPS} & \boxed{A} \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle \\
\text{OUTPUT} & \left\langle \text{Y}\,, \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\, \text{NP[FORM it]}\, \rangle \\ \text{COMPS} & \boxed{A} \oplus \langle \boxed{2} \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle
\end{bmatrix}
$$

This rule creates new words from any word whose first argument is a CP (or can be resolved to CP). The output word always takes a dummy *it* as its subject and takes as a final argument whatever kind of CP was specified as the input's first argument. Notice that this rule, unlike the Passive Lexical Rule, is formulated in terms of SPR and COMPS, not ARG-ST. The ARG-ST values will be supplied by the ARP (as the outputs are still of type *word*).

This analysis is strikingly simple. All we needed was a new value of FORM (it), and a new subtype of *l-rule*.[11] Then we were able to formulate a lexical rule that captures the regularity illustrated by the sentence pairs at the beginning of this section. We do not need any new phrase structure rules to handle extraposition. Any word structure formed from one of the outputs of this rule fits one of the general patterns already provided for by our existing grammar rules. Furthermore, this lexical rule as written also accounts for extraposition with adjectives (see Problem 6) and interacts correctly with our analysis of passive (Problem 7).

[11] We'll see more instances of *pi-rule* in Chapter 13.

11.5 Idioms

We have now encountered two nonreferential NPs with highly restricted distributions, namely, the dummies *there* and *it*. Other NPs that share the properties of nonreferentiality and restricted distribution can be found in idioms – that is, in fixed (or partially fixed) combinations of words that are used to express meanings that aren't determined in the usual way from those words. For example:

(45) a. Carrie kicked the bucket last night. ('Carrie died last night')
b. The FBI kept (close) tabs on Sandy. ('The FBI (carefully) observed Sandy')
c. The candidates take (unfair) advantage of the voters. ('The candidates exploit the voters (unfairly)')

The idioms *kick the bucket*, *keep tabs on*, and *take advantage of* each have an idiosyncratic meaning, which requires that all of its parts co-occur. That is, the words in these idioms take on their idiomatic meanings only when they appear together with the other parts of the idioms. For example, the following sentences do not have interpretations related to those in (45):

(46) a. Chris dreads the bucket.
b. The police put tabs on undocumented workers.
c. The candidates bring advantage to the voters.

Since the lexical entries for verbs contain information about the arguments they co-occur with (but not vice versa), one way to capture the idiosyncratic properties of idioms is to encode them in the entries of the verbs. That is, we can treat idiomatic nouns, such as *tabs* and *advantage* by:

- giving them their own FORM values;
- marking them as [MODE none] and [INDEX none]; and
- specifying that they are [RESTR $\langle\ \rangle$]

This amounts to treating idiom parts (or 'idiom chunks', as they are often called) in much the same way that we treated the dummies *there* and *it* in the previous sections of this chapter.

We can now have entries for *keep* and *take* specifying that their objects must be [FORM tabs] and [FORM advantage], respectively. These verbal entries will contain all of the idioms' semantic information.[12] The detailed entries for idiomatic nouns *tabs* and *advantage* and the verbs that go with them are given in (47) and (48):[13]

[12] This treatment (like a number of others in this book) is a simplification. For a more thorough discussion of (some of) the authors' views on the semantics of idioms, see Nunberg et al. 1994 and Sag et al. 2002.

[13] You might think that *tabs* and *advantage* are irregular in another way, namely in not occurring with a determiner. But in fact, there are examples where idiomatic nouns do combine with determiners:

(i) Sandy and Kim resented the tabs that were being kept on them by the Attorney General.
(ii) We all regret the unfair advantage that has been taken of the situation by those unwilling to exercise fundamental caution.

(47) a.
$$\left\langle \text{tabs}\ , \begin{bmatrix} \textit{cntn-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{tabs} \\ \text{AGR} & \begin{bmatrix} \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{none} \\ \text{INDEX} & \text{none} \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

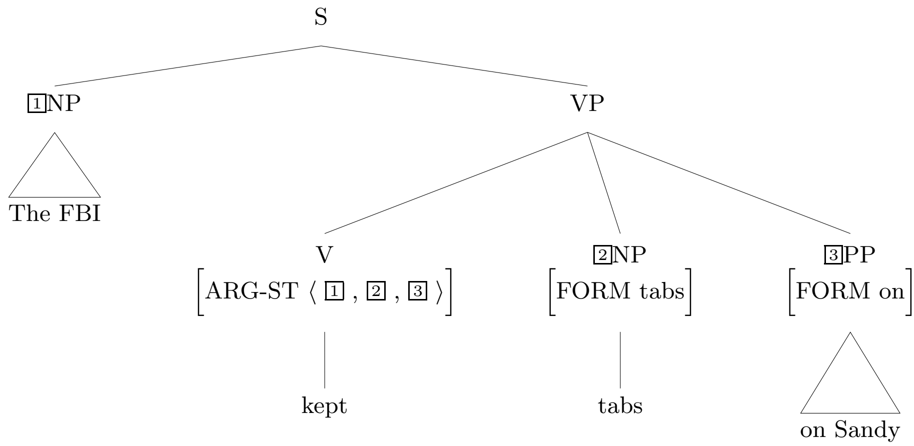
b.
$$\left\langle \text{advantage}\ , \begin{bmatrix} \textit{massn-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{advantage} \\ \text{AGR} & \textit{3sing} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{none} \\ \text{INDEX} & \text{none} \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(48) a.
$$\left\langle \text{keep}\ , \begin{bmatrix} \textit{ptv-lxm} & \\ \text{ARG-ST} & \left\langle \text{NP}_i\ , \begin{bmatrix} \text{FORM tabs} \end{bmatrix}, \begin{bmatrix} \text{FORM} & \text{on} \\ \text{INDEX} & j \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{observe} \\ \text{SIT} & s \\ \text{OBSERVER} & i \\ \text{OBSERVED} & j \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

b.
$$\left\langle \text{take}\ , \begin{bmatrix} \textit{ptv-lxm} & \\ \text{ARG-ST} & \left\langle \text{NP}_i\ , \begin{bmatrix} \text{FORM advantage} \end{bmatrix}, \begin{bmatrix} \text{FORM} & \text{of} \\ \text{INDEX} & j \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{exploit} \\ \text{SIT} & s \\ \text{EXPLOITER} & i \\ \text{EXPLOITED} & j \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Using these lexical entries, we get tree structures like the following:

(49)



Notice that we have given no entry for *kick the bucket*. There is a reason for this: different idioms exhibit different syntactic behavior, so not all idioms should be analyzed in the same fashion. In particular, *kick the bucket* differs from *keep tabs on* and *take advantage of* in its lack of a passive form. That is, while (50a,b) allow idiomatic interpretations, (50c) can only convey its literal meaning, which entails that Pat's foot made contact with a real bucket.

(50) a. Tabs are kept on suspected drug dealers by the FBI.
b. Advantage is taken of every opportunity for improvement.
c. The bucket was kicked by Pat.

The analysis of *keep tabs on* and *take advantage of* presented above correctly allows them to have passive forms. These idiomatic verb entries meet the input conditions of the Passive Lexical Rule, and so can give rise to passive forms. The FORM restrictions on the NP complements of the active idiomatic verbs are restrictions on the subjects (that is, the SPR element) of their passive versions. Hence, idiomatic *taken* (as a passive) requires that its subject be *advantage*.

If *kick the bucket* were to be analyzed in a parallel fashion, we would incorrectly predict that (50c) had an idiomatic interpretation (namely, 'Pat died'). To avoid this, we need a different analysis of this idiom. The most straightforward treatment is to say that the whole string, *kick the bucket*, is the verb.[14] Thus, there is a single lexical entry for the entire idiom *kick the bucket*, given in (51):

(51)

$$\left\langle \langle \text{kick, the, bucket} \rangle , \begin{bmatrix} \textit{siv-lxm} \\ \text{SEM} \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{die} \\ \text{CORPSE} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

[14]In order to ensure that the verbal morphology appears on the first word in this multiword lexical entry, we adopt the general convention that morphological functions apply only to the first element of such entries. This also covers a number of other cases, such as the locations of the plural *-s* in *runs batted in* and *attorneys general*, and the comparative suffix *-er* in *harder of hearing*.

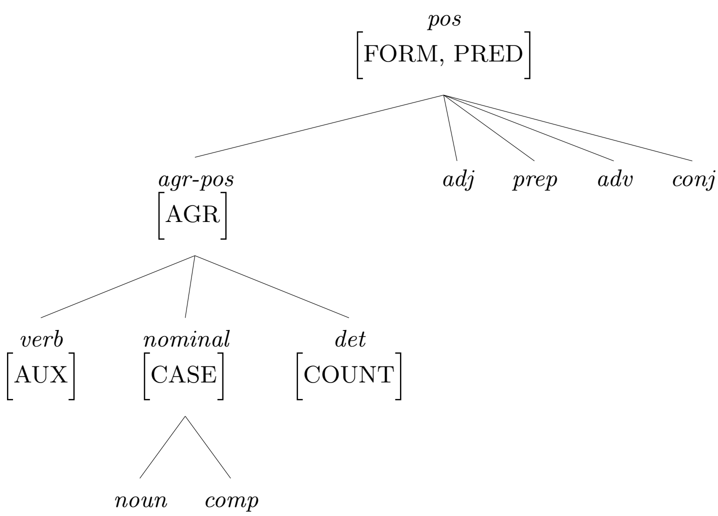
This entry is a strict-intransitive multi-element verbal lexeme, so it doesn't have a passive form. Or, to put it more formally, entry (51) does not satisfy the conditions necessary to serve as input to the Passive Lexical Rule: since it is not a *tv-lxm*, it does not passivize.

11.6 Summary

In this chapter, we have extended the use of the FORM feature to NPs and made use of it in the analysis of existential sentences containing the dummy *there*, the extraposition construction, and idioms. Each of these three constructions involves nonreferential NPs. The distribution of such NPs is more than an idle curiosity, however. In more complex sentences, it plays a crucial role in motivating the analysis of infinitival and other kinds of complements, which is precisely the concern of the next chapter.

11.7 Changes to the Grammar

In this chapter we introduced a revision to the type hierarchy under the type *pos*, adding the types *nominal* and *comp* and adding the feature PRED on *pos*.



We introduced a new value of FORM (cform) and made the type *comp* subject to the following constraint:

$$\textit{comp} : \begin{bmatrix}\text{FORM} & \text{cform}\end{bmatrix}$$

Now that CASE is appropriate for *comp* as well as *noun*, we revised the Case Constraint:

Case Constraint: An outranked NP or CP is [CASE acc].

In addition we introduced a type *comp-lxm* (a daughter type of *const-lxm*), subject to the following type constraint:

$$
comp\text{-}lxm : \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} comp \\ \text{AGR} & 3sing \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \begin{matrix} \text{S} \\ \begin{bmatrix} \text{INDEX} & s \end{bmatrix} \end{matrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \end{bmatrix}
$$

We introduced one lexical entry that instantiates *comp-lxm*:

$$
\left\langle \text{that}\ ,\ \begin{bmatrix} comp\text{-}lxm \\ \text{ARG-ST} & \langle \begin{bmatrix} \text{FORM fin} \end{bmatrix} \rangle \\ \text{SEM} & \begin{bmatrix} \text{MODE prop} \end{bmatrix} \end{bmatrix} \right\rangle
$$

To allow for CP subjects, we revised the constraint on the ARG-ST of *verb-lxm*. We also added the constraint that instances of *verb-lxm* are [PRED −]. With these revisions, *verb-lxm* is now subject to the following constraints:

$$
verb\text{-}lxm : \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} verb \\ \text{PRED} & - \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \begin{bmatrix} \text{HEAD} & nominal \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix}, \ldots \right\rangle \end{bmatrix}
$$

We also generalized the requirement on the second argument (first complement) of *tv-lxm* from NP to a [HEAD *nominal*] saturated constituent, so that the constraint on this type is now:

$$
\begin{bmatrix} \text{ARG-ST} & \left\langle \text{X}\ ,\ \begin{bmatrix} \text{HEAD} & nominal \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix}, \ldots \right\rangle \end{bmatrix}
$$

In order to make passive and present participles [PRED +], we revised the Passive Lexical Rule and the Present Participle Lexical Rule:

Passive Lexical Rule

$$
\left[\begin{array}{ll}
\textit{d-rule} & \\
\text{INPUT} & \left\langle \boxed{1}\ ,\ \left[\begin{array}{ll}
\textit{tv-lxm} & \\
\text{SYN} & \left[\text{HEAD}\ \left[\text{PRED}\ -\right]\right] \\
\text{ARG-ST} & \langle\ [\text{INDEX}\ i]\ \rangle\ \oplus\ \boxed{\text{A}}
\end{array}\right]\right\rangle \\
\text{OUPUT} & \left\langle \text{F}_{PSP}(\boxed{1})\ ,\ \left[\begin{array}{ll}
\textit{part-lxm} & \\
\text{SYN} & \left[\text{HEAD}\ \left[\begin{array}{ll}\text{FORM} & \text{pass}\\ \text{PRED} & +\end{array}\right]\right] \\
\text{ARG-ST} & \boxed{\text{A}}\ \oplus\ \left\langle \left(\begin{array}{c}\text{PP}\\ \left[\begin{array}{ll}\text{FORM} & \text{by}\\ \text{INDEX} & i\end{array}\right]\end{array}\right)\right\rangle
\end{array}\right]\right\rangle
\end{array}\right]
$$

Present Participle Lexical Rule

$$
\left[\begin{array}{ll}
\textit{d-rule} & \\
\text{INPUT} & \left\langle \boxed{3}\ ,\ \left[\begin{array}{ll}
\textit{verb-lxm} & \\
\text{SYN} & \left[\text{HEAD}\ \left[\text{PRED}\ -\right]\right] \\
\text{SEM} & \left[\text{RESTR}\ \boxed{\text{A}}\right]
\end{array}\right]\right\rangle \\
\text{OUTPUT} & \left\langle \text{F}_{PRP}(\boxed{3})\ ,\ \left[\begin{array}{ll}
\textit{part-lxm} & \\
\text{SYN} & \left[\text{HEAD}\ \left[\begin{array}{ll}\text{FORM} & \text{prp}\\ \text{PRED} & +\end{array}\right]\right] \\
\text{SEM} & \left[\text{RESTR}\ \boxed{\text{A}} \oplus \ldots.\right]
\end{array}\right]\right\rangle
\end{array}\right]
$$

We encountered a new subtype of *l-rule* (*pi-rule*) for post-inflectional lexical rules:

$$
\begin{array}{ccc}
 & \textit{l-rule} & \\
\swarrow & \downarrow & \searrow \\
\textit{d-rule} & \textit{i-rule} & \textit{pi-rule}
\end{array}
$$

$$
pi\text{-}rule : \left[\begin{array}{ll}
\text{INPUT} & \left\langle / \boxed{0} , \left[\begin{array}{ll} \multicolumn{2}{l}{word} \\ \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & / \boxed{1} \\ \text{VAL} & \left[\text{MOD } \boxed{A}\right] \end{array}\right] \end{array}\right] \right\rangle \\
\text{OUTPUT} & \left\langle / \boxed{0} , \left[\begin{array}{ll} \multicolumn{2}{l}{word} \\ \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & / \boxed{1} \\ \text{VAL} & \left[\text{MOD } \boxed{A}\right] \end{array}\right] \end{array}\right] \right\rangle
\end{array}\right]
$$

The Extraposition Lexical Rule is an instance of *pi-rule*:

Extraposition Lexical Rule

$$
\left[\begin{array}{ll}
\multicolumn{2}{l}{pi\text{-}rule} \\
\text{INPUT} & \left\langle \text{X} , \left[\text{SYN} \left[\text{VAL} \left[\begin{array}{ll} \text{SPR} & \langle \boxed{2}\text{CP} \rangle \\ \text{COMPS} & \boxed{A} \end{array}\right]\right]\right] \right\rangle \\
\text{OUTPUT} & \left\langle \text{Y} , \left[\text{SYN} \left[\text{VAL} \left[\begin{array}{ll} \text{SPR} & \langle \text{NP[FORM it]} \rangle \\ \text{COMPS} & \boxed{A} \oplus \langle \boxed{2} \rangle \end{array}\right]\right]\right] \right\rangle
\end{array}\right]
$$

Semantically empty lexical entries were also introduced in this chapter. One key property of semantically empty lexical entries is that they are [INDEX none]. Previously, INDEX could only take something of type *index* as its value. We revise the constraint on the type *sem-cat* to allow the specification [INDEX none]:

$$
sem\text{-}cat : \left[\begin{array}{ll}
\text{MODE} & \left\{\text{prop, ques, dir, ref, ana, none}\right\} \\
\text{INDEX} & \left\{\textit{index}, \text{none}\right\} \\
\text{RESTR} & \textit{list}(\textit{predication})
\end{array}\right]
$$

The following semantically empty lexical entries were introduced:

$$
\left\langle \text{advantage} , \left[\begin{array}{ll}
\multicolumn{2}{l}{massn\text{-}lxm} \\
\text{SYN} & \left[\text{HEAD} \left[\begin{array}{ll} \text{FORM} & \text{advantage} \\ \text{AGR} & \textit{3sing} \end{array}\right]\right] \\
\text{SEM} & \left[\begin{array}{ll} \text{MODE} & \text{none} \\ \text{INDEX} & \text{none} \\ \text{RESTR} & \langle\ \rangle \end{array}\right]
\end{array}\right] \right\rangle
$$

$$\left\langle \text{tabs}\ ,\ \begin{bmatrix} \textit{cntn-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{tabs} \\ \text{AGR} & \begin{bmatrix} \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{none} \\ \text{INDEX} & \text{none} \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

$$\left\langle \text{there}\ ,\ \begin{bmatrix} \textit{pron-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{there} \\ \text{AGR} & \begin{bmatrix} \text{PER}\ \text{3rd} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{none} \\ \text{INDEX} & \text{none} \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

$$\left\langle \text{it}\ ,\ \begin{bmatrix} \textit{pron-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{it} \\ \text{AGR} & \textit{3sing} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{none} \\ \text{INDEX} & \text{none} \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Finally, this chapter introduced the following lexical entries for verbs:

$$\left\langle \text{be}\ ,\ \begin{bmatrix} \textit{be-lxm} & \\ \text{ARG-ST} & \left\langle \boxed{1}\ ,\ \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & [\text{PRED}\ \ +] \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

$$\left\langle \text{be}\ , \begin{bmatrix} \textit{exist-be-lxm} & \\ \text{ARG-ST} & \left\langle \begin{matrix}\text{NP}\\ \begin{bmatrix}\text{FORM} & \text{there}\end{bmatrix}\end{matrix}, \boxed{2}\ , \begin{bmatrix} \text{PRED} & + \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle \boxed{2} \rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s\end{bmatrix} \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s \\ \text{RESTR} & \langle\ \rangle\end{bmatrix} \end{bmatrix} \right\rangle$$

$$\left\langle \text{matter}\ , \begin{bmatrix} \textit{siv-lxm} & \\ \text{ARG-ST} & \left\langle \begin{bmatrix}\text{SEM [INDEX } \boxed{1}]\end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{matter} \\ \text{SIT} & s \\ \text{MATTERING} & \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

$$\left\langle \text{keep}\ , \begin{bmatrix} \textit{ptv-lxm} & \\ \text{ARG-ST} & \left\langle \text{NP}_i\ , \begin{bmatrix}\text{FORM tabs}\end{bmatrix}, \begin{bmatrix}\text{FORM} & \text{on} \\ \text{INDEX} & j\end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{observe} \\ \text{SIT} & s \\ \text{OBSERVER} & i \\ \text{OBSERVED} & j \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

$$\left\langle \text{take}\ , \begin{bmatrix} \textit{ptv-lxm} & \\ \text{ARG-ST} & \left\langle \text{NP}_i\ , \begin{bmatrix}\text{FORM advantage}\end{bmatrix}, \begin{bmatrix}\text{FORM} & \text{of} \\ \text{INDEX} & j\end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{exploit} \\ \text{SIT} & s \\ \text{EXPLOITER} & i \\ \text{EXPLOITED} & j \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

11.8 Further Reading

Influential early discussions of the existential *there* and extraposition include Rosenbaum 1967, Milsark 1977 and Emonds 1975. See also Chomsky 1981 and Postal and Pullum 1988. Of the many generative discussions of idioms, see especially Fraser 1970, Chomsky 1980, Ruwet 1991, Nunberg et al. 1994, Riehemann 2001 and Villavicencio and Copestake 2002. A number of papers on idioms are collected in Cacciari and Tabossi 1993 and Everaert et al. 1995.

11.9 Problems

Problem 1: *There* and Agreement

The analysis of existential *there* sentences presented so far says nothing about verb agreement.

A. Consult your intuitions (and/or those of your friends, if you wish) to determine what the facts are regarding number agreement of the verb in *there* sentences. Give an informal statement of a generalization covering these facts, and illustrate it with both grammatical and ungrammatical examples. [*Note: Intuitions vary regarding this question, across both individuals and dialects. Hence there is more than one right answer to this question.*]

B. How would you elaborate or modify our analysis of the *there* construction so as to capture the generalization you have discovered? Be as precise as you can.

Problem 2: Santa Claus

There is another type of sentence with expletive *there*, illustrated in (i).

(i) Yes, Viriginia, there is a Santa Claus.

A. Why can't the lexical entry for *be* in (11) be used in this sentence?

B. Give a lexical entry for the lexeme *be* that gives rise to *is* in (i).

C. With the addition of your lexical entry for part (B), does (ii) become ambiguous, according to the grammar? Why or why not?

(ii) There is a book on the table.

⚠ **Problem 3: Passing Up the Index**

A. Give the RESTR value that our grammar should assign to the sentence in (i). Be sure that the SIT value of the *smoke* predication is identified with the ANNOYANCE value of the *annoy* predication.

   (i) That Dana is smoking annoys Leslie.

   [*Hint: This sentence involves two of the phenomena analyzed in this chapter: predicative complements of be and CP subjects. Refer to (5) for the relevant lexical entry for be and (37) for the relevant lexical entry for that.*]

B. Draw a tree structure for the sentence in (i). You may use abbreviations for the node labels, but be sure to show the INDEX value on all of the nodes.

C. Explain how the SIT value of the *smoke* predication gets identified with the ANNOYANCE value of the *annoy* predication. Be sure to make reference to lexical entries, phrase structure rules, and principles, as appropriate.

Problem 4: An Annoying Problem

Assume that the lexical entry for the verb *annoy* is the following:

(i)
$$\left\langle \text{annoy}\ , \begin{bmatrix} \textit{stv-lxm} & \\ \text{ARG-ST} & \left\langle \begin{bmatrix}\text{SEM [INDEX } \boxed{1}]\end{bmatrix}, \text{NP}_i \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{annoy} \\ \text{SIT} & s \\ \text{ANNOYED} & i \\ \text{ANNOYANCE} & \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

A. What constraints are imposed on the lexical sequences that result from applying the 3rd-Singular Verb Lexical Rule to this entry (including those that involve inheritance of constraints from the entry's supertypes)?

B. What constraints are imposed on lexical sequences that result from applying the Extraposition Lexical Rule to your answer to part (A)?

C. Draw a tree structure for the sentence in (ii). You should show the value of all SEM features on all of the nodes, as well as the SPR and COMPS features for *annoys*.

   (ii) It annoys Lee that Fido barks.

   [*Hint: The lexeme for the complementizer that is shown in (38). The SEM value of the phrase that Fido barks should end up being the same as (35c).*]

D. The lexical entry for *annoy* allows NP subjects as well, as in (iii). Why doesn't the grammar then also license (iv)?

   (iii) Sandy annoys me.
   (iv)*It annoys me Sandy.

Problem 5: Optional *That*

As noted in Section 11.4.1, most verbs that can take a CP complement can also take an S complement:

(i) I guessed Alex might be suspicious.
(ii) Dana knows Leslie left.
(iii) Everyone assumed Lee would win.

A. What is the ARG-ST value of *guessed* in (i)? [*Note: You may use abbreviations in formulating this ARG-ST.*]

B. Formulate a new subtype of *verb-lxm* for verbs with this ARG-ST value. [*Note: Be sure to rule out ungrammatical strings like *I guessed Alex being suspicious.*]

C. Formulate a derivational lexical rule that relates transitive verbs (i.e. instances of subtypes of *tv-lxm*) to S-complement taking verbs. [*Hints: The type of feature structure in the OUTPUT value should be the type you posited in part (B). Also, your rule should ensure that whatever semantic role is played by the CP argument of the input is played by the S argument of the output.*]

While a verb like *assume* can appear with an NP, CP or S complement, in the passive, it can only take an NP or CP subject:

(iv) The responsibility was assumed by no one.

(v) That Lee would win was assumed by everyone.

(vi)*Lee would win was assumed by everyone.

D. Does your rule interact with the Passive Lexical Rule (Chapter 10) to (incorrectly) license (vi)? If not, why not? If so, how could you fix it so that it doesn't?

Problem 6: Extraposition and Adjectives

In our discussion of extraposition, we focussed on verbs, but in fact, extraposition is a more general phenomenon. Adjectives which take CP subjects show the same alternation:

(i) That the ad works is obvious.

(ii) It is obvious that the ad works.

Note that it won't do to say that it is *be* alone that licenses the extraposition in these examples. Adjectives show up in the extraposed valence pattern without *be*:[15]

(iii) I find it obvious that the ad works.

A. Find two other examples of adjectives that take CP subjects, and show that they also allow the extraposed valence pattern (examples with *be* are fine).

As noted in Section 11.4.2, our Extraposition Lexical Rule is formulated so as to apply to adjectives. The input only requires something with a CP as the first member of its argument list, and says nothing specific to verbs.

B. Write a lexical entry for *obvious* or one of the extraposing adjectives you supplied.

C. Give the OUTPUT value of the Extraposition Lexical Rule when your adjective is the INPUT.[16]

D. Give the tree structure for (ii). Abbreviated node labels are acceptable, but be sure to indicate SPR and COMPS values on all nodes.

---

[15] Our grammar at present cannot generate examples like (iii). We will see how to handle them in the next chapter.

[16] Of course, the Constant Lexeme Lexical Rule has to apply first, to make a *word* from the adjective lexeme. This *word* will a suitable INPUT for the Extraposition Lexical Rule.

Problem 7: Passive and Extraposition

The example in (i) involves both extraposition and passive:

(i) It was assumed that the ad worked.

A. Give the lexical entry for *assume*.

B. In order to get from *assume* to the passivized, extraposed word in (i), three lexical rules must apply. Passive and Extraposition are two, what is the third, and which order do they apply in?

C. Show the OUTPUT value of each of the lexical rules.

D. Give the tree structure for (i). Abbreviated node labels are acceptable, but be sure to indicate SPR and COMPS values on all nodes.

Problem 8: Idiomatic *Kept*

A. Show the passive lexeme based on the lexical entry for idiomatic *kept* – that is, the result of applying the Passive Lexical Rule to (48a).

B. Explain precisely how the contrast between (i) and (ii) is explained on our analysis:

(i) Tabs were kept on Chris by the FBI.

(ii)*Advantage was kept on Chris by the FBI.

Be sure to discuss the role of the verb *be*.

12

Infinitival Complements

12.1 Introduction

So far in this book, we have seen two examples of sentences expressing complex meanings, i.e. sentences in which one situation functions as a semantic argument of another.[1] The first was sentences with modifiers such as *today*, discussed in Chapter 8. The second was sentences involving extraposition, discussed in the last chapter. In this chapter, we will investigate additional constructions that involve semantic embedding. In particular, we will focus on infinitival complements in sentences such as (1):

(1) a. Pat continues to avoid conflict.
    b. Pat tries to avoid conflict.

We will see that, despite their superficial parallelism, examples (1a) and (1b) are quite different in their semantics and in certain associated syntactic properties. These two examples are representative of two basic ways in which propositions can be combined into complex meanings.

12.2 The Infinitival *To*

Before we delve into the subtle properties that distinguish (1a) from (1b), we need to provide an analysis for the word *to* that appears in both sentences. Like the lexemes we will consider in Chapter 13, the infinitival *to* functions as an auxiliary ([AUX +]) verb.[2] But it is a peculiar verb, one that has only a nonfinite form. In order to allow other verbs to select for VPs headed by *to*, we will need a way of distinguishing it (and the phrases it projects) from (the projections of) all other verbs. To this end, we introduce a new binary feature INF. The lexical entry for infinitival *to* will be specified as [INF +], whereas all other verbs will be [INF −]. We will in fact make [INF / −] a defeasible constraint on the type *verb-lxm*, one that is overridden only by *to*. Since *to* will also be

[1] As we noted, the semantic analysis we have given for a sentence like *That dogs bark annoys people* (or its extraposed counterpart) involves not the embedding of one feature structure within another, but rather the identification of the SIT value of one predication with the ARG value of another.

[2] Among the properties of *to* that lead us to call it an auxiliary verb is the fact that, like all auxiliary verbs, it may undergo VP-Ellipsis:

(i) Do you think they will go? They will __ .
(ii) Do you think they will go? They want to __ .

specified as [FORM base] in its lexical entry, it will not be able to undergo any lexical rule that specifies a different FORM value. Thus, only one kind of word will result from *to* – the kind that is the output of the Base Form Lexical Rule.

In addition, *to*, like the verb *be*, does not contribute to the semantics of the sentences in any substantive way. This is evident in those cases where it is optional. For example, there is no apparent difference in meaning between (2a) and (2b) or between (3a) and (3b):

(2) a. Pat helped Chris [to solve the problem].
    b. Pat helped [Chris solve the problem].

(3) a. They wouldn't dare [to attack us].
    b. They wouldn't dare [attack us].

Data like (2) and (3), by the way, provide independent motivation for treating infinitival *to* as [FORM base], as that analysis allows us to write lexical entries for *help* and *dare* that select for a VP[FORM base] complement, leaving the INF value unspecified.

The following lexical entry for *to* will allow our analysis to capture all these properties:

(4)
$$\left\langle \text{to}\ ,\ \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{base} \\ \text{INF} & + \\ \text{AUX} & + \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \boxed{1}\ ,\ \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{INF} & - \\ \text{FORM} & \text{base} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

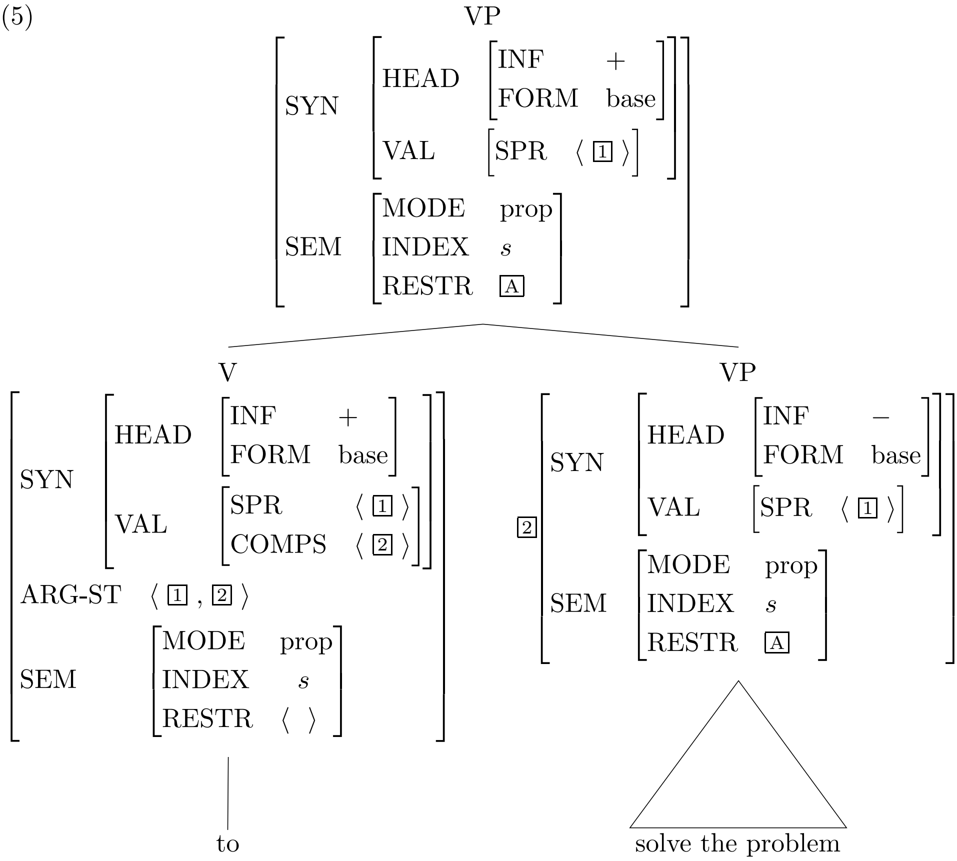
We haven't specified the type that this entry is an instance of, because it is a new type *auxiliary-verb-lexeme* (*auxv-lxm*), to be discussed in Chapter 13. We will find that *to* shares many of the above properties with other verbs, and we will be able to state these generalizations as constraints on that type. For the moment, it is sufficient to note that *auxv-lxm* is a subtype of the type *subject-raising-verb-lexeme* (*srv-lxm*; discussed in Section 12.3 below), and therefore *to* is a kind of *verb-lxm*. This means that *to* will also inherit all of the constraints associated with *verb-lxm*, *srv-lxm*, and *auxv-lxm* that are not overridden by constraints in its lexical entry.

The semantic emptiness of *to* is modeled in this lexical entry by the specification [RESTR ⟨ ⟩] and the fact that it shares the INDEX value of its VP complement. From these constraints, it follows that when *to* combines with its VP complement, only the latter contributes to the semantic restriction of the resulting VP. The rest of the constraints on the ARG-ST of *to* specify that it takes a VP complement that is both [INF −] and

[FORM base] (such as *bake a cake* or *be a hero*) as its second argument, and the SPR requirement of that VP as its first argument.

Once we include this somewhat unusual nonfinite verb in our lexicon, our grammar rules and principles interact to license structures like the following:

(5)



Structures like these will be the complement of verbs like *continue* and *try*, which are the topics of the next two sections.

Exercise 1: *To Fix This!

Given the analysis of infinitives just introduced, our grammar will now incorrectly generate imperative sentences like the following:

(i)*To get out of here!

(ii)*To stop that!

This overgeneration can be prevented by making a minor revision to our grammar. What is it?

12.3 The Verb *Continue*

Recall that the dummies *it* and *there*, as well as idiom chunks like *(close) tabs* or *(unfair) advantage*, have a restricted distribution – they occur only as subjects or objects of verbs that select them in those positions. What these NPs all have in common is that they are nonreferential – that is, they take 'none' as their value for MODE and INDEX. They are therefore inherently unsuited to play a role in any predication. Consequently, on semantic grounds, we have already explained the ungrammaticality of (6) and the fact that *it* must be referential in (7), as we noted in Chapter 11:

(6) a. $\text{*I hate} \left\{ \begin{array}{l} \text{advantage} \\ \text{tabs} \\ \text{there} \end{array} \right\}.$

b. $* \left\{ \begin{array}{l} \text{Advantage} \\ \text{tabs} \\ \text{there} \end{array} \right\} \text{really affected us.}$

(7) a. I hate it.

b. It really affected us.

It might seem surprising, then, that there are some other verbs that allow subject NPs that lack referential indices. *Continue* is one such example:

(8) a. Sandy continues to eat oysters.

b. There continued to be no easy answer to the dilemma.

c. It continues to bother me that Chris lied.

d. (Close) tabs continue to be kept on Bo by the FBI.

e. (Unfair) advantage continues to be taken of the refugees.

Let's consider this phenomenon more carefully. Suppose we have a finite VP like *eats oysters*. This VP, as we have seen, requires a referential subject, rather than a nonreferential one like *there*, (dummy) *it*, or *advantage*. The pattern that we find here is that whenever a verb phrase imposes such a constraint on its subject, then a larger VP made up of *continues to* or *continued to* plus the original VP (with the head verb in the base form) must obey the same constraint. There is a correlation: if the subject of *eats oysters* has to be referential, then so does the subject of *continues/continued to eat oysters*. Similarly, a finite VP like *is no compromise on this issue* must combine with a dummy *there* as its subject (even the dummy *it* is disallowed). Correlated with this is the fact that the larger VP *continued to be no compromise on this issue* also requires a dummy *there* as its subject. The same is true for VPs like *bothers me that Chris lied*, *were kept on Bo by the FBI*, and *was taken of the refugees*. These VPs require subjects that are dummy *it*, (*close*) *tabs*, and (*unfair*) *advantage*, respectively.[3] And for each of these

[3]In the last two cases, there are other subjects that can appear with superficially identical VPs. This is because the verbs *take* and *keep* participate in multiple different idioms in English, as illustrated in (i):

(i) Good care was taken of the refugees.

Under our current analysis of idioms, (i) would involve a different lexical entry for *take* which selects for an NP[FORM care]. The important point for the current discussion is that the range of possible subjects for *continues to be taken of the refugees* is exactly the same as the range of possible subjects for *was taken of the refugees*.

verbs, their '*continue-to-be*' counterpart exhibits exactly the same requirements. These theoretically critical contrasts are summarized in (9a-d):

(9) a.
$$\text{There continues to} \left\{ \begin{array}{l} \text{ be no easy answer to the dilemma} \\ \text{*eat oysters} \\ \text{*bother me that Chris lied} \\ \text{*be kept on Bo by the FBI} \\ \text{*be taken of the refugees} \end{array} \right\}.$$

b.
$$\text{It continues to} \left\{ \begin{array}{l} \text{ bother me that Chris lied} \\ \text{*eat oysters} \\ \text{*be no easy answer to the dilemma} \\ \text{*be kept on Bo by the FBI} \\ \text{*be taken of the refugees} \end{array} \right\}.$$

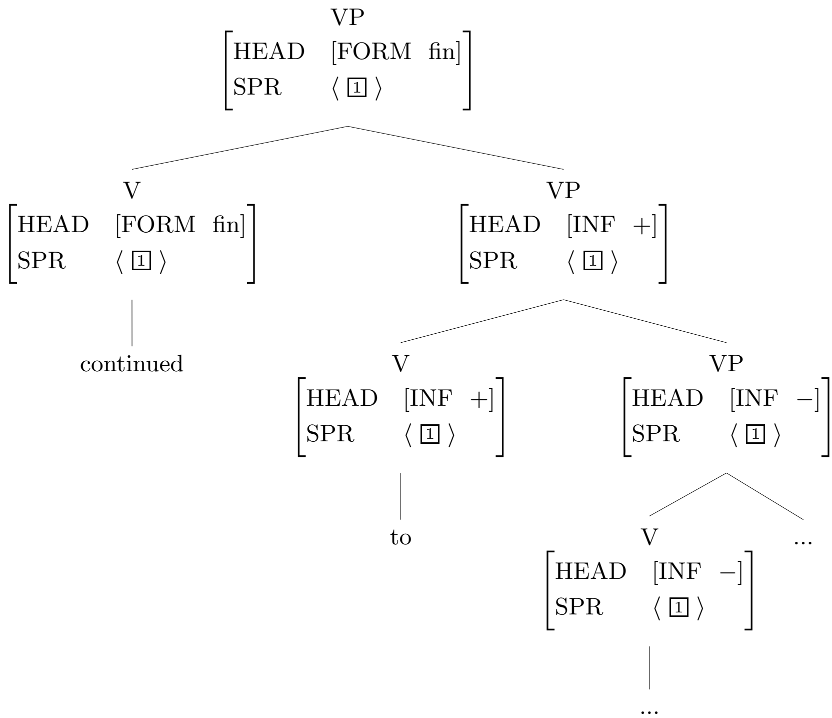
c.
$$\text{(Close) tabs continue to} \left\{ \begin{array}{l} \text{ be kept on Bo by the FBI} \\ \text{*eat oysters} \\ \text{*be no easy answer to the dilemma} \\ \text{*bother me that Chris lied} \\ \text{*be taken of the refugees} \end{array} \right\}.$$

d.
$$\text{(Unfair) advantage continues to} \left\{ \begin{array}{l} \text{ be taken of the refugees} \\ \text{*eat oysters} \\ \text{*be no easy answer to the dilemma} \\ \text{*bother me that Chris lied} \\ \text{*be kept on Bo by the FBI} \end{array} \right\}.$$

The contrasts illustrated in (9) suggest that the verb *continue* is intuitively TRANSPARENT to the selectional demands that its VP complement imposes on its subject. That is, a verb like *continue* heads a VP that requires the same kind of subject that its VP complement requires.

We can capture this intuition by simply specifying that *continue* and its complement must have the same subject. We do this as we did earlier for the passive *be* and for the infinitival *to* above: the first element in *continue*'s ARG-ST list (the subject) will be identical to the SPR value of the second element in the ARG-ST list. Since the complement is a VP headed by *to*, the SPR value of the VP *continue to…* will be identical to the SPR value of the embedded VP. Hence the co-occurrence restrictions involving the nonreferential NPs will be transmitted from the verbs heading the infinitival VPs, through the infinitival *to*, up to the subject of the verb *continue*, as illustrated in (10):

(10)



Thus we have an account for the first striking property of the verb *continue*: it places no restrictions of its own on its subject, but rather takes as a subject whatever kind of subject its VP complement is looking for.

A second, related property of *continue* is that it doesn't do anything semantically with its subject. We can see that by comparing sentences with active and passive verbs in the VP complement of *continue*. One such pair of examples is given in (11):

(11) a. The FBI continued to visit Lee.
 b. Lee continued to be visited by the FBI.

In (11a), the complement of *to* is a VP headed by the verb *visit*. In (11b), the complement of *to* is a VP headed by *be* which in turn takes as a complement headed by *visited*, the passive form of *visit*. In what follows, we will informally describe sentences like (11a) and (11b) simply as 'active-passive pairs' to have a simple way of referring to them since we will use them as a diagnostic. Pairs like this, i.e. pairs like NP_1 *continued to* V NP_2 and NP_2 *continued to be* V-ed by NP_1, have essentially the same meaning. That is, examples (11a) and (11b) are very close paraphrases of one another.[4]

In (11a) *the FBI* is a syntactic argument of *continue* and *Lee* isn't. In (11b) it is *Lee* that is a syntactic argument of *continue*, while *the FBI* isn't. The fact that these two sentences mean the same thing suggests that in neither case is the subject of *continue*

[4]We say 'very close' because there are subtle differences in emphasis between the two sentences. The crucial test, for our purposes, is that there are no conceivable conditions under which one of the sentences would be true and the other would be false. This is the operational test we will use throughout to determine whether sentences do or do not mean the same thing.

one of its semantic arguments. Rather, semantically, *continue* takes only one argument – the situation of its infinitival complement – and predicates of it that it continues to be the case. Thus, both sentences in (11) mean that it continues to be the case that the FBI visits Lee. Formally, we represent this as in (12):

(12)
$$\begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s_1 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{The FBI} \\ \text{NAMED} & i \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{continue} \\ \text{SIT} & s_1 \\ \text{ARG} & s_2 \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{visit} \\ \text{SIT} & s_2 \\ \text{VISITOR} & i \\ \text{VISITED} & j \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{Lee} \\ \text{NAMED} & j \end{bmatrix} \right\rangle \end{bmatrix}$$

Note that the **continue** predication has only one role slot (called ARG) and this is filled by the situational index of the **visit** predication (s_2). There is no role in the **continue** predication for either the index of *the FBI* or the index of *Lee*. This semantic fact is crucial not only to the active-passive paraphrase property of *continue*, but also to the first property we discussed: if *continue* were to assign a semantic role to its subject, it would be unable to accept nonreferential subjects like dummy *it* and *there* and idiom chunks ((*unfair*) *advantage*, (*close*) *tabs*, etc.).

Since *continue* is not an isolated example, but rather representative of a class of verbs (including *to*), we will posit a lexical type *subject-raising-verb-lexeme* (*srv-lxm*).[5] We thus postulate the following lexical type, which is a kind of (i.e. an immediate subtype of) *verb-lxm*:

(13) *subject-raising-verb-lxm* (*srv-lxm*):
$$\begin{bmatrix} \text{ARG-ST} & \left\langle \boxed{1}, \begin{bmatrix} \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{INDEX} & s_2 \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{RESTR} & \left\langle \begin{bmatrix} \text{ARG} & s_2 \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

With this type constraint in place, we can assign *continue* the following streamlined lexical entry:

[5]The perhaps nonmnemonic terms that permeate this discussion – 'raising' and 'control' verbs – reflect commonly used terminology in the field. They derive from the analysis of this distinction that was developed in transformational grammar (see Appendix B).

(14)
$$\left\langle \text{continue}\ , \begin{bmatrix} \textit{srv-lxm} & \\ \text{ARG-ST} & \left\langle \text{X}\ , \begin{array}{c}\text{VP}\\ \begin{bmatrix}\text{INF} & +\end{bmatrix}\end{array}\right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s_1 \\ \text{RESTR} & \left\langle \begin{bmatrix}\text{RELN} & \textbf{continue} \\ \text{SIT} & s_1\end{bmatrix}\right\rangle \end{bmatrix}\end{bmatrix}\right\rangle$$

In this analysis, the lexeme *continue* inherits information not only from the type *srv-lxm* but also from the supertype *verb-lxm.* The lexical sequences satisfying this lexical entry are schematized in (15), which also displays all of the inherited constraints:

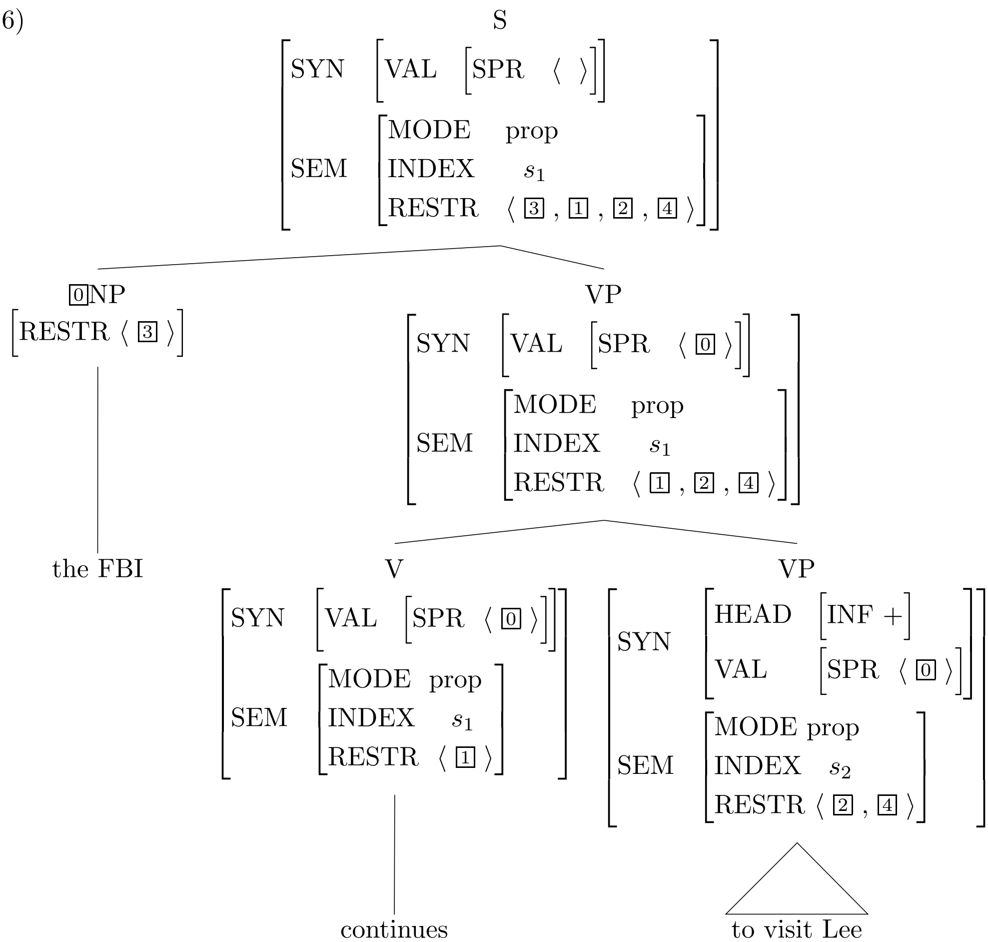
(15)
$$\left\langle \text{continue}\ , \begin{bmatrix} \textit{srv-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{PRED} & - \\ \text{INF} & - \\ \text{AGR} & \boxed{2} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ [\text{AGR}\ \boxed{2}]\ \rangle\end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \boxed{1}\begin{bmatrix} \text{HEAD} & \textit{nominal} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix}\end{bmatrix}\ , \begin{array}{c}\text{VP}\\ \begin{bmatrix}\text{INF} & + \\ \text{SPR} & \langle\ \boxed{1}\ \rangle \\ \text{INDEX} & s_2\end{bmatrix}\end{array}\right\rangle \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s_1 \\ \text{RESTR} & \left\langle \begin{bmatrix}\text{RELN} & \textbf{continue} \\ \text{SIT} & s_1 \\ \text{ARG} & s_2\end{bmatrix}\right\rangle \end{bmatrix}\end{bmatrix}\right\rangle$$

Our analysis derives all of the following:

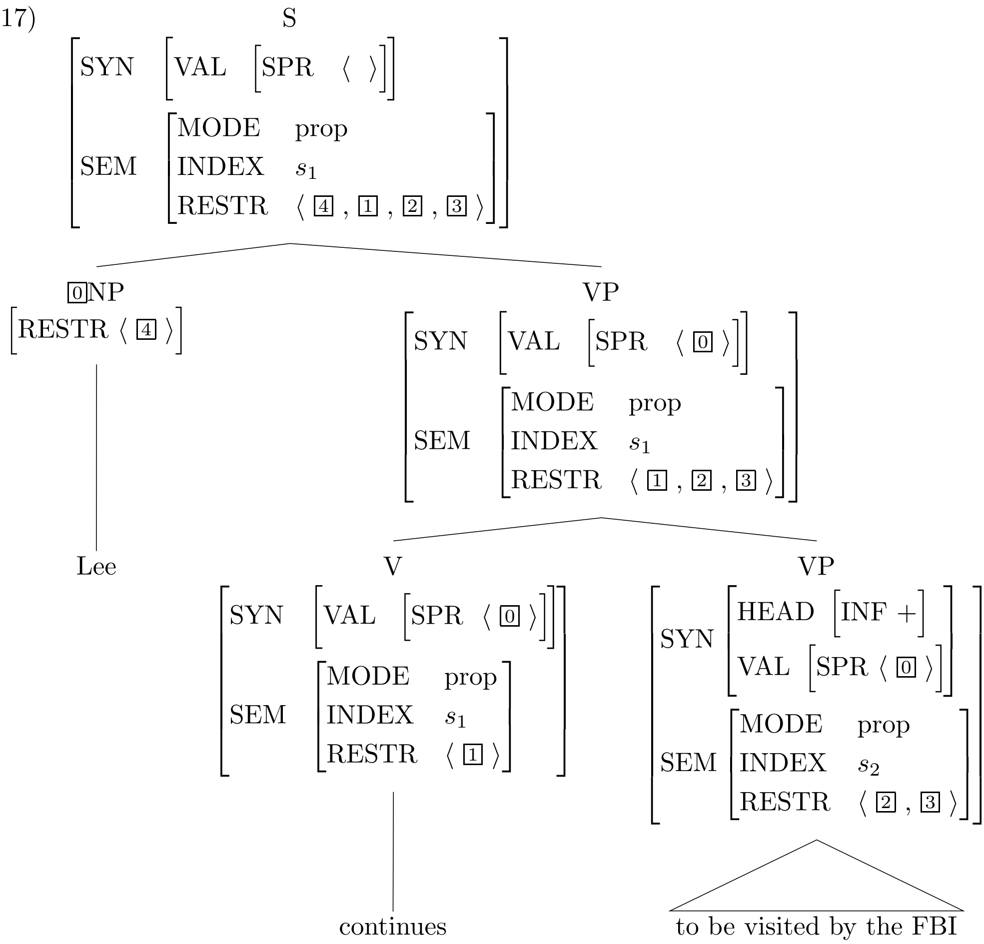
- the VP complement of *continue* is infinitival
- the VP complement of *continue* is its semantic argument (since (14) inherits the relevant constraint from the type *srv-lxm*),
- the subject of *continue* is the subject of the VP complement (since (14) inherits the relevant constraint from the type *srv-lxm*),
- the subject of *continue* plays no role in the **continue** predication, and
- as a result of the above points, the sentences in (11) are assigned equivalent semantic analyses.

These properties are illustrated in (16) and (17): (Note that the tags $\boxed{1}$–$\boxed{4}$ refer to the same feature structure descriptions in (16)–(18).)

(16)



(17)



Here the relevant predications are those given earlier and tagged appropriately in (18):

(18)

$$\begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s_1 \\ \text{RESTR} & \left\langle \boxed{4}\begin{bmatrix}\text{RELN} & \textbf{name} \\ \text{NAME} & \text{Lee} \\ \text{NAMED} & j\end{bmatrix}, \boxed{1}\begin{bmatrix}\text{RELN} & \textbf{continue} \\ \text{SIT} & s_1 \\ \text{ARG} & s_2\end{bmatrix}, \boxed{2}\begin{bmatrix}\text{RELN} & \textbf{visit} \\ \text{SIT} & s_2 \\ \text{VISITOR} & i \\ \text{VISITED} & j\end{bmatrix}, \boxed{3}\begin{bmatrix}\text{RELN} & \textbf{name} \\ \text{NAME} & \text{The FBI} \\ \text{NAMED} & i\end{bmatrix} \right\rangle \end{bmatrix}$$

As discussed in Chapter 5, the order of elements on the RESTR list has no semantic significance. Hence, since the semantic values assigned to these two sentences differ only in the order of elements on the RESTR list, active-passive pairs like these are correctly predicted to be semantically equivalent.

12.4 The Verb *Try*

The analysis of the verb *continue* that we just developed was motivated by two observations: (i) that *continue* is transparent to co-occurrence restrictions between its subject and its complement's verb; and (ii) that active-passive pairs like those discussed in the previous section are paraphrases.

Turning to the superficially similar verb *try*, we see that it differs from *continue* with respect to both (i) and (ii). Thus the analogs to (8b–e), with nonreferential subjects, are systematically ill formed (even though the verb embedded in *try*'s complement does indeed select for the relevant nonreferential subject):

(19) a. Sandy tried to eat oysters.
b.*There tried to be riots in Freedonia.
c.*It tried to bother me that Chris lied.
d.*(Close) tabs try to be kept on Bo by the FBI.
e.*(Unfair) advantage tries to be taken of the refugees.

Likewise, the following two sentences are not synonymous:

(20) a. The FBI tried to find Lee.
b. Lee tried to be found by the FBI.

(20a) could be true under circumstances where (20b) would be false; indeed, it is quite likely that most people whom the FBI is trying to find are not trying to be found by them (or by anybody else!). Since the analysis of *continue* was designed to account for points (i) and (ii) above, it is clear that we need to analyze *try* quite differently.

Let us begin with the semantics of *try*. Unlike **continue** predications, which take only one semantic role (ARG, whose value is a situation), predications of trying involve two things: an individual (the entity that is trying) and some situation or state of affairs that the trier is trying to bring about. This is why the examples in (20) differ in meaning: the two triers are not the same. Notice also what the trier is trying to bring about always involves the trier. That is, it is not possible to express a meaning in which, say, what Kim is trying is for Sandy to visit Bo.[6] These remarks are synthesized in the following semantic structure for *Sandy tries to visit Bo*:

[6] Maybe you could force an interpretation on this, something like 'Kim tried to bring it about that Sandy visit Bo', but notice that in so doing you are coercing the interpretation of the complement to a meaning that DOES contain the trier. We will ignore such coercions here.

(21)
$$\begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s_1 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{Sandy} \\ \text{NAMED} & i \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{try} \\ \text{SIT} & s_1 \\ \text{TRIER} & i \\ \text{ARG} & s_2 \end{bmatrix}, \right. \\ & \left. \begin{bmatrix} \text{RELN} & \textbf{visit} \\ \text{SIT} & s_2 \\ \text{VISITOR} & i \\ \text{VISITED} & j \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{Bo} \\ \text{NAMED} & j \end{bmatrix} \right\rangle \end{bmatrix}$$

Semantic structures like this immediately rule out the use of nonreferential subjects (i.e. dummies and idiom chunks) with *try*. This is because the subject position of *try* always corresponds to a semantic argument, namely the TRIER. Since nonreferential NPs are specified as [INDEX none], it follows that there can be no semantics for examples like (19b–e). The index value of the TRIER role cannot be identified with the subject NP's index if the subject has no index.

Just as *continue* is representative of a class of verbs (RAISING verbs), *try* is representative of another class, called CONTROL verbs. In general, the control verbs assign a semantic role to their subject, while the raising verbs do not. From this critical difference, it follows that raising verbs can take nonreferential subjects while control verbs cannot, and that raising verbs allow active-passive pairs to be paraphrases, while control verbs do not.

As before, we will want to use lexical types to express constraints that apply generally to verbs of the control class. So we will want to introduce another subtype of *verb-lxm* like the one shown in (22):

(22) *subject-control-verb-lxm* (*scv-lxm*):
$$\begin{bmatrix} \text{ARG-ST} & \left\langle \text{NP}_i\ , \begin{bmatrix} \text{SPR} & \langle\ \text{NP}_i\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{INDEX} & s_2 \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{RESTR} & \left\langle \begin{bmatrix} \text{ARG} & s_2 \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

The lexical entry for *try* can now be given in the streamlined form shown in (23):

(23)

$$\left\langle \text{try}\ ,\ \begin{bmatrix} \textit{scv-lxm} & \\ \text{ARG-ST} & \left\langle \text{NP}_i\ ,\ \begin{matrix}\text{VP}\\ \begin{bmatrix}\text{INF}\ +\end{bmatrix}\end{matrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s_1 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{try} \\ \text{SIT} & s_1 \\ \text{TRIER} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Lexical sequences satisfying (23) thus inherit all the constraints shown in (24):

(24)

$$\left\langle \text{try}\ ,\ \begin{bmatrix} \textit{scv-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{PRED} & - \\ \text{INF} & - \\ \text{AGR} & \boxed{1} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ [\text{AGR}\ \boxed{1}]\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \text{NP}_i\ ,\ \begin{matrix}\text{VP}\\ \begin{bmatrix} \text{INF} & + \\ \text{SPR} & \langle\ \text{NP}_i\ \rangle \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s_2\end{bmatrix} \end{bmatrix}\end{matrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s_1 \\ \text{MODE} & \text{prop} \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{try} \\ \text{SIT} & s_1 \\ \text{TRIER} & i \\ \text{ARG} & s_2 \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Note that the first argument of *try* and the subject of the VP are not identified; only their indices are. The subject-sharing analysis is necessary for raising verbs, because verbs like *continue* select for exactly the kind of subject that their complements select for. This includes information contained in the FORM value in the case of idiom chunks and dummy subjects, but also other HEAD information and the VAL values. At the same time, it is important that the index of the subject of *continue* be the same as the index of the subject of the embedded verb. This is because the subject can play a semantic role with respect to the embedded verb (when it is referential). Therefore, in order to get the semantics right, we need to ensure that the index of the subject is available to the embedded verb. The smallest feature structure containing all of the relevant values is the entire *expression*.

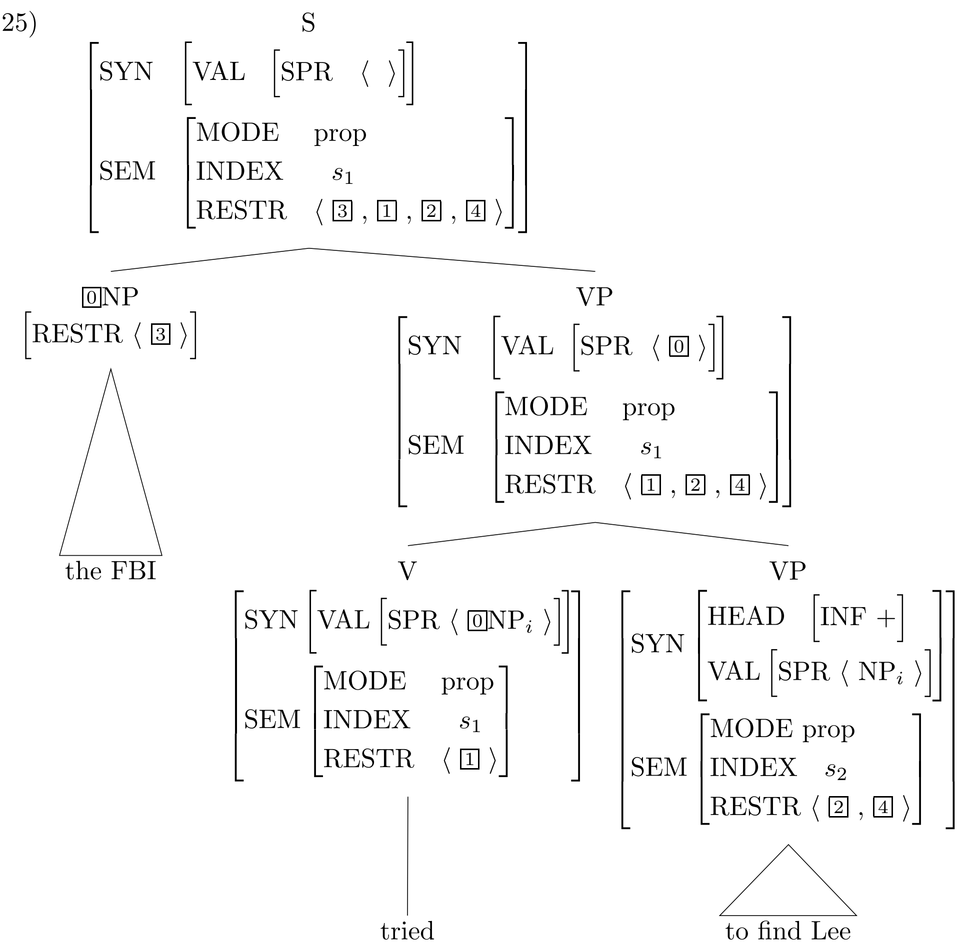
Judging only from the facts we've seen so far, we could also use the subject-sharing analysis for control verbs (like *try*). However, there is no data that requires sharing any information beyond the indices, so we take the more conservative step of sharing only what is needed. In fact, it turns out that data from other languages motivate this difference in the analyses of raising and control verbs. This point is developed in Problem 5.

Our analysis of control verbs like *try* guarantees that:

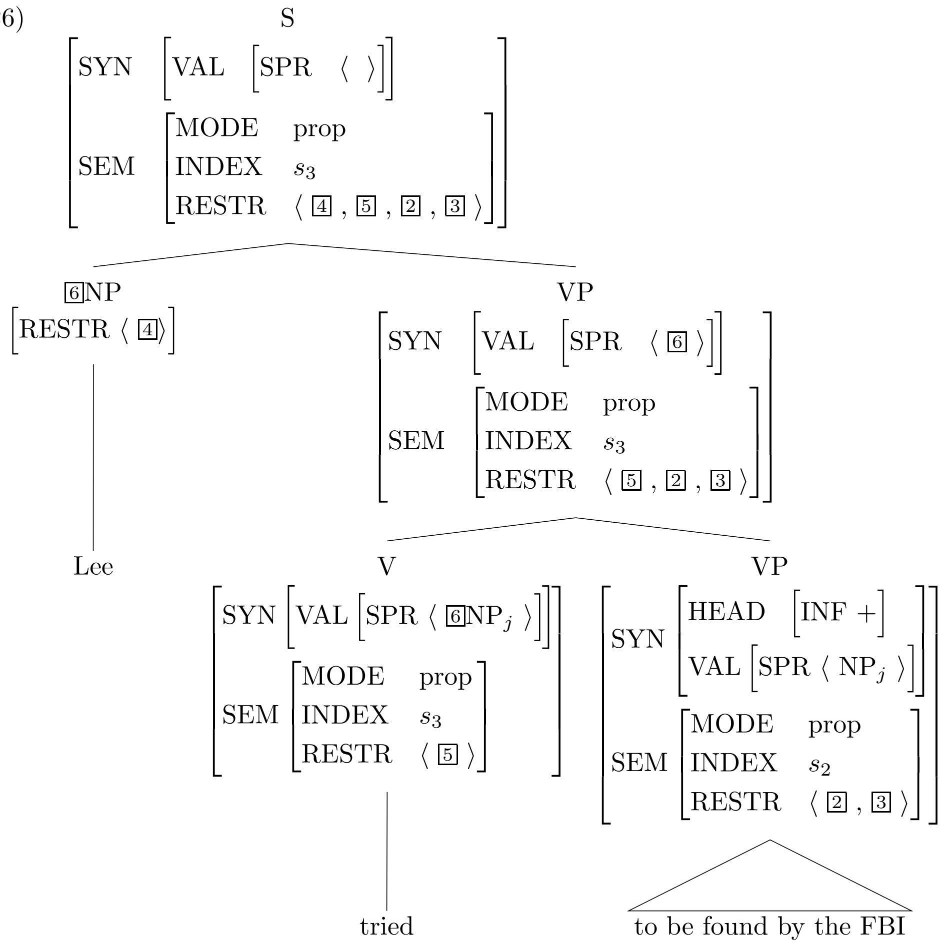
- The complement of *try* is an infinitival VP,
- the VP complement is a semantic argument of the **try** predication (since (23) inherits the relevant constraint from the type *scv-lxm*),
- the subject of *try* is assigned to the TRIER role; and hence
- nonreferential NPs can never be the subject of *try*,
- the infinitival complements of *try* can never be of a kind that requires a nonreferential subject (because they must have an index identified with the trier), and
- that (20a) and (20b) have different meanings (because in one case the FBI is the trier and in the other, Lee is).

This analysis is illustrated in the following pair of semantically contrasting examples:

(25)



(26)



The first of these has the semantics shown in (27):

(27)
$$\begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s_1 \\ \text{RESTR} & \left\langle \boxed{3}\begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{The FBI} \\ \text{NAMED} & i \end{bmatrix}, \boxed{1}\begin{bmatrix} \text{RELN} & \textbf{try} \\ \text{SIT} & s_1 \\ \text{TRIER} & i \\ \text{ARG} & s_2 \end{bmatrix}, \right. \\ & \left. \boxed{2}\begin{bmatrix} \text{RELN} & \textbf{find} \\ \text{SIT} & s_2 \\ \text{FINDER} & i \\ \text{FOUND} & j \end{bmatrix}, \boxed{4}\begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{Lee} \\ \text{NAMED} & j \end{bmatrix} \right\rangle \end{bmatrix}$$

In contrast, the sentence with the passive complement in (26) has the semantics given in (28), where the trier is *j*, the index of Lee, not the FBI.

(28)
$$\begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s_3 \\ \text{RESTR} & \left\langle \boxed{4}\begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{Lee} \\ \text{NAMED} & j \end{bmatrix}, \boxed{5}\begin{bmatrix} \text{RELN} & \textbf{try} \\ \text{SIT} & s_3 \\ \text{TRIER} & j \\ \text{ARG} & s_2 \end{bmatrix}, \right. \\ & \left. \boxed{2}\begin{bmatrix} \text{RELN} & \textbf{find} \\ \text{SIT} & s_2 \\ \text{FINDER} & i \\ \text{FOUND} & j \end{bmatrix}, \boxed{3}\begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{The FBI} \\ \text{NAMED} & i \end{bmatrix} \right\rangle \end{bmatrix}$$

By positing a lexical distinction between raising and control verbs in the hierarchy of lexemes, we thus correctly account for their differing properties without adjusting our grammar rules or any other aspect of our theory.

12.5 Subject Raising and Subject Control

As noted above, the verbs *continue* and *try* are representative of the classes subject raising verb and subject control verb, respectively. To review the properties of these classes, subject raising verbs like *continue* express properties of situations, allow nonreferential subjects, and give rise to paraphrastic active-passive pairs like those examined above. Subject control verbs like *try*, on the other hand, express a relation between an individual and a situation, never take nonreferential subjects, and fail to give rise to analogous paraphrastic active-passive pairs.

In fact, it is not just verbs that can be divided into these two classes; there are also raising adjectives and control adjectives. They are exemplified in (29), with the diagnostic properties illustrated in (30)–(33).[7]

(29) a. Pat is likely to scream.
b. Pat is eager to scream.

(30) a. There is likely to be a letter in the mailbox.
b. It is likely to upset Pat that Chris left.
c. Tabs are likely to be kept on participants.
d. Advantage is likely to be taken of unwary customers.

(31) a.*There is eager to be a letter in the mailbox.
b.*It is eager to upset Pat that Chris left.
c.*Tabs are eager to be kept on participants.
d.*Advantage is eager to be taken of unwary customers.

(32) The doctor is likely to examine Pat. ≈ Pat is likely to be examined by the doctor.

(33) The doctor is eager to examine Pat. ≠ Pat is eager to be examined by the doctor.

This suggests that our system of lexical types should be somewhat more abstract (perhaps introducing a type like *subject-raising-lxm* as a supertype of *srv-lxm* and a similar type of

[7] Here we use the symbol '≈' to indicate sameness of truth conditions, and '≠' to indicate difference of truth conditions.

adjectival lexeme), in order to accommodate generalizations that cut across the various part of speech distinctions such as verb vs. adjective.[8]

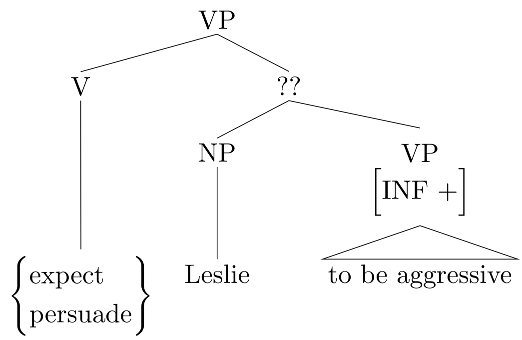
12.6 Object Raising and Object Control

Consider now two new verbs: *expect* and *persuade*. These two verbs are similar in that both can occur in examples like the following:

(34) a. I expected Leslie to be aggressive.
b. I persuaded Leslie to be aggressive.

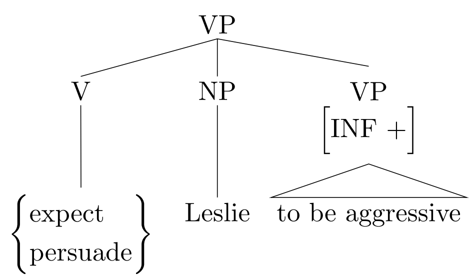
There are two possible analyses one could imagine for these verbs. There could be some kind of phrase that includes both the NP and the infinitival VP *to be aggressive*, as in:

(35)



Alternatively, it is possible that the NP is the direct object of the verb and the infinitival VP is also a complement of the verb:

(36)



But in fact, only the latter structure is consistent with the analyses of other phenomena presented in earlier chapters. We will return to why this is so at the end of this section. First, we briefly consider the analyses we will give to these verbs.

The difference between *expect* and *persuade* in structures like (36) is analogous to the distinction we just drew between *continue* and *try*. Just as the subject of *continue* plays no semantic role with respect to the **continue** predication, the object of *expect* plays no role with respect to the **expect** predication. Rather, in both cases, the semantic role of the NP in question is whatever the complement's verb assigns to its subject. Similarly, the object of *persuade* is like the subject of *try* in that it plays a semantic role with respect to the **persuade** predication while also playing the semantic role assigned to the subject of the complement's verb. *Expect* is an example of what is usually called

[8]This matter is taken up again in Chapter 16.

an ‘object raising’ verb and *persuade* is an ‘object control’ verb. Hence we will want to introduce the two types in (37) with the indicated constraints and then provide lexical entries for *expect* and *persuade* like the ones shown in (38):

(37) a. *object-raising-verb-lxm* (*orv-lxm*):

$$\begin{bmatrix} \text{ARG-ST} & \left\langle \text{NP}\ ,\ \boxed{1}\ ,\ \begin{bmatrix}\text{SPR} & \langle\ \boxed{1}\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{INDEX} & s_2\end{bmatrix}\right\rangle \\ & \\ \text{SEM} & \begin{bmatrix}\text{RESTR} & \left\langle [\text{ARG} \quad s_2] \right\rangle\end{bmatrix}\end{bmatrix}$$

b. *object-control-verb-lxm* (*ocv-lxm*):

$$\begin{bmatrix} \text{ARG-ST} & \left\langle \text{NP}\ ,\ \text{NP}_i\ ,\ \begin{bmatrix}\text{SPR} & \langle\ \text{NP}_i\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{INDEX} & s_2\end{bmatrix}\right\rangle \\ & \\ \text{SEM} & \begin{bmatrix}\text{RESTR} & \left\langle [\text{ARG} \quad s_2] \right\rangle\end{bmatrix}\end{bmatrix}$$

(38) a.

$$\left\langle \text{expect}\ ,\ \begin{bmatrix} \textit{orv-lxm} & \\ & \\ \text{ARG-ST} & \left\langle\ \text{NP}_j\ ,\ \text{X}\ ,\ \begin{array}{c}\text{VP}\\ [\text{INF}\ +]\end{array}\right\rangle \\ & \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s \\ & \\ \text{RESTR} & \left\langle \begin{bmatrix}\text{RELN} & \textbf{expect} \\ \text{SIT} & s \\ \text{EXPECTER} & j\end{bmatrix}\right\rangle\end{bmatrix}\end{bmatrix}\right\rangle$$

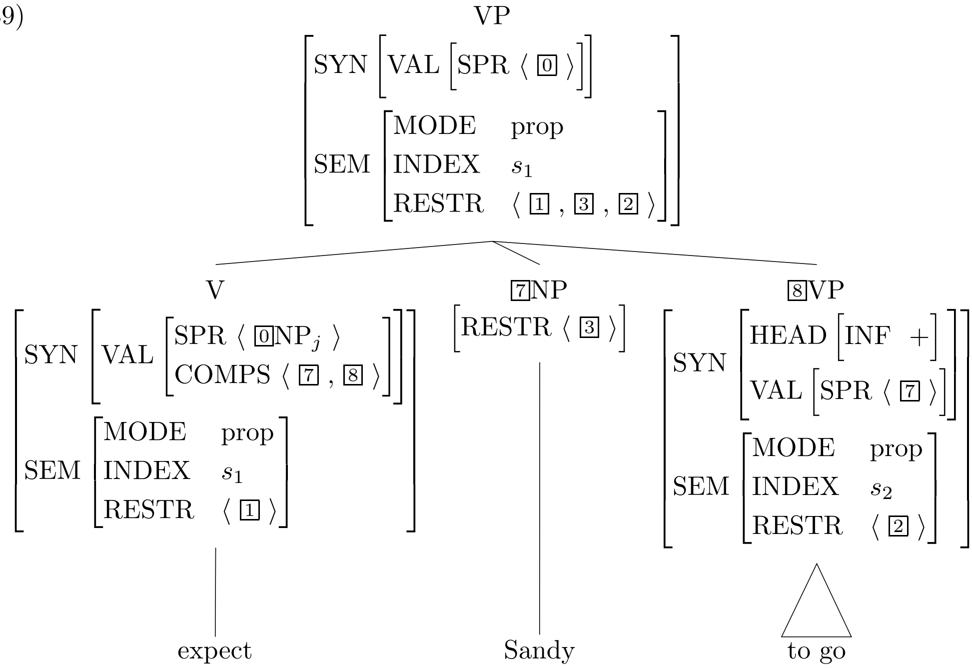
b.

$$\left\langle \text{persuade}\ ,\ \begin{bmatrix} \textit{ocv-lxm} & \\ & \\ \text{ARG-ST} & \left\langle\ \text{NP}_j\ ,\ \text{NP}_i\ ,\ \begin{array}{c}\text{VP}\\ [\text{INF}\ +]\end{array}\right\rangle \\ & \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s \\ & \\ \text{RESTR} & \left\langle \begin{bmatrix}\text{RELN} & \textbf{persuade} \\ \text{SIT} & s \\ \text{PERSUADER} & j \\ \text{PERSUADEE} & i\end{bmatrix}\right\rangle\end{bmatrix}\end{bmatrix}\right\rangle$$

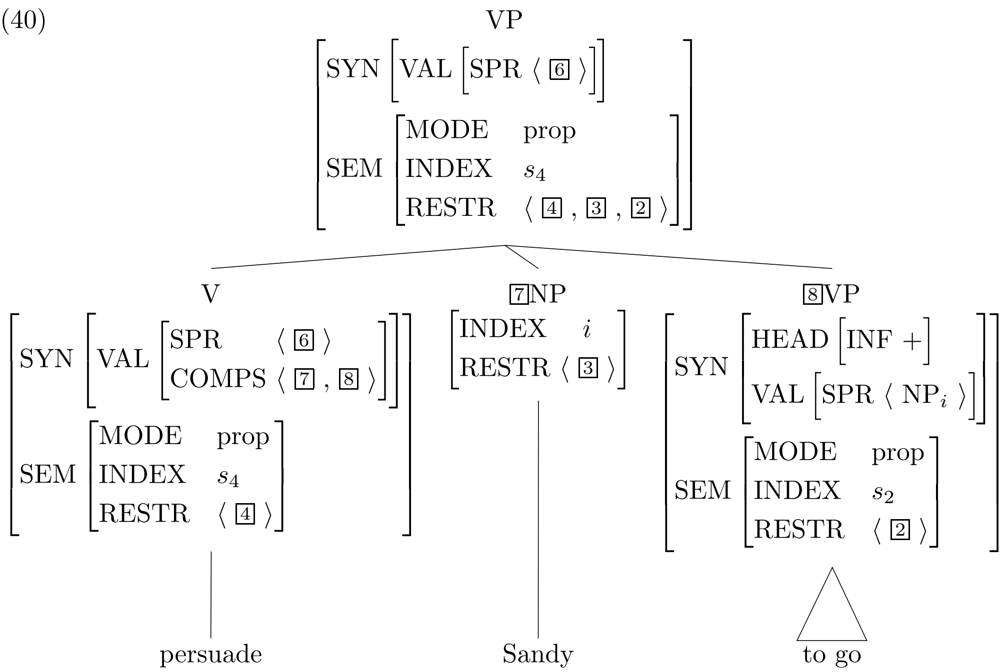
Notice that the contrast between the types *orv-lxm* and *ocv-lxm* is analogous to the contrast between *srv-lxm* and *scv-lxm*. The type *orv-lxm* specifies that the second argument is the same as the specifier of the third argument ($\boxed{1}$). In addition, the second argument isn’t assigned any role in the predication in the entry for the object raising verb *expect*. In contrast, the type *ocv-lxm* specifies that the index of the second argument is the same as the specifier of the third argument. Further, the second argument of *persuade* is assigned a role (PERSUADEE) in the **persuade** predication.

The active words derived from these lexemes will then give rise to structures like the following:

(39)



(40)



And the semantic analyses associated with these structures are as shown in (41) and (42):

(41)
$$\begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s_1 \\ \text{RESTR} & \left\langle \boxed{1}\begin{bmatrix} \text{RELN} & \textbf{expect} \\ \text{SIT} & s_1 \\ \text{EXPECTER} & j \\ \text{ARG} & s_2 \end{bmatrix}, \boxed{3}\begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{Sandy} \\ \text{NAMED} & i \end{bmatrix}, \boxed{2}\begin{bmatrix} \text{RELN} & \textbf{go} \\ \text{SIT} & s_2 \\ \text{GOER} & i \end{bmatrix} \right\rangle \end{bmatrix}$$

(42)
$$\begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s_4 \\ \text{RESTR} & \left\langle \boxed{4}\begin{bmatrix} \text{RELN} & \textbf{persuade} \\ \text{SIT} & s_4 \\ \text{PERSUADER} & j \\ \text{PERSUADEE} & i \\ \text{ARG} & s_2 \end{bmatrix}, \boxed{3}\begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{Sandy} \\ \text{NAMED} & i \end{bmatrix}, \boxed{2}\begin{bmatrix} \text{RELN} & \textbf{go} \\ \text{SIT} & s_2 \\ \text{GOER} & i \end{bmatrix} \right\rangle \end{bmatrix}$$

We are now in a position to discuss why the structure in (36) is compatible with our grammar so far and why the structure in (35) isn't. Consider the following passive sentences:

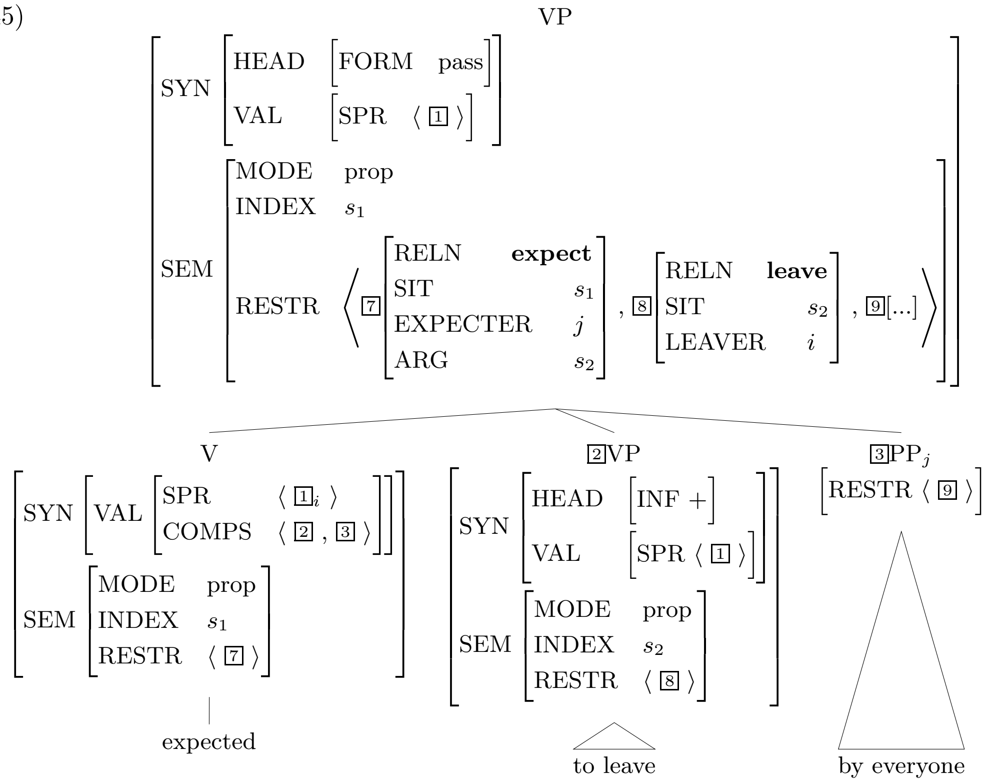
(43) a. Chris was expected to leave (by everyone).
b. Chris was persuaded to leave (by Ashley).

These examples are predicted to be grammatical by our analysis, assuming the type constraints in (37) and the lexical entries in (38). The lexical entry for *expect* in (38a) will give rise to the passive word sketched in (44):

(44)
$$\left\langle \text{expected}\ , \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{pass} \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \boxed{1}\ , \overset{\text{VP}}{\begin{bmatrix} \text{INF} & + \\ \text{SPR} & \langle \boxed{1} \rangle \\ \text{INDEX} & s_2 \end{bmatrix}} \Big(, \text{PP[by]}_j\Big) \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s_1 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{expect} \\ \text{SIT} & s_1 \\ \text{EXPECTER} & j \\ \text{ARG} & s_2 \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

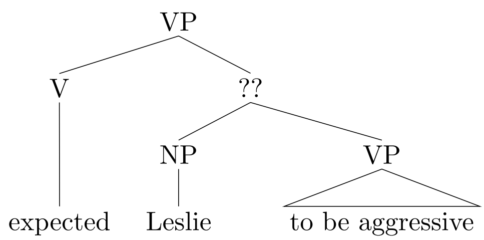
And this word will give rise to structures like (45) (analogous to (36)), which are precisely what we need to accommodate examples like (43a):

(45)



If, on the other hand, the structure in (35) (repeated here as (46)) were the correct structure for active sentences like (34), we would predict the passive examples in (43) to be ungrammatical.

(46)



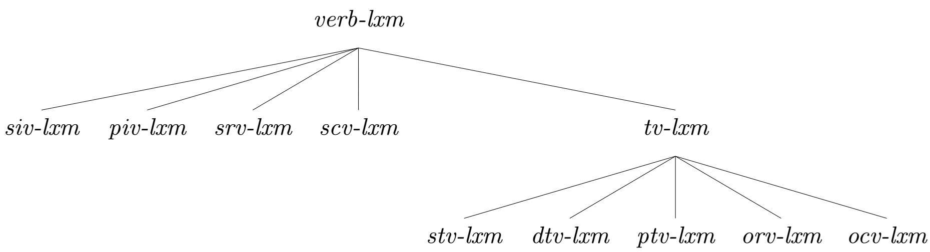
If structures like (46) were correct, then the lexical entries for these verbs would involve a doubleton ARG-ST list containing the subject NP and some kind of infinitival phrase that included the NP. But since passivization involves a rearrangement of the ARG-ST list, i.e. a lexical rule that 'promotes' an object NP to become the first argument of the passive verb form, such putative lexical entries would give us no way to analyze examples like (43). We would need to assume some passivization mechanism beyond those that are, as we saw in Chapter 10, independently motivated in our grammar. We conclude that the structure in (36) and the constraints we have posited on *orv-lxm* and *ocv-lxm* are correct.

12.7 Summary

This chapter explored further subtleties in the patterned distribution of nonreferential NPs. These patterns led us to posit a fundamental difference between two kinds of verbs: raising verbs, which select one ARG-ST member assigned no semantic role, and control verbs, which are superficially similar, but which assign a semantic role to each member of their ARG-ST list. We explored the various subclasses of raising and control verbs, including the defective infinitival verb *to* and concluded by examining the interaction of our proposed analysis with the passive analysis introduced in Chapter 10.

12.8 Changes to the Grammar

In this chapter, we revised the type hierarchy, introducing the new lexeme types: *subject-raising-verb-lxm* (*srv-lxm*), *subject-control-verb-lxm* (*scv-lxm*), *object-raising-verb-lxm* (*orv-lxm*), and *object-control-verb-lxm* (*ocv-lxm*). The hierarchy under *verb-lxm* now looks like this:



We also introduced the binary feature INF(INITIVE), appropriate for feature structures of type *verb*. The type *verb-lxm* was made subject to the following constraint:

$$\textit{verb-lxm}: \left[\text{SYN}\ \left[\text{HEAD}\ \left[\text{INF}\ /\ -\right]\right]\right]$$

We then posited the following type constraints:

subject-raising-verb-lxm (*srv-lxm*):

$$\begin{bmatrix} \text{ARG-ST} & \left\langle \boxed{1}\ , \begin{bmatrix}\text{SPR} & \langle\ \boxed{1}\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{INDEX} & s\end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix}\text{RESTR} & \left\langle \begin{bmatrix}\text{ARG} & s\end{bmatrix}\right\rangle\end{bmatrix}\end{bmatrix}$$

subject-control-verb-lxm (*scv-lxm*):

$$\begin{bmatrix} \text{ARG-ST} & \left\langle \text{NP}_i\ , \begin{bmatrix}\text{SPR} & \langle\ \text{NP}_i\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{INDEX} & s\end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix}\text{RESTR} & \left\langle \begin{bmatrix}\text{ARG} & s\end{bmatrix}\right\rangle\end{bmatrix}\end{bmatrix}$$

object-raising-verb-lxm (*orv-lxm*):

$$\begin{bmatrix}\text{ARG-ST} & \left\langle \text{NP}\ ,\ \boxed{1}\ ,\ \begin{bmatrix}\text{SPR} & \langle\ \boxed{1}\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{INDEX} & s\end{bmatrix}\right\rangle \\ \text{SEM} & \begin{bmatrix}\text{RESTR} & \left\langle \begin{bmatrix}\text{ARG} & s\end{bmatrix}\right\rangle\end{bmatrix}\end{bmatrix}$$

object-control-verb-lxm (*ocv-lxm*):

$$\begin{bmatrix}\text{ARG-ST} & \left\langle \text{NP}\ ,\ \text{NP}_i\ ,\ \begin{bmatrix}\text{SPR} & \langle\ \text{NP}_i\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{INDEX} & s\end{bmatrix}\right\rangle \\ \text{SEM} & \begin{bmatrix}\text{RESTR} & \left\langle \begin{bmatrix}\text{ARG} & s\end{bmatrix}\right\rangle\end{bmatrix}\end{bmatrix}$$

We added the following entries to our lexicon:

$$\left\langle \text{to}\ ,\ \begin{bmatrix} \textit{auxv-lxm}^{9} & \\ \text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{INF} & + \\ \text{AUX} & + \\ \text{FORM} & \text{base}\end{bmatrix}\end{bmatrix} \\ \text{ARG-ST} & \left\langle \boxed{1}\ ,\ \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\textit{verb} & \\ \text{INF} & - \\ \text{FORM} & \text{base}\end{bmatrix} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \boxed{1}\ \rangle\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s\end{bmatrix}\end{bmatrix}\right\rangle \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s \\ \text{RESTR} & \langle\ \rangle\end{bmatrix}\end{bmatrix}\right\rangle$$

$$\left\langle \text{continue}\ ,\ \begin{bmatrix} \textit{srv-lxm} & \\ \text{ARG-ST} & \left\langle \text{X}\ ,\ \overset{\text{VP}}{\begin{bmatrix}\text{INF} & +\end{bmatrix}}\right\rangle \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix}\text{RELN} & \textbf{continue} \\ \text{SIT} & s\end{bmatrix}\right\rangle\end{bmatrix}\end{bmatrix}\right\rangle$$

[9] The type *auxv-lxm* is discussed in the next chapter.

$$\left\langle \text{try}\ ,\ \begin{bmatrix} \textit{scv-lxm} & \\ \text{ARG-ST} & \left\langle \text{NP}_i\ ,\ \begin{matrix}\text{VP}\\ \begin{bmatrix}\text{INF} & +\end{bmatrix}\end{matrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{try} \\ \text{SIT} & s \\ \text{TRIER} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

$$\left\langle \text{expect}\ ,\ \begin{bmatrix} \textit{orv-lxm} & \\ \text{ARG-ST} & \left\langle \text{NP}_j\ ,\ \text{X}\ ,\ \begin{matrix}\text{VP}\\ \begin{bmatrix}\text{INF} & +\end{bmatrix}\end{matrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{expect} \\ \text{SIT} & s \\ \text{EXPECTER} & j \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

$$\left\langle \text{persuade}\ ,\ \begin{bmatrix} \textit{ocv-lxm} & \\ \text{ARG-ST} & \left\langle \text{NP}_j\ ,\ \text{NP}_i\ ,\ \begin{matrix}\text{VP}\\ \begin{bmatrix}\text{INF} & +\end{bmatrix}\end{matrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{persuade} \\ \text{SIT} & s \\ \text{PERSUADER} & j \\ \text{PERSUADEE} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Finally, in Exercise 1 we modified the Imperative Rule so as to require that the daughter be [INF −], as well as [FORM base].

12.9 Further Reading

The raising/control distinction was first introduced into the generative literature (but with different terminology) by Chomsky (1965) and Rosenbaum (1967). Other discussions of these phenomena include Jackendoff 1972, Postal 1974, Bach 1979, Bresnan 1982a, Postal and Pullum 1988, and Sag and Pollard 1991. Some of the terms that you might find in this literature include ‘equi’ for ‘control’, ‘subject-subject raising’ for ‘subject raising’ and ‘object-subject raising’ for ‘object raising’.

12.10 Problems

Problem 1: Classifying Verbs

Classify the following verbs as raising or control:

- tend
- decide
- manage
- fail
- happen

Justify your classification by applying each of the following four tests to each verb. Show your work by providing relevant examples and indicating their grammaticality.

(i) Can the verb take a dummy *there* subject if and only if its complement selects for a dummy *there* subject?

(ii) Can the verb take a dummy *it* subject if and only if its complement selects for a dummy *it* subject?

(iii) Can the verb take an idiom chunk subject if and only if the rest of the idiom is in its complement?

(iv) Do pairs of sentences containing active and passive complements to the verb end up being paraphrases of each other?

Make sure to restrict your attention to cases of the form: NP V *to* VP. That is, ignore cases like *Kim manages a store*, *Alex failed physics*, and any other valence that doesn't resemble the *continue* vs. *try* pattern.

Problem 2: Classifying Adjectives

Classify the following adjectives as raising or control:

- anxious
- bound
- certain
- lucky

Justify your classification by providing each of the four types of data discussed in Problem 1 for each adjective.

Make sure to restrict your attention to cases of the form: NP *be* Adj *to* VP. That is, ignore cases like *Kim is anxious about the exam*, *Carrie is certain of the answer*, and any other valence that doesn't resemble the *likely* vs. *eager* pattern.

Problem 3: Lexical Entries for Adjectives

To accommodate raising and control adjectives in our grammar, we need types *subject-raising-adjective-lexeme* (*sra-lxm*) and *subject-control-adjective-lexeme* (*sca-lxm*).

A. What is the immediate supertype of these two types? How else (if at all) do they differ from *srv-lxm* and *scv-lxm*?

B. Provide lexical entries for *likely* and *eager*, making use of these new types.

[*Hint: Keep in mind as you do this problem that in sentences like (i),* be *is a raising verb that mediates the relationship between* likely *and its subject.*]

(i) Kim is likely to leave.

Problem 4: ***Expect*** **vs.** ***Persuade***

In Section 12.6, we sketched an analysis of the verbs *expect* and *persuade* without providing justification for the fundamental distinction between the two types of lexeme we have posited. The purpose of this problem is to have you construct the arguments that underlie the proposed distinction between *orv-lxm* and *ocv-lxm*.

Construct examples of each of the following four types which show a contrast between *expect* and *persuade*. Explain how the contrasts are accounted for by the differences in the types *orv-lxm* and *ocv-lxm* and/or the lexical entries for *expect* and *persuade*.[10]

(i) Examples with dummy *there*.

(ii) Examples with dummy *it*.

(iii) Examples with idiom chunks.

(iv) Examples of relevant pairs of sentences containing active and passive complements. Indicate whether they are or are not paraphrases of each other.

Problem 5: Raising/Control in Icelandic

In Section 12.4 we discussed a formal difference in our treatment of raising and control. In raising, the whole *synsem* of the first argument of the embedded verb is identified with some argument of the higher verb. In control, the two arguments are only coindexed. This problem investigates some data from Icelandic that help motivate this formal distinction.

As noted in Problem 7 of Chapter 4, Icelandic has verbs that assign idiosyncratic cases to their subjects. Thus we get contrasts like the following (where other case markings on the subjects are unacceptable):

(i) Hun er vinsael.
She.NOM is popular

(ii) Hana vantar peninga.
Her.ACC lacks money

(iii) Henni batanaði veikin.
Her.DAT recovered-from the-disease

[10] Again, make sure you ignore all irrelevant uses of these verbs, including cases of CP complements, e.g. *persuade NP that ...* or *expect that ...* and anything else not directly relevant (*I expect to go*, *I am expecting Kim*, *She is expecting*, and so forth).

In infinitival constructions, two patterns are observed (again, other case markings on the subjects are unacceptable):

(iv) Eg vonast til að vanta ekki peninga.
I.NOM hope for to lack not money

(v) Eg vonast til að batnað veikin
I.NOM hope for to recover-from the-disease

(vi) Hana virðist vanta peninga.
Her.ACC seems to-lack money

(vii) Henni virðist hafa batnað veikin.
Her.DAT seems to-have recovered-from the-disease

A. The verbs *vonast* and *virðist* differ in the case they require on their subjects. Describe the pattern for each verb.

B. Assume that our analysis of raising and control for English is broadly applicable to Icelandic. Which class do the data in (i)–(vii) suggest that *vonast* and *virðist* each belong to? Why?

C. One alternative analysis of control verbs would identify the whole *synsem* of the first argument of a control verb with the subject of the infinitival complement. Use the data in (i)–(vii) to construct an argument against this alternative analysis.

Problem 6: A Type for Existential ***Be***

The *be* that takes *there* (see (11) on page 336) as its subject wasn't given a true lexical type in Chapter 11, because no suitable type had been introduced. One of the types in this chapter will do, if we make some of its constraints defeasible.

A. Which of the types introduced in this chapter comes closest to being consistent with the constraints on *there*-taking *be*?

B. Rewrite that type indicating which constraints must be made defeasible.

C. Give a stream-lined lexical entry for the *there*-taking *be* which stipulates only those constraints which are truly idiosyncratic to the lexeme.

Problem 7: There, There...

Problem 1 of Chapter 11 asked you to investigate verb agreement in sentences with *there* as the subject. There is actually considerable variation on this point, but the normative or prescriptively correct pattern is that finite forms of *be* that take *there* as their subject agree in number with the NP following *be*:

(i) There was/*were a riot in the park.

(ii) There were/*was many people at the party.

One way to formalize this is to have the lexical entry for the existential *be* lexeme stipulate that the NUM value on *there* is the same as the NUM value on the second element of the ARG-ST list. This entry would then undergo the normal inflectional lexical rules. Note that this analysis requires *there* to have an underspecified value for the feature NUM.

A. Give a lexical entry for the lexeme *be* that is consistent with the analysis described above.
B. Explain how your lexical entry interacts with the rest of the grammar to account for the contrast between (i) and (ii). Be sure to make reference to the role of lexical rules, grammar rules, and principles, as appropriate.
C. Does this analysis correctly predict the grammaticality of (iii) and the ungrammaticality of (iv)? Why or why not?

(iii) There continues to be a bug in my program.
(iv)*There continue to be a bug in my program.

Problem 8: Reflexives in Infinitival Complements
In Problem 4 above, you justified our analysis of *expect* and *persuade*.

A. Does that analysis (and in particular the ARG-ST values) interact with the Binding Theory of Chapter 7 to make the right predictions about the data in (i)–(viii)? Explain why or why not. Be sure to address all of the data given.

(i) We expect the doctor to examine us.
(ii)*We expect the doctor to examine ourselves.
(iii) We expect them to examine themselves.
(iv)*We expect them$_i$ to examine them$_i$.
(v) We persuaded the doctor to examine us.
(vi)*We persuaded the doctor to examine ourselves.
(vii) We persuaded them to examine themselves.
(viii)*We persuaded them$_i$ to examine them$_i$.

Now consider two more verbs: *appear* and *appeal*. *Appear* is a raising verb, and *appeal* is a control verb. They also differ as to which of their arguments is identified (or coindexed) with the subject of the lower clause.

B. Use the binding data in (ix)–(x) to decide which argument of *appear* is identified with the subject of *support*. Justify your answer.

(ix) They appeared to us to support themselves.
(x)*They$_i$ appeared to us to support them$_i$.

C. Use the binding data in (xi)–(xii) to decide which argument of *appeal* is coindexed with the subject of *support*. Justify your answer.

(xi)*They appealed to us to support themselves.
(xii) They$_i$ appealed to us to support them$_i$.

Problem 9: Extraposition and Raising
Our grammar as it currently stands gives three parses for sentences like (i), because the Extraposition Lexical Rule can apply to three different words in the sentence. This ambiguity is spurious, that is, it is not clear that there are really three different meanings for the sentence corresponding to the three parses.

(i) It seems to annoy Kim that dogs bark.

A. Which words could undergo the Extraposition Lexical Rule?
B. Draw the three structures (trees) that the grammar licenses for (i). You may use abbreviations like NP and S on all of the nodes.
C. Extra credit: Modify the Extraposition Lexical Rule to rule out the extra parses, or provide a reason that this can't easily be achieved.

Problem 10: Control and PP Complements

In Section 11.2 of Chapter 11, we noted that predicational prepositions must have ARG-ST lists with two elements in order to account for sentences like (i), where *be* is a raising verb:

(i) The fence is around the house.

If predicational prepositions like *around* have two arguments, we have to be careful what we say about sentences like (ii) and (iii):

(ii) The house$_j$ had a fence around it$_j$.
(iii)*The house$_j$ had a fence around itself$_j$.

In particular, if we don't say anything about the first argument of *around* in (iii) it could just happen to have the same index as the house (j), predicting that (iii) should be grammatical. Intuitively, however, the first argument of *around* should be the fence, and not the house.

A. Assuming the meaning of *around* involves a two-argument predication whose RELN is **around** and whose roles are ENCLOSED and COVER, write a lexical entry for *around* as it is used in (i).
B. Give the RESTR value that the grammar (including your lexical entry for *around*) should assign to the sentence in (i). (Recall that *the* is treated as a generalized quantifier, similar to *a*.)
C. Write a lexical entry for *have* as it is used in (ii) and (iii) which requires coindexing between the NP *a fence* and the first argument of *around*. [*Hints: This will be similar to lexical entries for object control verbs. However, since the ARG-ST of this* have *doesn't match the constraints on the type* ocv-lxm, *it can't be an instance of that type. Assume instead that it's an instance of* ptv-lxm. *Further assume that it selects for a predicational PP complement by specifying [MODE prop] on that argument. Finally, assume that the meaning of (ii) is 'the house has a fence, and the fence is around the house.' This makes it relatively easy to write the lexical entry for* have, *because you don't have to worry about how the predication introduced by the PP fits in: the Semantic Compositionality Principle will take care of that. What you need to attend to is the coindexing of elements in the lexical entry of* have.]
D. Explain how your lexical entry in part (C) interacts with the Binding Theory to correctly predict the judgments in (ii) and (iii).

13

Auxiliary Verbs

13.1 Introduction

In this chapter, we investigate the English auxiliary verb system. This is one of the most extensively analyzed (and frequently reanalyzed) empirical domains in the literature on generative syntax. Chomsky's transformational treatment of auxiliaries in *Syntactic Structures* was immensely influential; it galvanized the field around the framework of transformational grammar. In the intervening four decades, numerous alternative treatments have been advanced within a wide range of theoretical frameworks.

The auxiliary verb system is a particularly attractive domain for syntacticians because it involves a relatively small number of elements (basically, just a handful of verbs and the word *not*) which interact with each other in intricate and apparently complex ways. Moreover, though the English auxiliary system is quite language-specific (even closely related languages like Dutch and French have verbal systems that behave very differently), there are analogous elements in many other languages. Thus, this is a fertile domain for examining the interaction of universal grammatical principles with language-specific variation.

Cross-linguistically, the elements that are called 'auxiliaries' tend to share the following semantic and syntactic characteristics: (i) they express notions of time (past, present, future; continuation, completion), necessity, possibility, obligation, permission, negation, or questioning; and (ii) they occur in fixed positions, usually at or near the beginning of the sentence or at the end. English auxiliaries are a special kind of verb, including what are called 'modals' (*can, could, may, might, must, shall, should, will, would*), and uses of *be, do,* and *have* as 'helping verbs'.

Our analysis treats these auxiliaries as a special kind of subject raising verb, an idea originally proposed by J.R. Ross in the 1960s. We proceed from this basis to show how the special properties of auxiliaries with respect to such phenomena as negation and questioning can be handled in terms of syntactic features, and how the relevant generalizations about them can be expressed in terms of lexical rules.

13.2 The Basic Analysis

13.2.1 Some Facts about Auxiliaries

Consider the following data:

(1) Pat tap-danced.

(2) a. Pat can tap-dance.
b.*Pat tap-dance can.
c.*Pat can $\left\{\begin{array}{l}\text{tap-danced}\\ \text{tap-dancing}\end{array}\right\}$.
d. Pat is tap-dancing.
e.*Pat is $\left\{\begin{array}{l}\text{tap-dance}\\ \text{tap-danced}\end{array}\right\}$.
f. Pat has tap-danced.
g.*Pat has $\left\{\begin{array}{l}\text{tap-dance}\\ \text{tap-dancing}\end{array}\right\}$.

(3) a. Pat could have tap-danced.
b. Pat could be tap-dancing.
c. Pat has been tap-dancing.
d. Pat could have been tap-dancing.
e.*Pat has could tap-danced.
f.*Pat is having tap-danced.
g.*Pat could be having tap-danced.

(4) a.*Pat could will tap-dance.
b.*Pat has had tap-danced.
c.*Pat is being tap-dancing.

These examples illustrate the following generalizations about auxiliary verbs:[1]

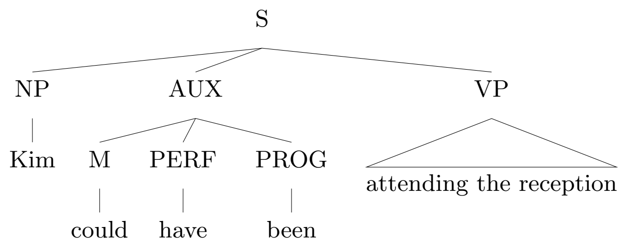
(5) a. Auxiliaries are optional.
b. All auxiliaries precede any (non-auxiliary) verb.
c. Auxiliaries determine the FORM of the following verb.
d. Auxiliaries can co-occur with each other, but only in a fixed order.
e. Auxiliaries (of any given type) cannot iterate.

[1] In some dialects of English, certain sequences of modals, such as in *Pat might could tap-dance* are possible, apparently violating (5d). However, even in those dialects, such sequences are restricted to certain combinations. This suggests that the variation should be handled in terms of differences in individual lexical entries (e.g. in their lexical types or in their possible FORM values), rather than through a wholly different analysis of modals as a class. Investigating this dialect variation is beyond the scope of this book.

We find in the literature two basic approaches to the analysis of English auxiliaries. The first, going back to Chomsky's original treatment, involves introducing a new (phrasal) category often called AUX which dominates all auxiliary verbs. AUX is introduced and expanded by rules like the following, giving rise to structures like (7):[2]

(6) a. S → NP AUX VP
    b. AUX → (M)(PERF)(PROG)

(7)



This approach has the attractive feature that it straightforwardly captures the optionality, the ordering, and the noniterability of auxiliaries – that is, the properties listed in (5a,b,d,e). On the other hand, it doesn't say anything about the FORM dependencies – so it needs augmentation, if it is to rule out the starred alternatives in (2).

The other type of analysis treats auxiliaries as verbs that take VP complements. This has the immediate advantage that it provides familiar tools for restricting the FORM value of the head of the following VP. Its potential disadvantage is that the mechanisms expressing the ordering and iteration constraints are somewhat less direct.

The second approach fits much better with the grammar we have been developing. This is because the AUX constituent doesn't seem to have a head, so it isn't clear how to fit it in with our phrase structure rules or with our way of handling co-occurrence restrictions such as the FORM dependencies between auxiliaries.[3] In what follows, we will pursue the VP complement analysis of auxiliary verbs. In fact, as will become clear below, this treatment can be incorporated into our grammar without assuming any new grammar rules – all the essential information will be in the lexical entries for auxiliary verbs.

---

[2]'M' in (6) stands for 'modal', 'PERF' for 'perfect(ive)', and 'PROG' for 'progressive'. The latter two are fairly standard terms for the uses of *have* and *be* under discussion.

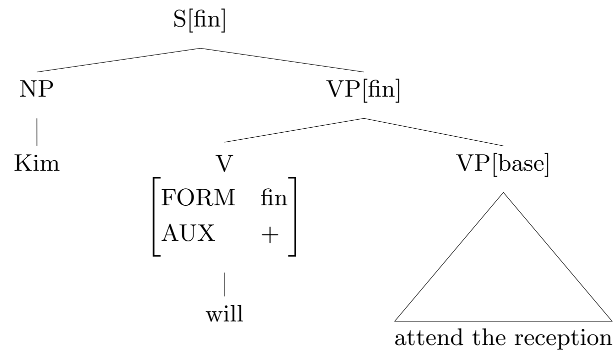
[3]In addition, there are empirical objections to be made against this analysis. It denies verb status to the auxiliaries *have* and *be*, yet the auxiliary uses of these elements are inflectionally indistinguishable from the verbal uses:

(i) Kim is happy. (cf. Kim is talking.)
(ii) I am in trouble. (cf. I am talking.)
(iii) Pat has troubles. (cf. Pat has troubled them.)

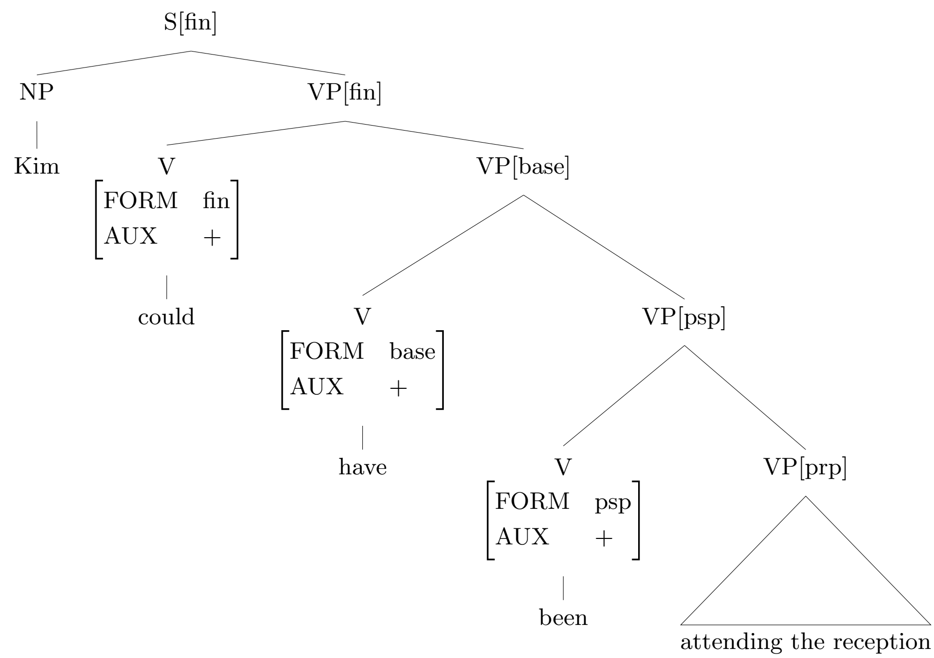
In addition, the AUX analysis has trouble dealing with ellipsis of the sort discussed below. See footnote 18.

Our analysis assigns structures like the following to sentences containing auxiliary verbs:

(8)



(9)



13.2.2 Lexical Entries for Auxiliary Verbs

We have already seen the lexical entries for two auxiliary verbs, namely, the infinitival verb *to* and the verb *be*. We will use the latter as a model for the other entries for auxiliary verbs and then return to the issues we have raised about the ordering and iteration of auxiliary verbs.

The lexical entry for *be* that we originally used for the passive was generalized in Chapter 11 to cover other uses of *be*, including the progressive – that is, forms immediately preceding verbs suffixed with *-ing*. The entry we gave for *be* (which was intended to cover its co-occurrence with both passive and progressive VP complements) was the following:

(10)
$$\left\langle \text{be}\ , \begin{bmatrix} \textit{be-lxm} & \\ \text{ARG-ST} & \left\langle \boxed{1}\ , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{PRED} & + \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Recall that the specification [PRED +] distinguishes predicative phrases (certain APs, PPs, and NPs, as well as passive and present participial VPs) from other phrases. (See Section 11.2 of Chapter 11.)

Note that (10) clearly bears a crucial resemblance to the type *srv-lxm*, which we introduced in Chapter 12, in that the first element of the ARG-ST is identical to the specifier of the second element of the ARG-ST. Instead of listing such shared properties in the lexical entries of *be* and the other auxiliary verbs, we can let them inherit the constraints from *srv-lxm*. Since all auxiliaries have certain other properties in common, we will posit a subtype of *srv-lxm*, called *auxiliary-verb-lexeme* (*auxv-lxm*). The feature structures in the lexical entries for auxiliaries will all be of this type. As all lexical entries must specify a maximal (i.e. leaf) type for their feature structure, we also posit a contrasting type *infinitival-complement-subject-raising-verb-lexeme* (*ic-srv-lxm*) for non-auxiliary subject-raising verbs. Such verbs all share the property of selecting an infinitival VP as their second argument, so *ic-srv-lxm* will be constrained appropriately. Recall that *srv-lxm* is a subtype of *verb-lxm*, thus the resulting hierarchy looks like (11):

(11)
$$\begin{tikzcd} & \textit{verb-lxm} \arrow[d, dash] & \\ & \textit{srv-lxm} \arrow[dl, dash] \arrow[dr, dash] & \\ \textit{ic-srv-lxm} & & \textit{auxv-lxm} \end{tikzcd}$$

These types are subject to the constraints shown in (12):

(12)

TYPE	FEATURES/CONSTRAINTS	IST
verb-lxm	$\begin{bmatrix} \text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{PRED} & - \\ \text{INF} & / - \\ \text{AUX} & / - \end{bmatrix}\end{bmatrix} \\ \text{ARG-ST} & \left\langle \begin{bmatrix}\text{HEAD} & \textit{nominal} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix}\end{bmatrix}, \ldots \right\rangle \\ \text{SEM} & \begin{bmatrix}\text{MODE} & \text{prop}\end{bmatrix}\end{bmatrix}$	*infl-lxm*
srv-lxm	$\begin{bmatrix} \text{ARG-ST} & \left\langle \boxed{1}, \begin{bmatrix}\text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix} \right\rangle \end{bmatrix}$	*verb-lxm*
ic-srv-lxm	$\begin{bmatrix} \text{ARG-ST} & \left\langle \text{X}, \overset{\text{VP}}{\begin{bmatrix}\text{INF} & + \\ \text{INDEX} & s\end{bmatrix}} \right\rangle \\ \text{SEM} & \begin{bmatrix}\text{RESTR} & \left\langle \begin{bmatrix}\text{ARG} & s\end{bmatrix} \right\rangle\end{bmatrix} \end{bmatrix}$	*srv-lxm*
auxv-lxm	$\begin{bmatrix} \text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{AUX} & +\end{bmatrix}\end{bmatrix}\end{bmatrix}$	*srv-lxm*

Note that the constraints stated on *srv-lxm* in the last chapter have been split up – the subject-sharing constraint now applies to all subject-raising verbs (including auxiliaries), while the constraint requiring the semantic embedding of the second argument applies only to instances of type *ic-srv-lxm*, i.e. only to non-auxiliary subject-raising verbs.

Finally, the feature AUX will be used to distinguish auxiliary verbs from all others in the analyses given below. Accordingly, the constraint [AUX +] on *auxv-lxm* interacts with the defeasible constraint [AUX / −] on *verb-lxm* to ensure that auxiliary verbs are specified as [AUX +] and all other verbs are [AUX −]. Since AUX is a feature of the part of speech type *verb*, non-verbal lexeme types need not mention it at all.

Once constraints are set up in this way, the type *be-lxm* (introduced in Chapter 11) can be eliminated in favor of a simpler analysis where *be* is assigned to the type *auxv-lxm*. This allows us to simplify the lexical entry of *be*, relying on inheritance to express further properties that *be* shares with other lexemes. Thus the lexical entry specified in (13) is enriched to (14) through constraint inheritance.

(13)

$$\left\langle \text{be}\ , \begin{bmatrix} \textit{auxv-lxm} & \\ \text{ARG-ST} & \left\langle \text{X}\ , \begin{bmatrix} \text{SYN} & \begin{bmatrix}\text{HEAD} & [\text{PRED} \quad +]\end{bmatrix} \\ \text{SEM} & [\text{INDEX} \quad s\ ] \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(14)

$$\left\langle \text{be}\ , \begin{bmatrix} \textit{auxv-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AUX} & + \\ \text{AGR} & \boxed{0} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ [\text{AGR} \quad \boxed{0}]\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \boxed{3}\ , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & [\text{PRED} \quad +] \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \boxed{3}\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & [\text{INDEX} \quad s\ ] \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

And the entry for auxiliary *have* will look like this:

(15)

$$\left\langle \text{have}\ , \begin{bmatrix} \textit{auxv-lxm} & \\ \text{ARG-ST} & \left\langle \text{X}\ , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{psp} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & [\text{INDEX} \quad s_2] \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s_1 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{have} \\ \text{SIT} & s_1 \\ \text{ARG} & s_2 \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

This entry differs from (13) in two essential ways. First, the second argument (i.e. the complement) must be a VP specified as [FORM psp]. This guarantees that the VP complement of *have* will be headed by a past participle. Second, the semantics of auxiliary *have*, unlike that of *be*, is not vacuous. The form of the complement and the meaning are, in fact, what distinguish *have* from other auxiliary verbs in English. Again, no constraint needs to be stated in (15) to identify the first element of the ARG-ST list with the VP complement's SPR value, as this information is inherited from the supertype *srv-lxm*.

Turning to the modals, we observe that they have a peculiarity, illustrated by the following examples:

(16) a.*Pat hopes to can study syntax.
b.*Sandy has musted study syntax.
c.*Sandy is musting study syntax.

What these examples show is that modals can never appear as base-form words or as past or present participles. In fact, the only contexts in which modals can occur are ones where we normally expect finite verbs. Notice, by the way, that this restriction appears not to be semantically based, since appropriate forms of *be able to* (which is virtually synonymous with *can*) are fine in these contexts.

It is straightforward to deal with this restriction on modals. We may assume that each modal's lexical entry is specified so as to be inconsistent with [FORM base], [FORM psp], or [FORM prp]. In fact, modals must be finite; so our entry for a modal could look like (17):

(17)
$$\left\langle \text{can}\ , \begin{bmatrix} \textit{auxv-lxm} & \\ \text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{FORM} & \text{fin}\end{bmatrix}\end{bmatrix} \\ \text{ARG-ST} & \left\langle \text{X}\ , \begin{bmatrix} \text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\textit{verb} & \\ \text{INF} & - \\ \text{FORM} & \text{base}\end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s_2\end{bmatrix}\end{bmatrix}\right\rangle \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s_1 \\ \text{RESTR} & \left\langle \begin{bmatrix}\text{RELN} & \textbf{can}\\ \text{SIT} & s_1 \\ \text{ARG} & s_2\end{bmatrix}\right\rangle\end{bmatrix}\end{bmatrix}\right\rangle$$

Notice that unlike most lexical entries of type *verb*, (17) specifies a FORM value. The FORM values that we use for verbs (i.e. 'fin', 'psp', 'prp', 'base', and 'pass') are, in most cases, introduced by lexical rules like the Past-Tense Verb Lexical Rule or the Present Participle Lexical Rule. In all such rules, the FORM values of the INPUT and OUTPUT are identified: in inflectional rules, by an inviolable constraint on *i-rule* identifying the SYN values, and in the relevant derivational rules by the defeasible identity constraint on SYN values.[4] Thus, the lexical stipulation that *can* is [FORM fin] means that the lexical sequences satisfying that lexical entry cannot undergo any lexical rule introducing a FORM value other than fin. For example, the Present Participle Lexical Rule will only accept inputs that are compatible with the specification [FORM prp]. Hence, by specifying the lexical entries of modals as [FORM fin], we ensure that they cannot undergo this lexical rule.

[4]None of the relevant derivational rules override that defeasible identity constraint with respect to the FORM value.

It should also be mentioned that affixes don't get added to the modals. We can accommodate this fact by making certain assumptions about the morphological function F_{3SG} that is used by the 3rd-Singular Verb Lexical Rule that we introduced in Chapter 8. In particular, we can assume that it includes the following in its definition:

(18)

X	$F_{3SG}(X)$
will	will
shall	shall
might	might
....	...
(otherwise)	X-s

The modal lexemes thus may undergo the 3rd-Singular Verb Lexical Rule, but the relevant outputs will be forms like *will* and *shall*, rather than **wills* and **shalls*. The Non-3rd-Singular Verb Lexical Rule works fine as it is, as it introduces no affixation. Finally, to prevent past-tense modal forms, we can simply assume that the function F_{PAST}, introduced by the Past-Tense Verb Lexical Rule, is undefined for *will*, *shall*, and the other modals.[5]

Finally, notice that on our analysis, auxiliaries are treated semantically like raising verbs. That is, they assign no semantic role to their subjects; rather, the subject of an auxiliary verb gets its semantic role – and various syntactic properties – from the complement of the auxiliary. Given this, our analysis makes predictions regarding the possibility of auxiliaries occurring in sentences with dummy subjects or idiom chunk subjects. Specifically, we correctly predict that sentences like the following are grammatical:

(19) a. There might be a unicorn in the garden.
b. There has been a unicorn in the garden.
c. It will annoy people that dogs are barking.
d. It is annoying people that dogs are barking.
e. Close tabs have been kept on Pat by the FBI.
f. Close tabs are being kept on Pat by the FBI.

We also predict that there is no difference in truth conditions between *A doctor must have been examining Pat* and *Pat must have been being examined by a doctor*. This also seems correct.

Exercise 1: The Ambiguity of Modals

Most modals are semantically ambiguous between a sense relating to necessity or possibility and one relating to obligation or permission. This is perhaps clearest with *may*, in examples like *Pat may join the club*, but *can*, *could*, *should*, *must*, and *might* exhibit analogous ambiguities. The obligation/permission interpretations (often called 'deontic'), arguably express two-place relations between an individual and what that individual is

[5] We should mention that, historically, *would* evolved from the past tense form of *will*, *should* from the past tense form of *shall*, etc. We are assuming that such pairs of forms are unrelated in the grammar of Modern English.

obligated or permitted to do. For that reason, one might think that modals in their deontic uses should be treated as control verbs, rather than raising verbs. What do the syntactic tests for the raising/control distinction indicate about the status of deontic modals?

13.2.3 Co-Occurrence Constraints on Auxiliaries

Some of the facts we noticed earlier about the ordering among auxiliaries and the constraints on their iteration fall out naturally from our lexical entries. Others require some work.

• The fact that modals must come first in any string of auxiliaries follows from the fact that they must be finite. Since the complements to the auxiliaries *have* and *be* must have some FORM specification other than fin (namely, 'psp', 'prp', or 'pass'), a modal can never be the head daughter of the VP complement of *have* or *be*.

• The fact that modals don't iterate also follows from their obligatory finiteness, together with the requirement that the head of the VP complement to a modal must be specified as [FORM base].

• The fact that perfective *have* (that is, forms of *have* immediately preceding past participles) can't follow progressive *be* can be seen as a manifestation of a wider generalization. Not all verbs can appear in the progressive. Specifically, verbs (often referred to as 'stative' verbs) whose semantics involves a state rather than an action or an activity, generally sound bad in the progressive:

(20) a.*Pat is owning a house.
 b.*Chris is knowing the answer.

This is pretty clearly a semantic restriction: making something progressive turns an action or activity into a state, namely, the state of that action or activity being in progress; for stative verbs such as *own* or *know*, it doesn't make sense to talk about the state being in progress. The perfective *have* is also used to describe a state, namely, the state of completion for the event of its complement VP. (*Kim has left* intuitively conveys that the event of Kim's leaving is complete.) Since stative verbs don't have progressives, the perfective *have* doesn't either. Hence (21) is unacceptable for the same reason as (20):

(21)*Pat is having slept.

We will not attempt to formalize this restriction in our lexical entry, since it depends on dealing with aspects of semantics that go beyond the scope of this text.

• The same semantic restriction accounts for the failure of progressive *be* to iterate, since it, too, denotes a state, and hence can't be the complement to another occurrence of progressive *be*, ruling out (22):

(22)*Chris is being sleeping.

Note that since a progressive can never be the semantic argument of another progressive, other such examples, e.g. (23) are also correctly ruled out:

(23) a.*Chris continued being sleeping.
 b.*Chris kept being sleeping.

In short, it would be incorrect to formulate this problem in terms of why *being* can't head the VP complement of *be*. The issue involves a semantic problem of far greater scope.

• Finally, the failure of perfective *have* to iterate (24) cannot be handled in just the same way as the last two restrictions, since stative verbs can occur in the perfective.

(24)*Stevie has had traveled to China.

We could require that perfective *have* be an exception to the Past-Participle Lexical Rule. This would entail that it doesn't license any past participle lexical sequences so it couldn't appear as the head of the VP complement to another occurrence of perfective *have*. Alternatively, we could try to find a semantic explanation, for example, that iterating perfective *have* would be redundant. We will not work out the details of the latter analysis here, nor choose between these two approaches.

13.3 The NICE Properties

English auxiliary verb lexemes differ from other verbal lexemes in (at least) four ways that are often described as follows:

(25) a. NEGATION: Their finite forms can be immediately followed by *not* as a way of negating the sentence.[6]
b. INVERSION: Their finite forms can precede the subject in questions.
c. CONTRACTION: They have contracted finite forms that include the suffix *n't*.
d. ELLIPSIS: Their complements can be omitted when the meaning of the missing complement can be reconstructed from the surrounding linguistic context.

These are sometimes called the 'NICE' properties of auxiliaries. They are illustrated in the following examples:[7]

(26) a. Pat should not leave.
b.*Pat raked not leaves.

(27) a. Has Pat left town?
b.*Left Pat town?

(28) a. They haven't cut the price.
b.*They cutn't the price.
c.*They halven't the price.

(29) a. If anybody is spoiling the children, Pat is.
b.*If anybody keeps spoiling the children, Pat keeps.

Our analysis of these differences will be purely lexical in nature. It will make use of a handful of features (including AUX) and a small set of lexical rules.

[6]Below, we present a general account of 'polarized' adverbs that includes sentential negation as a special case.

[7]In earlier stages of the English language, nonauxiliary verbs also exhibited certain of the NICE properties. In fact, this earlier state of affairs still survives today in certain fixed expressions, e.g. *How goes it?* or old sayings, such as *Waste not, want not*.

13.4 Auxiliary *Do*

Notice that the negative (both contracted and uncontracted), interrogative, or elliptical counterparts to sentences with no auxiliary verb are usually expressed with the verb *do*:

(30) a. Pat raked leaves.
b. Pat did not rake leaves.

(31) a. Pat left town.
b. Did Pat leave town?

(32) a. They halved the price.
b. They didn't halve the price.

(33) If anybody keeps spoiling the children, Pat does.

So let's add *do* to our list of auxiliary verbs. In order to do this, we need to examine its properties a bit more carefully. For example, what is the FORM of the head of its complement? Are there any other restrictions on the kind of complements it can take? What kind of forms (participles, base forms, etc.) does *do* give rise to?

The following examples illustrate properties of *do* that will have to be incorporated into its analysis:

(34) a. Pat does not eat garlic.
b. $\text{*Pat does not} \left\{ \begin{array}{l} \text{eats} \\ \text{eating} \\ \text{eaten} \\ \text{to eat} \end{array} \right\} \text{garlic.}$

(35) a. Pat tried to take logic.
b.*Pat tried to don't take logic.
c.*Pat has done not take logic.
d.*Pat is doing not take logic.

(36) a. Does Pat like watching television?
b.*Does Pat be watching television?
c.*Does Pat have watched television?

These examples show:

- The lexical head daughter of the VP complement of *do* must be [FORM base] and [INF −].
- Auxiliary *do* itself only occurs in finite forms.[8]
- The lexical head daughter of the VP complement of *do* cannot be an auxiliary verb.

That is, *do* behaves just like the modals, but with the added conditions that (i) its VP complement cannot be headed by an auxiliary verb and (ii) it makes no semantic contribution. This can be encoded straightforwardly into the lexical entry for auxiliary *do*:[9]

[8]The (non-finite) *do* that is found in imperatives like *Do sit down!* is not the same *do*, as evidenced by the fact that it can co-occur with *be*, as in *Do be careful!* (cf **Do they be careful?*).

[9]Since the only [INF +] verb in our grammar (*to*) is also [AUX +], all [AUX −] VPs will necessarily be [INF −] as well. Therefore, we only need to specify [AUX −] on the complement of *do*.

(37)

$$\left\langle \text{do}\ , \begin{bmatrix} \textit{auxv-lxm} & \\ \text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{FORM} & \text{fin}\end{bmatrix}\end{bmatrix} \\ \text{ARG-ST} & \left\langle \text{X}\ , \begin{bmatrix} \text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{base} \\ \text{AUX} & - \end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s\end{bmatrix}\end{bmatrix}\right\rangle \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s \\ \text{RESTR} & \langle\ \rangle\end{bmatrix}\end{bmatrix}\right\rangle$$

The semantics specified in (37) correctly guarantees that *do* makes no contribution to the semantics; and the [FORM fin] specification ensures that the only words derived from this root are finite forms. Of course, those words have nonempty RESTR values in their semantics, because the lexical rules that generate them add temporal restrictions.

The analysis of *do* as an *auxv-lxm* (and hence as a *srv-lxm*) also predicts that it will allow dummies and idiom chunks as subjects, given the right complements. It is actually tricky to construct such sentences, since the verbs that license dummies or idiom chunk subjects are very often forms of *be*, and hence are [AUX +]. Since *do* doesn't allow [AUX +] complements, this requires an extra step in the construction of the relevant examples. The trick is to put a nonauxiliary raising verb in between, yielding sentences like (38):

(38) a. There did not continue to [be riots in the park].
b. Does it continue to [annoy people that dogs bark]?
c. Close tabs don't seem to have been [kept on anybody].
d. There DOES seem to [be an error in this proof].

Exercise 2: Understanding How Subject Selection Propagates
Sketch the tree for (38d). Show the SPR value of each VP constituent, explaining how the SPR value of the bracketed VP gets passed up to be the SPR value of the highest VP.

13.5 Analyzing the NICE Properties

In this section, we work out analyses within our theory for each of the NICE properties. This will involve one lexical rule per property, plus the addition of a couple of new features.

13.5.1 Negation and Reaffirmation

The word *not* can appear in a wide variety of contexts, but in this discussion, we will restrict our attention to sentence negation – that is, to cases in which it is the whole clause that is interpreted as being negated. Hence, we will not be dealing with CONSTITUENT NEGATION, that is, with uses of *not* like those in (39):[10]

(39) a. Not many arrows hit the target.
b. I try not to make trouble.
c. Pat must have not been listening.

We also exclude from our analysis the second occurrence of *not* in examples like (40), since this is negating only the VP, not the whole sentence.

(40) Kleptomaniacs can not NOT steal.

If both occurrences of *not* in (40) were instances of sentence negation, they would cancel each other out, and it would mean 'kleptomaniacs can steal'. While (40) clearly has this as an entailment, it actually asserts something stronger, namely, that kleptomaniacs MUST steal. This is the interpretation we get if we assume that the first *not* negates the whole sentence but the second one negates only *steal*. The sentence in (40) shows that there can be only one SENTENTIAL *not* per clause.

Notice that finite auxiliaries also combine with the focussed adverbs SO and TOO (capitalized here to indicate the presence of a pitch accent), as illustrated in (41):

(41) a. Sandy did SO write that.
b. Leslie can TOO lift that.

These adverbs are the positive counterparts of sentential *not*. While sentential *not* can be used to deny a proposition, SO and TOO are used to reaffirm a proposition whose truth has been called into question in a given dialogue. And these adverbs, like sentential *not*, cannot iterate:

(42) a.*Sandy did SO SO write that.
b.*Sandy did SO TOO write that.
c.*Leslie can TOO TOO lift that.
d.*Leslie can TOO SO lift that.

Since SO, TOO and *not* intuitively all involve 'polarized' meanings (reaffirming or denying), we will refer to them as polarized adverbs, abbreviated as ADV_{pol}. Note that the pattern illustrated in (42) is actually more general: only one ADV_{pol} can appear per clause:

(43) a.*Sandy did NOT SO write that.
b.*Sandy did NOT TOO write that.

Examples like (44) might appear to be counterexamples to this claim, but in fact here the negation is constituent negation, as the meaning makes clear:

(44) a.?Leslie did SO not go to the party.
b. Leslie can TOO not be intimidated.

[10] In a nutshell, our analysis of these is to treat *not* as a modifier that attaches to the left of phrases (other than finite verb phrases). However, working this analysis out in detail would require some additional machinery that would take us too far afield, so we will not pursue it here.

In addition to the above observations, we want to account for data like the following, which show that SO, TOO and sentential *not* must appear immediately following a finite auxiliary verb.

(45) a. Pat will not leave.
b. Pat will SO/TOO leave.

(46) a.*Pat not will leave.
b.*Pat SO/TOO will leave.

(47) a. Pat has not left.
b.*Pat not left.

(48) a. Pat has SO/TOO left.
b.*Pat SO/TOO left.

(49) a. Pat would not have left.
b.*Pat not has left.
c. Pat must not have been leaving.

(50) a. Pat would SO/TOO have left.
b.*Pat SO/TOO has left.
c. Pat must SO/TOO have been leaving.

In order to deal with the fact that polar adverbs don't iterate and the fact that they appear right after the auxiliary verb, we will treat ADV_{pol} as a complement selected by a finite auxiliary verb (rather than as, say, a modifier).[11]

To flesh this analysis out, we first introduce a feature POLARIZED (POL). The lexical rule we propose will add an ADV_{pol} to the ARG-ST of a [POL −] finite auxiliary word, requiring that the output word be specified as [POL +]. This will guarantee that the lexical rule can never apply to its own output, thereby preventing any clause from having more than one ADV_{pol}.[12]

[11] This might seem unintuitive at first, but in fact there are other verbs in English that take an adverb as complement:

(i) This book reads easily.
(ii) This book reads.
(iii) The management has treated Sandy contemptuously.
(iv) The management has treated Sandy.

While (ii) and (iv) are grammatical, they don't involve the same senses of *read* and *treat* as (i) and (iii). And indeed, (ii) is pragmatically odd under normal circumstances, while (i) is not. This shows that the adverbs are obligatory for the senses of *read* and *treat* in (i) and (iii). Since they're obligatory, they must be complements.

[12] POL is declared as appropriate for the part of speech type *verb*. By imposing the constraint in (i) on the type *verb-lxm*:

(i) $$\textit{verb-lxm}: \left[\mathrm{SYN}\ \left[\mathrm{HEAD}\ \left[\mathrm{POL}\ -\right]\right]\right],$$

we can ensure that the only verbs that are [POL +] are those that have undergone a lexical rule that makes them so.

(51) ADV_{pol}-Addition Lexical Rule

$$\begin{bmatrix} \textit{pi-rule} & \\ \text{INPUT} & \left\langle \text{X}\ , \begin{bmatrix} \text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\textit{verb} & \\ \text{FORM} & \text{fin}\\ \text{POL} & - \\ \text{AUX} & +\end{bmatrix}\end{bmatrix} \\ \text{ARG-ST} & \langle \boxed{1} \rangle \oplus \boxed{\text{A}} \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s_1\end{bmatrix}\end{bmatrix}\right\rangle \\ \text{OUTPUT} & \left\langle \text{Y}\ , \begin{bmatrix} \text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{POL} & +\end{bmatrix}\\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle \text{Z}\rangle\end{bmatrix}\end{bmatrix} \\ \text{ARG-ST} & \langle \boxed{1}\rangle \oplus \left\langle \overset{\text{ADV}_{pol}}{\begin{bmatrix}\text{INDEX} & s_2 \\ \text{RESTR} & \left\langle\begin{bmatrix}\text{ARG} & s_1\end{bmatrix}\right\rangle\end{bmatrix}}\right\rangle \oplus \boxed{\text{A}} \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s_2\end{bmatrix}\end{bmatrix}\right\rangle \end{bmatrix}$$

By factoring the ARG-ST list as '$\langle\boxed{1}\rangle \oplus \boxed{\text{A}}$', (51) specifies that the ADV_{pol} is inserted as the second member of the output's ARG-ST list.

This is not the first lexical rule we have seen that creates new words from words that have already been formed via application of inflectional rules. The type *post-inflectional-rule* was introduced in Chapter 11 to deal with the phenomenon of extraposition. *Pi-rule*s are subject to a defeasible constraint identifying the HEAD value of INPUT and OUTPUT, as well as the defeasible constraint on all *l-rule*s identifying the SEM value of INPUT and OUTPUT. In addition, both the INPUT and OUTPUT are of type *word*, and thus subject to the ARP. Hence this lexical rule imposes all the constraints shown in (52):[13]

[13] (52) doesn't show the effect of the ARP on the input, because, as far as the lexical rule is concerned, it could be satisfied in numerous ways. That is, the lexical rule doesn't constrain the length of either the specifier or complements lists of the input. On the other hand, it does specify that the output must be [SPR ⟨ Z ⟩]. This is to ensure that the ARP has the desired effect.

(52) ADV$_{pol}$-Addition Lexical Rule (with inherited constraints)

$$
\begin{bmatrix}
\textit{pi-rule} & \\
\text{INPUT} & \left\langle \boxed{0}\,,
\begin{bmatrix}
\textit{word} & \\
\text{SYN} & \begin{bmatrix}
\text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{fin} \\ \text{POL} & - \\ \text{AUX} & + \end{bmatrix} \\
\text{VAL} & \begin{bmatrix} \text{MOD} & \boxed{B} \end{bmatrix}
\end{bmatrix} \\
\text{ARG-ST} & \langle \boxed{1} \rangle \oplus \boxed{A} \\
\text{SEM} & \begin{bmatrix} \text{INDEX} & s_1 \\ \text{MODE} & \boxed{2} \\ \text{RESTR} & \boxed{C} \end{bmatrix}
\end{bmatrix}
\right\rangle \\
\text{OUTPUT} & \left\langle \boxed{0}\,,
\begin{bmatrix}
\textit{word} & \\
\text{SYN} & \begin{bmatrix}
\text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{fin} \\ \text{POL} & + \\ \text{AUX} & + \end{bmatrix} \\
\text{VAL} & \begin{bmatrix} \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle \boxed{3} \rangle \oplus \boxed{A} \\ \text{MOD} & \boxed{B} \end{bmatrix}
\end{bmatrix} \\
\text{ARG-ST} & \langle \boxed{1} \rangle \oplus \left\langle \overset{\boxed{3}\text{ADV}_{pol}}{\begin{bmatrix} \text{INDEX} & s_2 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{ARG} & s_1 \end{bmatrix} \right\rangle \end{bmatrix}} \right\rangle \oplus \boxed{A} \\
\text{SEM} & \begin{bmatrix} \text{INDEX} & s_2 \\ \text{MODE} & \boxed{2} \\ \text{RESTR} & \boxed{C} \end{bmatrix}
\end{bmatrix}
\right\rangle
\end{bmatrix}
$$

Here it is important to see that the INPUT and OUTPUT specify conflicting values for POL and INDEX.[14] As a result, the inherited defeasible identity constraints 'push down' to identify the values of all other features within HEAD and SEM whose values are not specified as incompatible.

[14]Whenever a single description mentions distinct indices (e.g. s_1 and s_2), the intention is that any feature structure satisfying that description will contain distinct indices in the relevant positions.

ADV_{pol} words[15] have lexical entries like the one shown in (53) for *not*:

(53)

$$\left\langle \text{not}\ ,\ \overset{ADV_{pol}}{\left[\begin{array}{ll}\text{SEM} & \left[\begin{array}{ll}\text{INDEX} & s_1 \\ \text{RESTR} & \left\langle \left[\begin{array}{ll}\text{RELN} & \textbf{not} \\ \text{SIT} & s_1 \\ \text{ARG} & s_2\end{array}\right] \right\rangle\end{array}\right]\end{array}\right]} \right\rangle$$

The rule in (51) has the effect of identifying the INDEX value of the auxiliary with the ARG value of the ADV_{pol}'s relation. The result of (51), then, is to allow lexical sequences like (54):

(54)

$$\left\langle \text{is}\ ,\ \left[\begin{array}{ll}\textit{word} & \\ \text{SYN} & \left[\begin{array}{ll}\text{HEAD} & \left[\begin{array}{ll}\textit{verb} & \\ \text{FORM} & \text{fin} \\ \text{POL} & + \\ \text{AUX} & +\end{array}\right] \\ \text{VAL} & \left[\begin{array}{ll}\text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle \boxed{2}\ ,\ \boxed{5} \rangle\end{array}\right]\end{array}\right] \\ \text{ARG-ST} & \left\langle \boxed{1}\ ,\ \overset{\boxed{2}ADV_{pol}}{\left[\begin{array}{ll}\text{INDEX} & s_3 \\ \text{RESTR} & \left\langle \left[\text{ARG}\ \ s_1\right] \right\rangle\end{array}\right]}\ ,\ \boxed{5}\left[\begin{array}{ll}\text{PRED} & + \\ \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{INDEX} & s_1\end{array}\right] \right\rangle \\ \text{SEM} & \left[\text{INDEX}\ \ s_3\right]\end{array}\right] \right\rangle$$

This word selects as complements both an ADV_{pol} like (53) and a predicative phrase. When this happens, the index of the ADV_{pol} (s_3 in (54)) is the INDEX of the verb itself, and the index of the rule input (s_1 in (52)) is the argument of the 'not' relation. Since *be* identifies its INDEX value with that of its predicative complement, this produces a negated semantics for a VP like *is not happy*, as shown in (55):

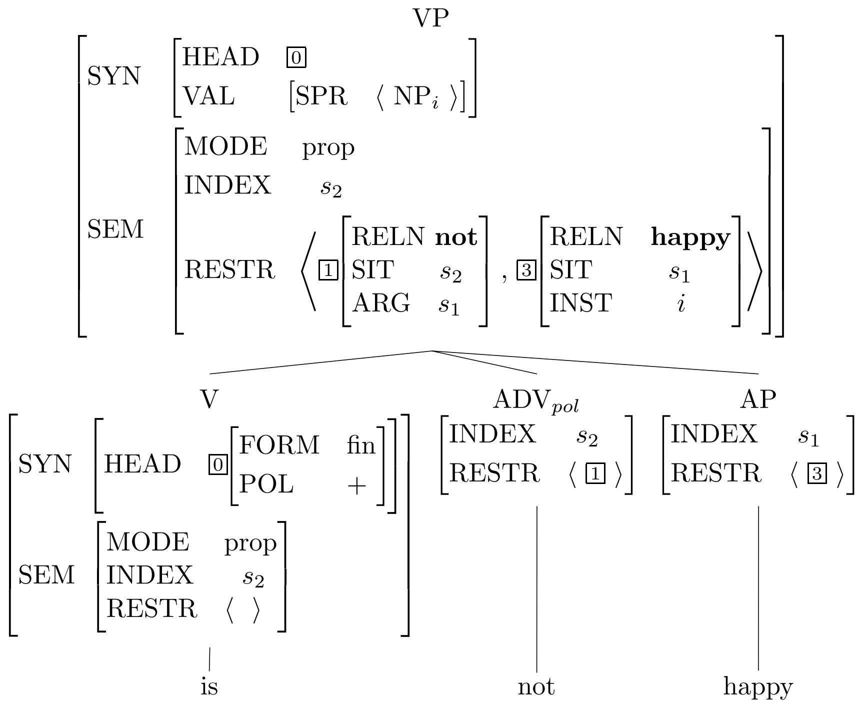
[15] ADV_{pol} is an abbreviation that could be defined in a number of ways. For present purposes it will suffice to define it as in (i):

(i)

$$ADV_{pol} = \left[\begin{array}{ll}\text{SYN} & \left[\text{HEAD}\ \ \textit{adv}\right] \\ \text{SEM} & \left[\text{RESTR}\ \ \langle\ [\text{RELN}\ \textbf{not} \mid \textbf{reaffirm}]\ \rangle\right]\end{array}\right],$$

where **reaffirm** is the predication we will assume for the reaffirming adverbs TOO and SO.

(55)



If this VP combines with a subject by the Head-Specifier Rule, then the INDEX value of this VP will also be passed up to be the INDEX of the S (in keeping with the Semantic Inheritance Principle). In this way, a sentence like *Kim is not happy* will have negation as its 'highest' predication, which is the correct semantic result. A similar analysis is provided for sentences that contain SO or TOO instead of *not*, but in this case the highest predication will be one of reaffirmation, not negation.

13.5.2 Inversion

In questions, a finite auxiliary verb precedes the subject. All the same restrictions that apply to auxiliaries in declaratives apply when they occur in this inverted order. Thus we have:

(56) a. Can Pat tap-dance?

b. *Can Pat $\left\{\begin{array}{l}\text{tap-danced}\\ \text{tap-dancing}\end{array}\right\}$?

c. Is Pat tap-dancing?

d. *Is Pat $\left\{\begin{array}{l}\text{tap-dance}\\ \text{tap-danced}\end{array}\right\}$?

e. Has Pat tap-danced?

f. *Has Pat $\left\{\begin{array}{l}\text{tap-dance}\\ \text{tap-dancing}\end{array}\right\}$?

g. Could Pat be tap-dancing?

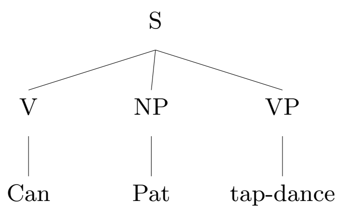
h. *Is Pat coulding tap-dance?

i. Could Pat have tap-danced?

j. *Has Pat could tap-dance?
k. Has Pat been tap-dancing?
l. *Is Pat having tap-danced?
m. Could Pat have been tap-dancing?
n. *Could Pat be having tap-danced?
o. *Could Pat will tap-dance?
p. *Has Pat had tap-danced?
q. *Is Pat being tap-dancing?

These data suggest that we want to maintain the relationship between the auxiliary verb and its complement even when the auxiliary precedes the subject. The immediate problem then is how to get the subject to appear on the right of the (finite) auxiliary instead of on the left. A simple approach is to treat the postauxiliary NP not as a specifier, but rather as the auxiliary verb's first complement. Then, because the Head-Complement Rule generates complements in the same order in which they are listed in the head's COMPS list, the subject will immediately follow the auxiliary verb. Hence, once the auxiliary's first ARG-ST member is associated with the first member of the COMPS list (rather than with the auxiliary's specifier), our grammar will put it right where we want it. This is illustrated in the tree in (57):

(57)



This effect can be easily achieved by the lexical rule in (58):

(58) Inversion Lexical Rule

$$\begin{bmatrix} \textit{pi-rule} & \\ \text{INPUT} & \left\langle \text{X}\ ,\ \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{fin} \\ \text{AUX} & + \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \boxed{\text{A}} \\ \text{SEM} & [\text{MODE}\ \ \text{prop}] \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \text{Y}\ ,\ \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & [\text{INV}\ \ +] \\ \text{VAL} & [\text{SPR}\ \ \langle\ \rangle] \end{bmatrix} \\ \text{ARG-ST} & \boxed{\text{A}} \\ \text{SEM} & [\text{MODE}\ \ \text{ques}] \end{bmatrix} \right\rangle \end{bmatrix}$$

The constraints inherited by (58) are shown in (59):[16]

[16]Once again, we have not shown the effect of the ARP on the INPUT, as it could be satisfied in many ways. See note 13.

(59) Inversion Lexical Rule (with inherited constraints)

$$
\left[\begin{array}{ll}
\textit{pi-rule} & \\
\text{INPUT} & \left\langle \boxed{0}\,, \left[\begin{array}{ll}
\textit{word} & \\
\text{SYN} & \left[\begin{array}{ll}
\text{HEAD} & \left[\begin{array}{ll} \textit{verb} & \\ \text{FORM} & \text{fin} \\ \text{INV} & + \\ \text{AUX} & + \end{array}\right] \\
\text{VAL} & \left[\begin{array}{ll} \text{MOD} & \boxed{B} \end{array}\right]
\end{array}\right] \\
\text{ARG-ST} & \boxed{A} \\
\text{SEM} & \left[\begin{array}{ll} \text{MODE} & \text{prop} \\ \text{INDEX} & s \\ \text{RESTR} & \boxed{C} \end{array}\right]
\end{array}\right] \right\rangle \\
\text{OUTPUT} & \left\langle \boxed{0}\,, \left[\begin{array}{ll}
\textit{word} & \\
\text{SYN} & \left[\begin{array}{ll}
\text{HEAD} & \left[\begin{array}{ll} \textit{verb} & \\ \text{FORM} & \text{fin} \\ \text{INV} & + \\ \text{AUX} & + \end{array}\right] \\
\text{VAL} & \left[\begin{array}{ll} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \boxed{A} \\ \text{MOD} & \boxed{B} \end{array}\right]
\end{array}\right] \\
\text{ARG-ST} & \boxed{A} \\
\text{SEM} & \left[\begin{array}{ll} \text{MODE} & \text{ques} \\ \text{INDEX} & s \\ \text{RESTR} & \boxed{C} \end{array}\right]
\end{array}\right] \right\rangle
\end{array}\right]
$$

This rule has two main effects: it ensures that the SPR list of the output is empty and that the output's semantic mode is 'ques(tion)', rather than 'prop'.[17]

The outputs of this rule are finite auxiliary verbs that take no specifier. Hence, all members of their ARG-ST list will also appear on the COMPS list. One such auxiliary word is shown in (60):

[17] Here we make the simplifying assumption that inverted sentences are always associated with interrogative semantics. In fact, inverted clauses appear in constructions, e.g. the following:

(i) Never have I heard such a beautiful rendition!
(ii) Had we known that beforehand, we would never have participated.
(iii) May your teeth fall out on your wedding night!

(60)

$$\left\langle \text{will}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{fin} \\ \text{AUX} & + \\ \text{INV} & + \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle \boxed{1}\ ,\ \boxed{2} \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \boxed{1}\text{NP[nom]}\ ,\ \boxed{2}\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{base} \end{bmatrix} \\ \text{SPR} & \langle \boxed{1} \rangle \end{bmatrix} \right\rangle \end{bmatrix} \right\rangle$$

And because such words have the first ARG-ST member on the COMPS list, that element will be realized as a complement, to the right of the lexical head in a phrase structure constructed in accordance with the Head-Complement Rule.

Recall that the lexical rules for finite forms presented in Chapter 8 constrain the CASE value of the first member of the ARG-ST list. Because the first element of the ARG-ST list is the first complement, all words formed from the outputs of this lexical rule will specify the appropriate CASE constraints not on the SPR value, but rather on the first COMPS member, as shown in (60). The lexical rules for finite verbal forms interact with the Inversion Lexical Rule in (59) to predict contrasts like the following:

(61) a. Can she tap-dance?
b. *Can her tap-dance?
c. Are we winning?
d. *Are us winning?

Similarly, for those auxiliaries that bear an inflection showing agreement with their subjects (namely, finite forms of *be*, *have*, and *do*), the agreement in the inverted words (i.e. the outputs of the Inversion Lexical Rule) should be with the first member of the COMPS list. Through a complex interaction of constraints, our grammar captures this fact and correctly predicts data like (62):

(62) a. Is the dog barking?
b. $*\left\{\begin{array}{l}\text{Am}\\ \text{Are}\end{array}\right\}$ the dog barking?
c. Have you finished the assignment?
d. *Has you finished the assignment?

The lexemes *be*, *have* and *do* are subject to the SHAC, which requires agreement between the verbs and their specifiers. In contrast, the outputs of the Inversion Lexical Rule are *word*s, and as such do not inherit this constraint. They are, however, related by the Inversion Lexical Rule to other lexical sequences (the inputs of the rule). Those inputs are words (subject to the ARP) which are related by other lexical rules to the lexemes which are subject to the SHAC. Because the ARP applies to these intermediate words, the outputs of the Inversion Lexical Rule will be constrained to have the same AGR value as their first ARG-ST element, i.e. their first complement.

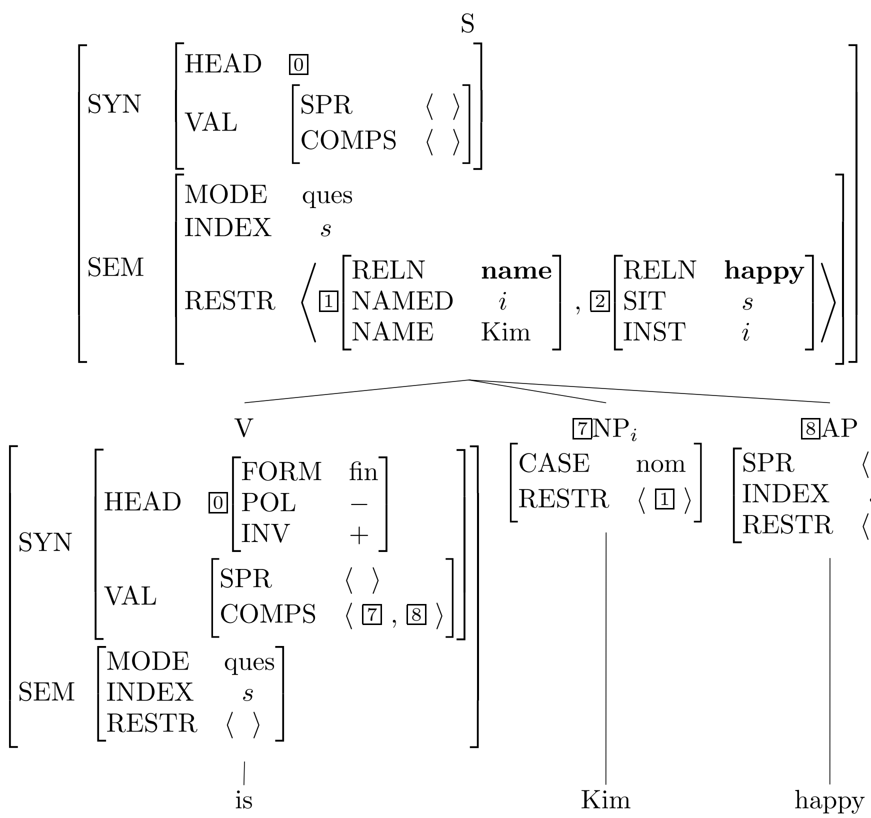
Exercise 3: SHAC and Inversion
Check whether this is so, sketching instantiations of the 3rd-Singular Verb Lexical Rule (with the lexeme *be* as input) and the Inversion Lexical Rule (with the output of the 3rd-Singular Verb Lexical Rule instantiation as input).

Moreover, as we have seen, auxiliaries are raising verbs, and as such, their first argument is identical to the element on the SPR list of their VP argument. Hence, in a word structure licensed by an inverted auxiliary verb, the first complement will also function as the second complement's subject. We therefore predict data like the following:

(63) a. Will there be children in the audience?
b.*Will there win the game?
c. Has it annoyed people that dogs bark?
d. Are tabs kept on all linguists by the FBI?
e.*Are tabs taken on all linguists by the FBI?
f. Was advantage taken of the opportunity by students?
g.*Was advantage kept of the opportunity by students?
h. Did it continue to annoy people that nobody listened?
i.*Did it try to annoy people that nobody listened?

In short, the Inversion Lexical Rule creates lexical sequences whose first complement has all the properties of a subject, except that it comes after the verb. This gives us structures like (64):

(64)



Finally, we must address the matter of the feature INV, which we have introduced without justification. Most auxiliary verbs are unspecified for INV, so that they can undergo inversion. However, there are finite auxiliary verbs that cannot appear in inverted structures, for example the auxiliary *better* illustrated in (65):

(65) a. They better (not) go downtown.
b.*Better they (not) go downtown?

Once we have the feature INV, this contrast can be accounted for by specifying that the lexical entry for *better* is (exceptionally) specified as [INV −].

There is also a subtle semantic contrast between pairs like the following:

(66) a. I shall go downtown.
b. Shall I go downtown?

In uttering (66a), one asserts a simple future-time proposition. Someone who asks (66b), on the other hand, is asking whether they should go downtown. That is, the simple future-time question corresponding to the proposition of (66a) can't be posed by asking (66b). Rather, to get that message across (perhaps asking a question to a fortune teller), one would need to ask something like (67a) or (67b):

(67) a. Will I go downtown?
b. Is it true that I shall go downtown?

To account for these contrasts, one could posit two distinct lexical entries for (the lexeme) *shall*. The first would be assigned a simple futurate meaning and would be specified as [INV −]. The second *shall* would include a modal predication in its RESTR and would be specified as [INV +]. In this way, the feature INV would play a key role in accounting for the semantic contrast in (66). In Chapter 15, we'll see another case where an auxiliary verb is specified as [INV −] to bar it from undergoing the Inversion Lexical Rule.

13.5.3 Contraction

Auxiliary verbs with the contracted negative suffix *-n't* exhibit a number of lexical idiosyncrasies. For example, we say *won't*, not **willn't*. Others are *don't, mustn't*, and *shan't* – though many speakers don't have any form of *shall* with *-n't* suffixed. (Don't be fooled by spelling: *don't* and *mustn't* are exceptional because of their pronunciation, which is not what would result from simply appending the sound 'nt' to *do* and *must*). There are also exceptions to the rule: **amn't* and, for most speakers, **mayn't*.

We will treat the morphological aspects of contraction in the same way we treated the morphological effects of the inflectional lexical rules we presented earlier, which also allowed for idiosyncratic morphological exceptions. That is, we will posit a morphological function, F_{NEG}, that relates inflected forms of auxiliaries to their negative contracted forms. The rule is then fairly straightforward: it applies to finite forms of auxiliaries, changing their morphology and semantics. The only other modification needed is to add the specification [POL −] to the input and [POL +] to the output. These specifications keep the rule from applying to its own output, and hence automatically blocks forms like **can'tn't*.

The rule, then, is as follows:

(68) Contraction Lexical Rule

$$
\left[\begin{array}{ll}
\textit{pi-rule} & \\
\text{INPUT} & \left\langle \boxed{2}\ ,\ \left[\begin{array}{ll}
\text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll} \textit{verb} & \\ \text{FORM} & \text{fin} \\ \text{AUX} & + \\ \text{POL} & - \end{array}\right] \end{array}\right] \\
\text{ARG-ST} & \boxed{\text{B}} \\
\text{SEM} & \left[\begin{array}{ll} \text{INDEX} & s_1 \\ \text{RESTR} & \boxed{\text{A}} \end{array}\right]
\end{array}\right] \right\rangle \\
\text{OUTPUT} & \left\langle \text{F}_{NEG}(\boxed{2})\ ,\ \left[\begin{array}{ll}
\text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll} \text{POL} & + \end{array}\right] \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \langle\ \text{X}\ \rangle \end{array}\right] \end{array}\right] \\
\text{ARG-ST} & \boxed{\text{B}} \\
\text{SEM} & \left[\begin{array}{ll} \text{INDEX} & s_2 \\ \text{RESTR} & \left\langle \left[\begin{array}{ll} \text{RELN} & \textbf{not} \\ \text{SIT} & s_2 \\ \text{ARG} & s_1 \end{array}\right] \right\rangle \oplus \boxed{\text{A}} \end{array}\right]
\end{array}\right] \right\rangle
\end{array}\right]
$$

This rule maps lexical sequences like (69) into counterparts like (70):

(69)

$$
\left\langle \text{is}\ ,\ \left[\begin{array}{ll}
\textit{word} & \\
\text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll} \textit{verb} & \\ \text{FORM} & \text{fin} \\ \text{POL} & - \\ \text{AUX} & + \end{array}\right] \end{array}\right] \\
\text{ARG-ST} & \left\langle \boxed{2}\ ,\ \left[\begin{array}{ll} \text{PRED} & + \\ \text{SPR} & \langle\ \boxed{2}\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{INDEX} & s_1 \end{array}\right] \right\rangle \\
\text{SEM} & \left[\begin{array}{ll} \text{INDEX} & s_1 \\ \text{RESTR} & \langle\ \rangle \end{array}\right]
\end{array}\right] \right\rangle
$$

(70)
$$\left\langle \text{isn't}\ , \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\textit{verb} & \\ \text{FORM} & \text{fin}\\ \text{POL} & +\\ \text{AUX} & +\end{bmatrix}\end{bmatrix}\\ \text{ARG-ST} & \left\langle \boxed{2}\ , \begin{bmatrix}\text{PRED} & +\\ \text{SPR} & \langle \boxed{2} \rangle \\ \text{COMPS} & \langle\ \rangle\\ \text{INDEX} & s_1\end{bmatrix}\right\rangle \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s_2\\ \text{RESTR} & \left\langle \begin{bmatrix}\text{RELN} & \textbf{not}\\ \text{SIT} & s_2\\ \text{ARG} & s_1\end{bmatrix}\right\rangle\end{bmatrix}\end{bmatrix}\right\rangle$$

Words like (70) allow examples like (71) to be analyzed with the correct semantics:

(71) a. Kim isn't a doctor.
b. That Kim is smart isn't obvious.
c. We aren't on your schedule.
d. Sandy wasn't investigated/investigating.

13.5.4 Ellipsis

Ellipsis is a discourse phenomenon, in the sense that the interpretation of the missing constituent sometimes depends on something said in an earlier sentence – possibly even by another speaker, as in the following dialogue:

(72) Speaker A: I haven't been reading the newspapers.
Speaker B: Well, I have.

Because a comprehensive treatment would take us too far afield, we will not try to provide an analysis of the semantic relation between an elliptical sentence and its antecedent. Instead, we will concentrate on the syntactic constraints on ellipsis.[18]

[18] The example in (72) also illustrates an argument for the phrase structure we assign to auxiliary constructions (with nested VPs), rather than the alternative with a node AUX containing all auxiliary verbs illustrated in (7) on page 393. On our analysis, what is missing from speaker B's utterance (namely, *been reading the newspapers*) forms a constituent (a VP). But under the AUX alternative, what is missing is part of the AUX, plus the following VP. In examples like (i), we see that all three of the possible ellipses correspond to a constituent in our analysis, but only one of them would be a constituent in the AUX analysis.

(i) Pat couldn't have been doing the reading, but Chris could (have (been (doing the reading)))

As noted earlier, ellipsis is only possible with auxiliary verbs. Unlike the other properties of auxiliary verbs, ellipsis is not limited to finite auxiliaries, as shown in (73):

(73) I haven't been following the debate on taxes, but I should have.

In this respect, ellipsis differs from the previous three NICE properties we looked at, in which the auxiliary is always finite. Hence the lexical rule we formulate for ellipsis will not have to stipulate [FORM fin], though it will still require [AUX +].

Note in addition that the class of auxiliary verbs allowing ellipsis of their complement includes the infinitival element *to*:

(74) a. We asked them to open the window, and they tried to.
b. We hoped that the wine would improve with age, but it didn't seem to.

As noted in Chapter 12, this is part of the motivation for treating *to* as a special kind ([INF +]) of auxiliary verb.

The following formulation of the rule captures our syntactic observations:

(75) Ellipsis Lexical Rule

$$\begin{bmatrix} \textit{d-rule} & \\ \text{INPUT} & \left\langle \boxed{1}\, , \begin{bmatrix} \textit{auxv-lxm} \\ \text{ARG-ST} \ \langle \boxed{2} \rangle \oplus \boxed{A} \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \boxed{1}\, , \begin{bmatrix} \textit{dervv-lxm} \\ \text{ARG-ST} \ \langle \boxed{2} \rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

The outputs of this rule are lexemes whose ARG-ST no longer contains any of its arguments except the first one (the auxiliary verb's subject argument). We have created a new type – *derived-verb-lexeme* (*dervv-lxm*) – to accommodate the outputs of this rule.

Exercise 4: *dervv-lxm*
Why do we need the type *dervv-lxm*? Why can't the outputs of the Ellipsis Lexical Rule be *auxv-lxm*?

The outputs of this rule are lexical sequences like (76):

(76)

$$\left\langle \text{will}\ ,\ \begin{bmatrix} \textit{dervv-lxm} \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{fin} \\ \text{AUX} & + \\ \text{POL} & - \\ \text{AGR} & \boxed{1} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ [\text{AGR}\ \boxed{1}]\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \langle\ \text{NP}\ \rangle \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s_1 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{will} \\ \text{SIT} & s_1 \\ \text{ARG} & s_2 \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

And these will give rise to lexical sequences like the following, where the ARP ensures that the COMPS list is empty:

(77)

$$\left\langle \text{will}\ ,\ \begin{bmatrix} \textit{word} \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{fin} \\ \text{AUX} & + \\ \text{POL} & - \\ \text{AGR} & \boxed{1} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \boxed{7}[\text{AGR}\ \boxed{1}]\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \langle\ \boxed{7}\text{NP}\ \rangle \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s_1 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{will} \\ \text{SIT} & s_1 \\ \text{ARG} & s_2 \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

The rule formulated in (75) says nothing about the discourse properties of ellipsis. In particular, it does not specify that ellipsis is only possible in contexts where there is an antecedent phrase that provides the interpretation of the missing complement. It does ensure that the semantics of its outputs is incomplete: In sentences built with word structures licensed by lexical sequences like (77), s_2 – the argument of the **will** predication – will not be associated with any predication. Hence the meaning of any clause built from (77) is inherently incomplete, and must be supplemented by appropriate material from the surrounding linguistic context. How this happens – the interpretation of elliptical

VPs – is a rich topic with a literature of its own that we cannot do justice to here.

However, this analysis nonetheless permits us to account for the fact that elliptical sentences behave in some ways as if they contained a constituent that is not in fact present. For example, nonreferential subjects are possible in elliptical sentences, so long as the missing complements are understood properly:

(78) a. Would there be any point in asking for seconds?
There would.
b. If any advantage could be taken of the situation, maximal advantage was.

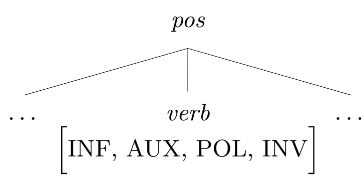
Since nothing is said in (76) about the INDEX or FORM of the NP on the SPR list, that feature structure will be compatible with nonreferential NPs, thus allowing examples like (78a,b).

13.6 Summary

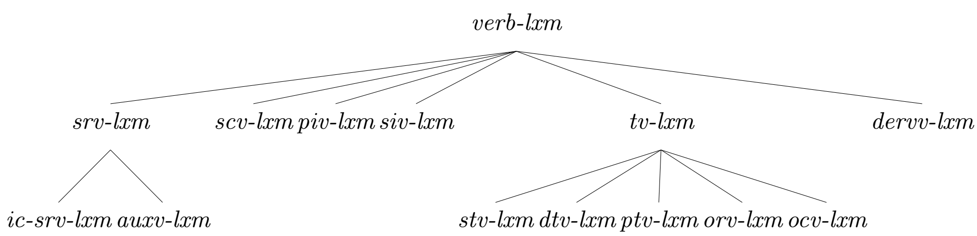
English auxiliaries comprise a richly structured system with numerous idiosyncratic properties. Our theory handles the complexities of the system straightforwardly. Auxiliary verbs are raising verbs that impose special conditions on their second argument. Auxiliary verbs exhibit some exceptional behaviors, such as the restriction of modals to [FORM fin]. And there are several lexical rules that are restricted to auxiliaries. These create specialized versions of the auxiliary verbs to account for the NICE properties – negation, inversion, contraction and ellipsis, which we have analyzed in considerable detail. In Chapter 15, we examine some of the dialect variation in the behavior of English auxiliaries and discuss how our analysis can be modified to handle it.

13.7 Changes to the Grammar

In this chapter, we added three new features: AUX(ILIARY), POL(ARIZED), and INV(ERTED). These features, along with INF, discussed in Chapter 12, are appropriate only for the type *verb*:



We revised the type hierarchy under *verb-lxm*, introducing the two new lexeme types: *auxv-lxm* and *dervv-lxm*. The revised hierarchy and related constraints look like this:



TYPE	FEATURES/CONSTRAINTS	IST
verb-lxm	$\left[\begin{array}{ll} \text{SYN} & \left[\text{HEAD}\ \left[\begin{array}{ll} \textit{verb} & \\ \text{PRED} & - \\ \text{INF} & / - \\ \text{AUX} & / - \\ \text{POL} & - \end{array}\right]\right] \\ \text{ARG-ST} & \left\langle \left[\begin{array}{ll} \text{HEAD} & \textit{nominal} \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{array}\right] \end{array}\right], \ldots \right\rangle \\ \text{SEM} & \left[\text{MODE}\ \ \text{prop}\right] \end{array}\right]$	*infl-lxm*
srv-lxm	$\left[\text{ARG-ST}\ \left\langle \boxed{1}, \left[\begin{array}{ll} \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle\ \rangle \end{array}\right] \right\rangle\right]$	*verb-lxm*
ic-srv-lxm	$\left[\begin{array}{ll} \text{ARG-ST} & \left\langle \text{X}, \overset{\text{VP}}{\left[\begin{array}{ll} \text{INF} & + \\ \text{INDEX} & s \end{array}\right]} \right\rangle \\ \text{SEM} & \left[\text{RESTR}\ \left\langle \left[\text{ARG}\ \ s\right] \right\rangle\right] \end{array}\right]$	*srv-lxm*
auxv-lxm	$\left[\text{SYN}\ \left[\text{HEAD}\ \left[\text{AUX}\ \ +\right]\right]\right]$	*srv-lxm*
dervv-lxm		*verb-lxm*

We also added a number of lexical entries for auxiliary verbs:

$$\left\langle \text{be}, \left[\begin{array}{ll} \textit{auxv-lxm} & \\ \text{ARG-ST} & \left\langle \text{X}, \left[\begin{array}{ll} \text{SYN} & \left[\text{HEAD}\ \left[\text{PRED}\ \ +\right]\right] \\ \text{SEM} & \left[\text{INDEX}\ \ s\right] \end{array}\right] \right\rangle \\ \text{SEM} & \left[\begin{array}{ll} \text{INDEX} & s \\ \text{RESTR} & \langle\ \rangle \end{array}\right] \end{array}\right] \right\rangle$$

$$\left\langle \text{have}\ ,\ \begin{bmatrix} \textit{auxv-lxm} & \\ \text{ARG-ST} & \left\langle \text{X}\ ,\ \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{psp} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s_2 \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s_1 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{have} \\ \text{SIT} & s_1 \\ \text{ARG} & s_2 \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

$$\left\langle \text{can}\ ,\ \begin{bmatrix} \textit{auxv-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{fin} \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \text{X}\ ,\ \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{INF} & - \\ \text{FORM} & \text{base} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s_2 \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s_1 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{can} \\ \text{SIT} & s_1 \\ \text{ARG} & s_2 \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

$$\left\langle \text{do}\ ,\ \begin{bmatrix} \textit{auxv-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{fin} \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \text{X}\ ,\ \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{base} \\ \text{AUX} & - \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

To deal with the NICE properties (and reaffirmative adverbs in addition), we added four lexical rules:

ADV$_{pol}$-Addition Lexical Rule

$$
\begin{bmatrix}
\textit{pi-rule} & \\
\text{INPUT} & \left\langle \text{X}\ ,\ \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{fin} \\ \text{POL} & - \\ \text{AUX} & + \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \langle \boxed{1} \rangle \oplus \boxed{\text{A}} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s_1 \end{bmatrix} \end{bmatrix} \right\rangle \\
\text{OUTPUT} & \left\langle \text{Y}\ ,\ \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{POL} & + \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR}\ \langle\ \text{Z}\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \langle \boxed{1} \rangle \oplus \left\langle \overset{\text{ADV}_{pol}}{\begin{bmatrix} \text{INDEX} & s_2 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{ARG} & s_1 \end{bmatrix} \right\rangle \end{bmatrix}} \right\rangle \oplus \boxed{\text{A}} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s_2 \end{bmatrix} \end{bmatrix} \right\rangle
\end{bmatrix}
$$

Inversion Lexical Rule

$$
\begin{bmatrix}
\textit{pi-rule} & \\
\text{INPUT} & \left\langle \text{X}\ ,\ \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{fin} \\ \text{AUX} & + \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \boxed{\text{A}} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \end{bmatrix} \end{bmatrix} \right\rangle \\
\text{OUTPUT} & \left\langle \text{Y}\ ,\ \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{INV} & + \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \boxed{\text{A}} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ques} \end{bmatrix} \end{bmatrix} \right\rangle
\end{bmatrix}
$$

Contraction Lexical Rule

$$\begin{bmatrix} \textit{pi-rule} & \\ \text{INPUT} & \left\langle \boxed{2}\, , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{fin} \\ \text{AUX} & + \\ \text{POL} & - \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \boxed{\text{B}} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s_1 \\ \text{RESTR} & \boxed{\text{A}} \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \text{F}_{NEG}(\boxed{2})\, , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{POL} & + \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \text{ X } \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \boxed{\text{B}} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s_2 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{not} \\ \text{SIT} & s_2 \\ \text{ARG} & s_1 \end{bmatrix} \right\rangle \oplus \boxed{\text{A}} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

Ellipsis Lexical Rule

$$\begin{bmatrix} \textit{d-rule} & \\ \text{INPUT} & \left\langle \boxed{1}\, , \begin{bmatrix} \textit{auxv-lxm} & \\ \text{ARG-ST} & \langle \boxed{2} \rangle \oplus \boxed{\text{A}} \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \boxed{1}\, , \begin{bmatrix} \textit{dervv-lxm} & \\ \text{ARG-ST} & \langle \boxed{2} \rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

13.8 Further Reading

As noted above, Chomsky 1957 is the seminal work on the English auxiliary system; Lasnik 1995 is an elementary (and updated) presentation of Chomsky's analysis, further elaborated in Lasnik et al. 2000. Among the most influential subsequent investigations are Ross 1969, McCawley 1971, Akmajian et al. 1979, Steele 1981, Gazdar et al. 1982, and Pollock 1989. The standard reference on negative contraction is Zwicky and Pullum 1983. The analysis presented here is based in part on Warner 1993, Warner 2000, Kim 2000, Kim and Sag 2002, and Sag to appear.

13.9 Problems

Problem 1: Modals and Imperatives

Our grammar does not license imperative sentences headed by modals, like (i). Explain how it rules them out.

(i)*Can solve all the problems! (cf. Be able to solve all the problems!)

Problem 2: *Do* Support

As noted in Section 13.4, the negated, inverted, contracted, and elliptical counterparts of sentences without main verbs all involve forms of the auxiliary *do*. This phenomenon has been called '*do* support' and has been conceptualized in transformational frameworks in terms of a special rule that 'inserts' *do* under certain conditions. The analysis of the English auxiliary system in this chapter accounts for the ungrammaticality of the a examples in (i)–(v) and the grammaticality of the b-sentences in (i)–(v) in a purely declarative fashion, with no recourse to a special rule of *do* support:

(i) a.*Dana put not the ice cream in the freezer.
b. Dana did not put the ice cream in the freezer.

(ii) a.*Dana put TOO the ice cream in the freezer.
b. Dana did TOO put the ice cream in the freezer.

(iii) a.*Put Dana the ice cream in the freezer?
b. Did Dana put the ice cream in the freezer?

(iv) a.*Dana putn't the ice cream in the freezer.
b. Dana didn't put the ice cream in the freezer.

(v) a.*No one else wanted to put the ice cream in the freezer, so Dana put.
b. No one else wanted to put the ice cream in the freezer, so Dana did.

For each of the pairs of sentences (i)–(v), explain how our rules and lexical entries account for the indicated contrast in grammaticality. In particular, explain why the grammar does not allow the a-sentence and describe how it licenses the b-sentence. Your explanations should include reference to lexical rules. Drawing trees for these examples is optional.

Problem 3: Inverted Trees

Draw trees for the following sentences. (You need not include all features on the nodes, but you should give a complete feature specification for the nodes dominating *Did* and *Is*. Otherwise, you can just concentrate on getting the tree geometry right, using standard abbreviations for node labels and using tags where appropriate.)

(i) Did Pat put the ice cream in the freezer?

(ii) Is there a monster in Loch Ness?

Problem 4: Negation and Inversion
Our ADV$_{pol}$-Addition and Inversion Lexical Rules can interact to allow us to generate negative questions.

A. Which of the following sentences will be licensed by our rules so far? [*Hint: Figure out what the ARG-ST value of has would have to be in each case, and then decide which of those is a possible ARG-ST given our rules and the lexical entry for have.*]
   (i) Has Pat not been sleeping?
   (ii) Has not Pat been sleeping?
B. Are these predictions of the grammar correct, according to your intuitions about the language?

In parts (C)-(E), be explicit about the effect of the lexical rules on the ARG-ST of *has* and on how this interacts with the grammar rules.

C. How does the grammar rule out the sentence it rules out?
D. How does the grammar license the sentence it licenses?
E. Does the grammar license sentence (iii)? Why or why not?
   (iii) Hasn't Pat been sleeping?

Problem 5: Contraction and Double Negation
Our formulation of the ADV$_{pol}$-Addition Lexical Rule and the Contraction Lexical Rule guarantees that neither rule can apply to the output of the other.

A. Explain why this is so.
B. Does the mutual exclusivity of these two lexical rules make the right empirical predictions? Provide data and an explanation to support your answer.

Problem 6: The Interaction of Ellipsis with Negation and Inversion
Sentential *not* can occur in elliptical clauses:

(i) We wanted to taste the salad, but we could not.
(ii) They were asked to help the cook, but they did not.
(iii) You thought you were clearing the table, but you were not.
(iv) We thought that they had arrived, but they had not.

In addition, inversion is possible in elliptical clauses, as in the following sentences:

(v) Kim said they will become famous. Will they?
(vi) You thought you were helping them out. Were you?

A. Does our grammar predict that examples like (i)–(iv) are well-formed? If it does, describe how these sentences are licensed. If not, explain how they are (incorrectly) ruled out.
B. Does our grammar predict that examples like (v)–(vi) are well-formed? If it does, describe how these sentences are licensed. If not, explain how they are (incorrectly) ruled out.

14

Long-Distance Dependencies

14.1 Introduction

One of the principal tasks of a theory of grammar is to provide mechanisms that allow economical formulations of the sorts of co-occurrence restrictions that exist in natural languages. In earlier chapters, we developed techniques for analyzing such aspects of syntax as differences in the valence of particular verbs, agreement between subject and verb, agreement between determiner and head noun, and restrictions on the distribution of dummy NPs. All of these co-occurrence restrictions are quite local, in the sense that they involve limitations on what can occur together as elements of a single clause. We extended this locality slightly with our analysis of raising, which in effect permits the co-occurrence restrictions of one verb to be transmitted to a higher verb.

The present chapter introduces a new type of construction in which the locality of co-occurrence restrictions appears to be violated in a more radical way. In these cases, two elements (say, an NP and a verb) appear far from one another in a sentence, despite the existence of a syntactic dependency (such as case marking or agreement) between them. Handling these 'long distance dependencies' (or LDDs, as we will call them) will require several changes to our theory:

- two new features,
- reformulation of the constraints on the types *word*, *lexeme* and *l-rule*, and on the initial symbol (in reference to the new features),
- a minor reformulation of some of our grammar rules,
- a new principle,
- a new grammar rule, and
- a new lexical rule.

14.2 Some Data

Our current grammar correctly rules out examples like the following:

(1) a.*They handed to the baby.
  b.*They handed the toy.
  c.*You have talked to.
  d.*The children discover.

Because the lexical entry for *hand* specifies that its COMPS list has both an object NP and a PP, (1a–b) are ruled out through the interaction of the lexicon, the headed grammar rules, the Argument Realization Principle, and the Valence Principle. Similarly, (1c–d) are ruled out because both the preposition *to* and the verb *discover* require an object NP, which is absent from these examples.

So it's interesting to find that there are grammatical sentences that contain exactly the ungrammatical strings of words in (1). For example, there are questions containing *wh*-words ('*wh*-questions') such as following:

(2) a. What did they hand to the baby?
  b. To whom did they hand the toy?
  c. Who(m) should you have talked to?
  d. What will the children discover?

There are also NPs modified by RELATIVE CLAUSES which contain the same ungrammatical strings:

(3) a. The toy which they handed to the baby...
  b. The baby to whom they handed the toy ...
  c. The people who(m) you have talked to...
  d. The presents that the children discover...

Another sort of example is a kind of sentence that is used for a certain sort of emphasis that is usually called a 'topicalized' sentence. In such sentences, a topicalized element can be followed by one of those same ungrammatical word sequences in (1):[1]

(4) a. That toy, they handed to the baby.
  b. To the baby, they handed a toy.
  c. That kind of person, you have talked to (many times).
  d. Presents that come from grandma, the children (always) discover.

And finally, there are certain adjectives like *easy* and *hard* whose infinitival complements may contain a verb or preposition lacking a normally obligatory object:

(5) a. That toy would be easy to hand to the baby.
  b. You are easy to talk to.
  c. The presents from grandma were hard for the children to discover.

In each of the examples in (2)–(5), there is a dependency between a phrase or 'filler' at the beginning of a clause and a 'gap' somewhere within the clause. In questions, relative

[1] When examples like (4) are first presented, some students claim that they find them unacceptable, but examination of actual usage indicates that topicalization is quite common, e.g. in examples like the following:

(i) Me, you bring an empty food dish; him, you bring a leash. (from a cartoon)
(ii) The film clips you're going to see tonight, no one's ever seen before. (Carol Burnett radio ad, November 26, 2001)

The name 'topicalization' is actually rather misleading. To be sure, the fronted element refers to an entity whose role in the discourse is distinguished in some way, but that entity need not correspond to the 'topic of discussion' in any straightforward way, as (i) indicates.

clauses, and topicalized sentences, the filler appears to be an extra phrase in that position; in examples like (5), the subject of the clause also serves as the filler.

In short, we see that elements whose presence is usually required in a clause are allowed to be absent if there is an appropriate filler in the right place. Likewise, if there is a filler, then there must be a gap somewhere within the sentence that follows the filler:

(6) a.*What did Kim hand the toys to the baby?
b.*The dolls that Kim handed the toys to the baby....
c.*The dolls, Kim handed the toys to the baby.
d.*The dolls are easy to hand the toys to the baby.

In such constructions, the filler can be separated from the gap by extra clauses, as indicated in (7)–(10). To help readers identify the location of the gaps, we have marked them with an underlined space.

(7) a. What did you say they handed __ to the baby?
b. Who(m) did he claim that they handed the toy to __ ?
c. Who(m) do you think you have talked to __ ?
d. What will he predict that the children discover __ ?

(8) a. The toy which we believe they handed __ to the baby...
b. The baby that I think they handed the toy to __ ...
c. The person who(m) everyone thinks you have talked to __ ...
d. The presents that it annoys me that the children discover __ ...

(9) a. That toy, I think they handed __ to the baby.
b. This baby, I know that they handed a toy to __ .
c. That kind of person, you know you have talked to __ .
d. Presents that come from grandma, I know that the children (always) discover __ .

(10) a. This toy isn't easy to try to hand __ to the baby.
b. The baby is easy to ask someone to hand a toy to __ .
c. That kind of person is hard to find anyone to talk to __ .
d. Presents from grandma are easy to help the children to discover __ .

In fact, there can be multiple extra clauses intervening:

(11) What did you think Pat claimed I said they handed __ to the baby?

14.3 Formulating the Problem

We want to be able to build clauses with elements missing within them. But somehow we have to keep track of the fact that something is missing. Furthermore, as the following contrasts show, we need to keep track of just what is missing:

(12) a. This, you can rely on.
b.*This, you can rely.
c.*On this, you can rely on.
d. On this, you can rely.
e.*On this, you can trust.

(13) a. Him, you can rely on.
b.*He, you can rely on.

(14) a. The twins, I can't tell the difference between.
b.*That couple, I can't tell the difference between.

Exercise 1: Long-Distance Selectional Dependencies
What exactly is wrong with the starred examples in (12)–(14)? Which element is selecting for the missing (or 'gapped') element, and which requirement of the selecting head does the filler not fulfill?

We can think of this as an information problem. We have to make sure that the phrases within the sentence keep track of what's missing from them as they are built. This has to be done just right, so that sentences missing a phrase of category X (no matter how deeply embedded that gap may be) combine with a filler of category X, and that fillers are allowed only when there is a gap for them to fill (cf. (6)).

14.4 Formulating a Solution

Our solution to this information problem will involve breaking it down into three parts: the bottom, the middle and the top. The bottom of an LDD is where the gap is 'introduced' – i.e. the smallest subtree where something is missing. Many theories handle the bottom by positing an empty element in the tree. We will avoid using empty elements in this way and instead handle the bottom by means of a feature (GAP) and a revision to the ARP that allows ARG-ST elements to show up on GAP instead of on the COMPS list. This is the topic of Section 14.4.1. The middle of an LDD is the 'transmission' of the information about what is missing from bottom to top (alternatively, the 'transmission' of what is available as a filler from top to bottom). We will handle this by means of a principle that relates the GAP values of phrases to the GAP values of their daughters. This is the topic of Section 14.4.2. The top of an LDD is where the filler is introduced, and the GAP requirement cancelled off. How exactly this happens depends on the particular kind of LDD. In Section 14.4.3, we will consider two kinds: 'topicalized' sentences, which we analyze in terms of a new phrase structure rule, and LDDs with *easy*-class adjectives, where the lexical entry for the adjective handles the top of the LDD.

14.4.1 The Feature GAP

We introduce the feature GAP (on *syn-cat*) to encode the fact that a phrase is missing a certain kind of element. There are examples of clauses where more than one phrase is missing,[2] a phenomenon we will return to in Problem 5 below:

(15) a. Problems this involved, my friends on the East Coast are hard to talk to __ about __ .
b. Violins this well crafted, these sonatas are easy to play __ on __ .

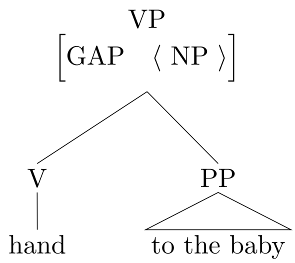
[2]Or, as linguists sometimes say (though it is somewhat of an oxymoron): 'where more than one gap appears'.

Note that the two gaps in each of these sentences have distinct fillers. In (15a), for example, the filler for the first gap is *my friends on the East Coast*, and the filler for the second one is *problems this involved*. Such examples are rare in English and sound a bit awkward, but there are other languages (for example several Slavic and Scandinavian languages) that allow multiple gaps more freely.

Given the existence of sentences with multiple gaps, we need a mechanism that can keep track of multiple missing elements. This suggests that the value of GAP is a list of feature structures, like the values of COMPS, SPR, MOD, and ARG-ST.

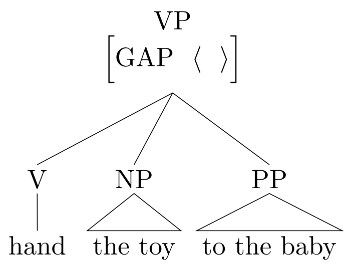
The intuitive significance of a phrase specified as, say, [GAP ⟨ NP ⟩] is that it is missing exactly one NP. The trick will be to make GAP have the right values in the right places. What we want is to allow a transitive verb or preposition to build a VP or PP without ever combining with an object NP. Furthermore, we want to ensure that it is only when an NP is absent that the relevant phrase is specified as [GAP ⟨ NP ⟩], as illustrated in (16):

(16)



When nothing is missing, we want the relevant phrase to be [GAP ⟨ ⟩], as in (17):

(17)



We will deal with this kind of 'missing element' as an instance of something that is present in argument structure but absent from the valence features. We could accomplish this by means of a lexical rule, but a more general solution is to modify the Argument Realization Principle. Our current version of the principle says that a word's SPR and COMPS lists add up to be its argument structure (ARG-ST) list. We now want to allow for the possibility that some element or elements of ARG-ST are on neither the SPR list nor the COMPS list, but on the GAP list instead.

To make this modification precise, we will introduce a kind of subtraction operation on lists, which we will mark with the symbol ⊖. Intuitively, if A and B are lists, then A ⊖ B is a list that results from removing the elements of B from A. A couple of caveats are in order here. First, we want A ⊖ B to be defined only when the elements of B all occur in A, and in the same order. So there are many pairs of lists for which this kind of list subtraction is undefined. This is unlike our form of list addition (⊕), which is defined for any pair of lists. Second, when A $\ominus$ B is defined, it need not be unique. For example, if A = ⟨NP, PP, NP⟩ and B = ⟨NP⟩, then there are two possible values for A $\ominus$ B, namely ⟨NP, PP⟩ and ⟨PP, NP⟩. We will interpret an equation like A $\ominus$ B = C to mean that there is some value for A $\ominus$ B that is identical to C.

With this new tool in hand, we can restate the Argument Realization Principle as follows:

(18) Argument Realization Principle:

$$\textit{word} : \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \boxed{\text{A}} \\ \text{COMPS} & \boxed{\text{B}} \ominus \boxed{\text{C}} \end{bmatrix} \\ \text{GAP} & \boxed{\text{C}} \end{bmatrix} \\ \text{ARG-ST} & \boxed{\text{A}} \oplus \boxed{\text{B}} \end{bmatrix}$$

The revised ARP guarantees that any argument that could appear on a word's COMPS list can appear on its GAP list instead. (We will deal with gaps that correspond to subjects, rather than complements, in Section 14.5) Further, (18) guarantees that whenever an argument is missing, any co-occurrence restrictions the word imposes on that argument will be registered on the element that appears on the GAP list.

Because the result of list subtraction ($\ominus$), as we have defined it, is not always unique, when we specify the ARG-ST in a verb's lexical entry without also specifying its SPR, COMPS, and GAP values, we are actually providing an underspecified lexical entry that will give rise to a family of words that differ with respect to how the ARP is satisfied. Consider, for example, the lexical entry for the lexeme *hand*, as specified in (19):

(19)

$$\left\langle \text{hand} ,\ \begin{bmatrix} \textit{ptv-lxm} & \\ \text{ARG-ST} & \left\langle \text{X}_i \ ,\ \text{Y}_k \ ,\ \begin{bmatrix} \text{FORM} & \text{to} \\ \text{INDEX} & j \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & \text{s} \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{hand} \\ \text{SIT} & s \\ \text{HANDER} & i \\ \text{HANDED-TO} & j \\ \text{HANDED-ITEM} & k \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

This can undergo the Non-3rd-Singular Verb Lexical Rule presented in Chapter 8, which gives rise to lexical sequences which satisfy the following description:

(20)

$$\left\langle \text{hand} ,\ \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{fin} \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \begin{bmatrix} \text{CASE} & \text{nom} \\ \text{AGR} & \textit{non-3sing} \end{bmatrix} ,\ \overset{\text{NP}}{\begin{bmatrix} \text{CASE acc} \end{bmatrix}} ,\ \begin{bmatrix} \text{FORM} & \text{to} \end{bmatrix} \right\rangle \end{bmatrix} \right\rangle$$

Since the second member of these lexical sequences is of type *word*, it is subject to the ARP. But now there are multiple ways to satisfy the ARP. In particular, the family of lexical sequences described in (20) includes lexical sequences meeting each of the following (more detailed) descriptions:

(21)

$$\left\langle \text{hand}\ , \left[\begin{array}{ll} \textit{word} & \\ \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll}\text{FORM} & \text{fin}\end{array}\right] \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \langle\ \boxed{1}\ \rangle \\ \text{COMPS} & \langle\ \boxed{2}\text{NP[acc]}\ ,\ \boxed{3}\text{PP[to]}\ \rangle \end{array}\right] \\ \text{GAP} & \langle\ \rangle \end{array}\right] \\ \text{ARG-ST} & \left\langle \overset{\boxed{1}\text{NP}}{\left[\begin{array}{ll}\text{CASE} & \text{nom}\\ \text{AGR} & \textit{non-3sing}\end{array}\right]}, \boxed{2}\ ,\ \boxed{3} \right\rangle \end{array}\right] \right\rangle$$

(22)

$$\left\langle \text{hand}\ , \left[\begin{array}{ll} \textit{word} & \\ \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll}\text{FORM} & \text{fin}\end{array}\right] \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \langle\ \boxed{1}\ \rangle \\ \text{COMPS} & \langle\ \boxed{3}\text{PP[to]}\ \rangle \end{array}\right] \\ \text{GAP} & \langle\ \boxed{2}\text{NP[acc]}\ \rangle \end{array}\right] \\ \text{ARG-ST} & \left\langle \overset{\boxed{1}\text{NP}}{\left[\begin{array}{ll}\text{CASE} & \text{nom}\\ \text{AGR} & \textit{non-3sing}\end{array}\right]}, \boxed{2}\ ,\ \boxed{3} \right\rangle \end{array}\right] \right\rangle$$

(23)

$$\left\langle \text{hand}\ , \left[\begin{array}{ll} \textit{word} & \\ \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll}\text{FORM} & \text{fin}\end{array}\right] \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \langle\ \boxed{1}\ \rangle \\ \text{COMPS} & \langle\ \boxed{2}\text{NP[acc]}\ \rangle \end{array}\right] \\ \text{GAP} & \langle\ \boxed{3}\text{PP[to]}\ \rangle \end{array}\right] \\ \text{ARG-ST} & \left\langle \overset{\boxed{1}\text{NP}}{\left[\begin{array}{ll}\text{CASE} & \text{nom}\\ \text{AGR} & \textit{non-3sing}\end{array}\right]}, \boxed{2}\ ,\ \boxed{3} \right\rangle \end{array}\right] \right\rangle$$

All of these are legitimate lexical sequences: (21) shows *hand*'s feature structure in sentences like (24a); (22) is the way *hand* appears in the tree our grammar assigns to sentences like (24b); and (23) shows *hand* as it appears in the tree we assign to sentences like (24c):[3]

(24) a. You handed the toy to the baby.
b. What did you hand to the baby?
c. To whom did you hand the toy?

The prepositional lexeme in (25) will now give rise to the word structures sketched in (26) and (27) (omitting what is not directly relevant):

(25)

$$\left\langle \text{to}\ ,\ \begin{bmatrix} \textit{argmkp-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{prep} & \\ \text{FORM} & \text{to} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \boxed{1}\text{NP[acc]} \right\rangle \end{bmatrix} \right\rangle$$

(26)

$$\begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{prep} & \\ \text{FORM} & \text{to} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \boxed{2}\text{NP[acc]}\ \rangle \end{bmatrix} \\ \text{GAP} & \langle\ \rangle \end{bmatrix} \\ \text{ARG-ST} & \langle\ \boxed{2}\ \rangle \end{bmatrix}$$

$$|$$

to

[3]The ARP also allows for a family of lexical sequences in which both the NP and PP complements are in the GAP list, rather than the COMPS list. We will return to multiple-gap sentences in Problem 5 below.

(27)
$$\begin{array}{c}\begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix}\textit{prep} & \\ \text{FORM} & \text{to}\end{bmatrix} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix} \\ \text{GAP} & \langle\ \boxed{2}\text{NP[acc]}\ \rangle \end{bmatrix} \\ \text{ARG-ST} & \langle\ \boxed{2}\ \rangle \end{bmatrix} \\ | \\ \text{to}\end{array}$$

This last lexical tree is the one that allows for sentences like (28):

(28) Which baby did you hand the toy to?

14.4.2 The GAP Principle

The GAP feature tells us which of a word's arguments is missing. The Argument Realization Principle, as we have reformulated it, permits us to instantiate gaps freely (other than elements that must be on the SPR list). Now we need some way of passing the information in the GAP value up[4] from words like those just illustrated so that the phrases that they head will register the fact that something is missing, and from those phrases to larger phrases. To do so, we adopt the principle shown in (29):

(29) The GAP Principle (Preliminary Version)

A local subtree Φ satisfies the GAP Principle with respect to a headed rule ρ if and only if Φ satisfies:

$$\begin{array}{ccc} & \begin{bmatrix}\text{GAP} & \boxed{A_1} \oplus ... \oplus \boxed{A_n}\end{bmatrix} & \\ \diagup & | & \diagdown \\ [\ \text{GAP}\ \boxed{A_1}\ ] & \ldots & [\ \text{GAP}\ \boxed{A_n}\ ] \\ \triangle & & \triangle \end{array}$$

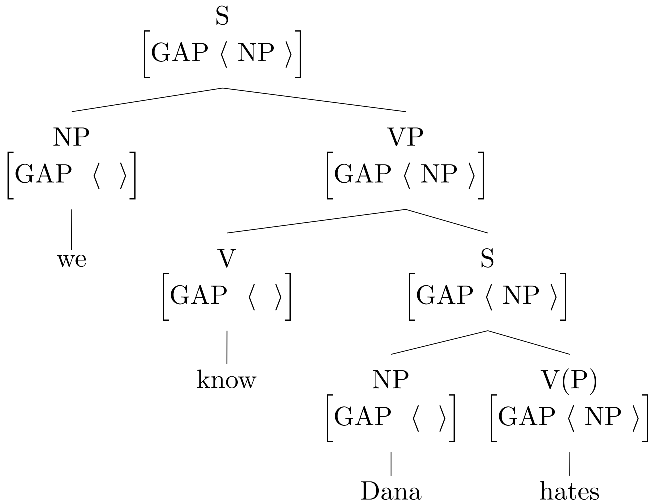
In other words, in a headed structure, the GAP values of all the daughters must add up to be the GAP value of the mother. That is, a phrase whose daughter is missing something is missing the exact same thing. There is one exception to this generalization, and that is the case where the larger phrase also contains the filler. We'll return to these cases directly.

The notion of lists 'adding up to' something is the same one we have employed before, namely the operation that we denote with the symbol '$\oplus$'. In most cases, most of the

[4]The metaphor of passing information between nodes should again not be taken literally. What the principle in (29) does is similar to what the Head Feature Principle and Valence Principle do, namely, enforce a particular relationship between certain feature values in mothers and daughters in phrase structure trees. That is, it is simply part of our definition of phrase-structure well-formedness.

GAP lists that are added up in this way are in fact empty, because most constituents don't contain gaps, so the addition is quite trivial. The effect of (29), then, given our lexical entries (and the word structures they sanction in virtue of our revision of the ARP), is illustrated in (30):

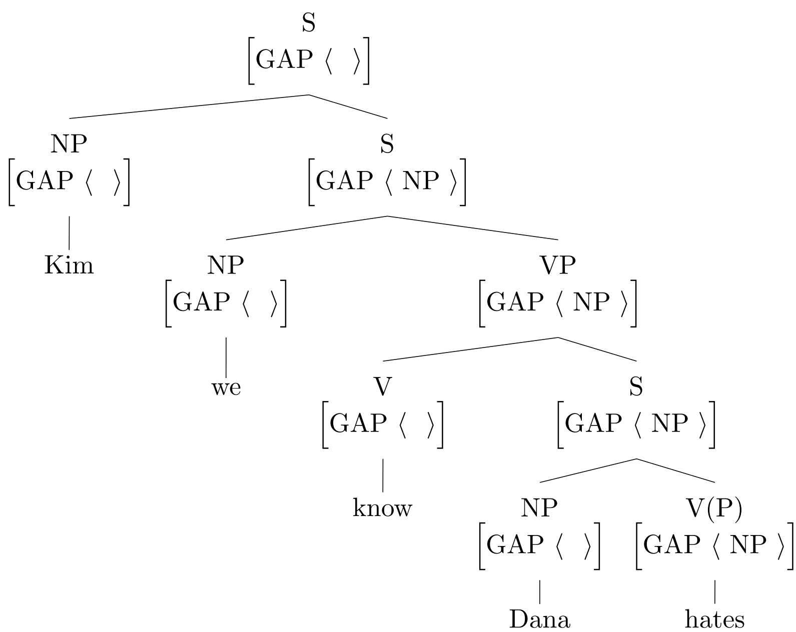
(30)



Note that each local tree in (30) satisfies the GAP Principle. That is, in each tree, the GAP values of the daughters add up to the mother's GAP value: $(\langle\ \rangle \oplus \langle \text{NP} \rangle) = \langle \text{NP} \rangle$

We now return to the exception (mentioned above) to the GAP Principle, as stated in the preliminary version: At the top of the LDD, where the gap is filled, we want the mother node to be [GAP ⟨ ⟩]. This is illustrated in (31):

(31)



We have not yet seen the phrase structure rule which licenses the topmost subtree of (31). It will be introduced in the next subsection. Here, we are concerned with the GAP values in that subtree. We want the mother to be [GAP $\langle\ \rangle$] as shown, because, intuitively, the NP *Kim* is 'filling' the gap. That is, the tree structure shown in (31) is no longer 'missing something', and this should be reflected in the GAP value of the root node in (31).

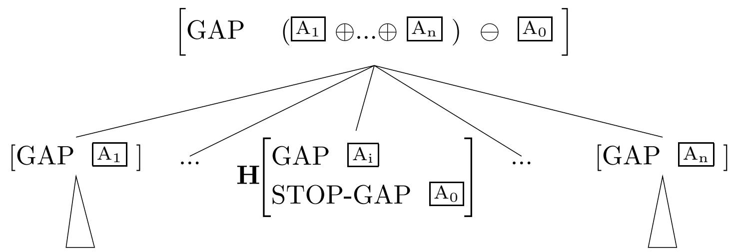
Adjectives like *hard* and *easy*, which we discussed earlier, also perform a gap-filling function, even though they also serve as the head daughter of a head-complement phrase. The VP in (32a) is 'gappy' – it is missing an NP and hence should be specified as [GAP $\langle$ NP $\rangle$], while the AP in (32b) is not gappy and should be specified as [GAP $\langle\ \rangle$], like all other APs that we have encountered.

(32) a. [to talk to __]
b. [easy to talk to __]

We will provide a unified account of gap filling by introducing a new list-valued feature called STOP-GAP. Like GAP, STOP-GAP is a feature of *syn-cat*s. This feature signals what gap is to be filled in the local subtree where it appears. Most nodes will be [STOP-GAP $\langle\ \rangle$], but where a gap is associated with its filler, the feature has a non-empty list as its value. In particular, the lexical entries for gap stoppers like *easy* and *hard* will specify a non-empty value for this feature, as will the grammar rule we introduce for the topicalization construction. Making use of this new feature, we can reformulate the GAP Principle so that it passes up GAP values only if they are not filled. This is shown in (33):

(33) The GAP Principle (Final Version)

A local subtree Φ satisfies the GAP Principle with respect to a headed rule ρ if and only if Φ satisfies:



What this revision says is that the GAP value of the mother node in a headed structure is determined by adding up the GAP values of all the daughters and then subtracting any gaps that are being filled, as indicated by the head daughter's STOP-GAP value.

14.4.3 The Head-Filler Rule and *Easy*-Adjectives

We have dealt with the bottom of LDDs, where non-empty values for GAP are introduced, and the middle of LDDs where those GAP values are propagated through the tree (until they meet their fillers). Now we turn to the top of LDDs: the filling of the gap. As noted above, we will consider two types of gap-filling here: topicalized sentences and *easy*-adjectives.

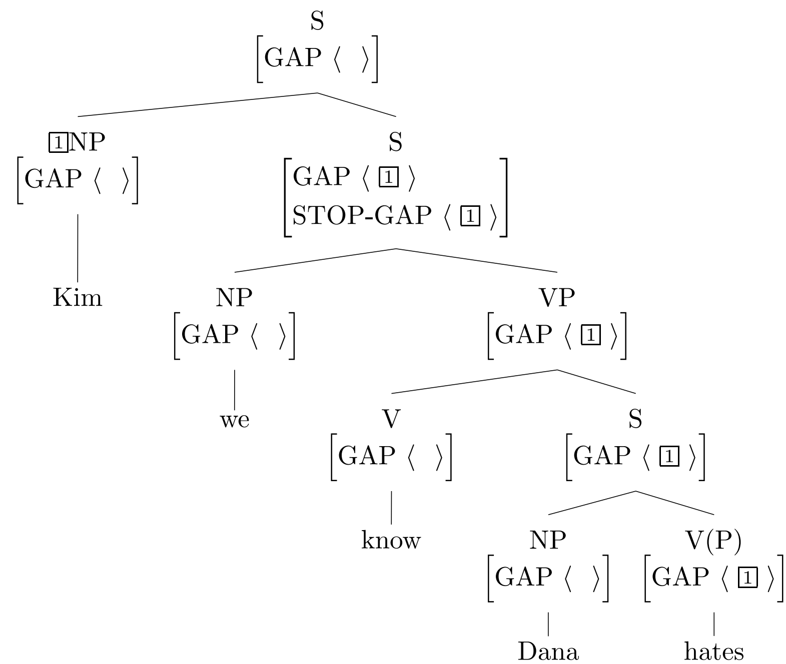
To deal with topicalized sentences, we now introduce a new grammar rule, formulated as follows:

(34) Head-Filler Rule

$$[\textit{phrase}] \rightarrow \boxed{1}\begin{bmatrix}\text{GAP} & \langle\ \rangle\end{bmatrix} \quad \mathbf{H}\begin{bmatrix} \text{HEAD} & \begin{bmatrix}\textit{verb} & \\ \text{FORM} & \text{fin}\end{bmatrix}\\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix}\\ \text{STOP-GAP} & \langle \boxed{1} \rangle \\ \text{GAP} & \langle \boxed{1}\rangle\end{bmatrix}$$

This rule says that a phrase can consist of a head with a gap preceded by an expression that meets whatever requirements the head places on that gap.[5] The Head-Filler Rule licenses the topmost subtree in (35), and it enforces the identity between the NP *Kim* and the element on the GAP list of the gappy S *we know Dana hates* ($\boxed{1}$). Because that GAP element is identified with the GAP element of the V *hates* (and therefore also with an element of its ARG-ST list), any requirements that *hates* places on its complement (that it be a [CASE acc] NP, that its INDEX be identified with the HATED value in the **hate** predication) must be satisfied by the filler *Kim*.

(35)



The topmost node of (35) is [GAP ⟨ ⟩], indicating that the gap has been filled, thanks to the GAP Principle: The Head-Filler Rule in (35) specifies that the head daughter's GAP list and STOP-GAP list both contain the filler daughter, so that element is subtracted from the head daughter's GAP value in determining the GAP value of the

[5] And further that the filler must not be gappy.

mother:
$(\langle\ \rangle \oplus \langle\boxed{1}\rangle) \ominus \langle\boxed{1}\rangle = \langle\ \rangle$.

It is important to see that our analysis entails that a filler NP can appear before a clause only when that clause is gappy, i.e. when that clause is missing an NP that would normally appear there. Moreover, the Head-Filler Rule does not require the filler to be an NP, but it does require that the filler's *synsem* be identified with the unique member of the head daughter's GAP list. From this it follows that topicalized sentences may contain PP fillers (and perhaps fillers of other categories) just as long as the gap within the clause matches the synsem of the filler. That is, if the filler is a PP, then the missing element must be a PP, not an NP. This is a consequence of the many identities triggered by the Head-Filler Rule and the GAP Principle, interacting with the Argument Realization Principle and particular lexically specified ARG-ST values.

We now turn to our other example of gap filling, adjectives like *easy* or *hard*. Most words don't fill gaps, so we will posit the following defeasible constraint on the type *lexeme*:

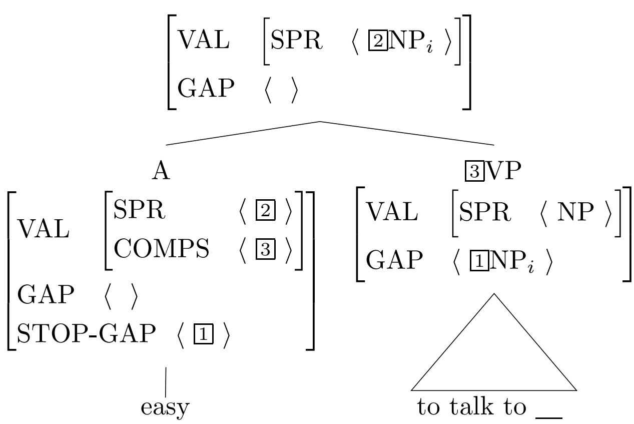
(36) $$\textit{lexeme} : \left[\text{STOP-GAP} \ \ / \ \langle\ \rangle\right]$$

Adjectives like *easy* or *hard* are the exceptions. We give them lexical entries which override this constraint, as shown for *easy* in (37):

(37) $$\left\langle \text{easy}\ , \begin{bmatrix} \textit{adj-lxm} & \\ \text{SYN} & \left[\text{STOP-GAP} \ \langle\ \boxed{1}\ \rangle\right] \\ \text{ARG-ST} & \left\langle \text{NP}_i\ ,\ \begin{array}{c}\text{VP}\\ \begin{bmatrix}\text{INF} & + \\ \text{GAP} & \langle\ \boxed{1}\text{NP}_i\ ,\ ...\ \rangle\end{bmatrix}\end{array} \right\rangle \end{bmatrix} \right\rangle$$

Because the member of the STOP-GAP list in (37) is identified with the first member of the VP argument's GAP list, adjectives of this type must perform gap stopping of the sort shown in (38):

(38)



Notice that the GAP list is empty at the top node of this subtree. That is, the AP *easy to talk to* is treated as having no gap, even though the infinitival VP *to talk to* inside it has an NP gap. This may seem puzzling, since *easy to talk to* seems to be missing the same NP as *to talk to*. But at the level of the AP, the referent of the missing NP is fully determined: it is the same as the subject of the AP. Hence, the GAP list at the AP level no longer needs to register the missing NP. Instead, the first argument (that is, the subject) of the AP is coindexed with the NP in the GAP list.[6] This guarantees that, in a sentence like (39), the Pat is understood as the person who is followed:

(39) Pat is easy to continue to follow __ .

14.4.4 GAP and STOP-GAP in the Rest of the Grammar

We have added two features to our grammar (GAP and STOP-GAP) which are involved in passing information around the tree. As such, we must pause and ask whether the rest of our grammar (in particular, lexical rules, the rest of our grammar rules and the initial symbol) are currently doing the right thing with respect to these new features. The answer is (unsurprisingly) that we will need to make a few modifications.

First, with respect to the feature GAP: Nothing we have said so far ensures that all gaps ultimately get filled. We make sure that SPR and COMPS requirements are ultimately fulfilled by requiring that both be empty on the initial symbol. We can do the same for GAP. That is, our initial symbol is now the following:

(40)
$$\begin{bmatrix} \textit{phrase} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} \\ \text{FORM} & \text{fin} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \\ \text{GAP} & \langle\ \rangle \end{bmatrix} \end{bmatrix}$$

Without this specification, we would license examples like (1), repeated here for convenience, as stand-alone utterances:

(41) a.*They handed to the baby.
b.*They handed the toy.
c.*You have talked to.
d.*The children discover.

The other consideration with respect to the feature GAP is whether its value is sufficiently constrained. The GAP Principle applies to headed phrases, but not non-headed phrases. Thus, in our discussion so far, we have not constrained the GAP value of coordinate phrases or imperatives. We will return to coordination in Section 14.6 below. As for imperatives, in order to ensure that we don't allow gappy VPs as the daughter (as in (42)), we can identify the mother's and daughter's GAP values, as shown in (43). Since imperative phrases must also satisfy the initial symbol, they must be [GAP $\langle\ \rangle$] on the mother.

(42)*Hand the toy!

[6] More precisely, with the NP in initial position in the GAP list.

(43) Imperative Rule (Revised Version)

$$\begin{bmatrix} \textit{phrase} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \rangle\end{bmatrix} \\ \text{GAP} & \boxed{\text{A}} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{dir} \\ \text{INDEX} & s \end{bmatrix} \end{bmatrix} \rightarrow \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{INF} & - \\ \text{FORM} & \text{base} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \text{NP[PER 2nd]} \right\rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \\ \text{GAP} & \boxed{\text{A}} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s\end{bmatrix} \end{bmatrix}$$

Thanks to the GAP Principle and the two modifications given above, GAP values are now sufficiently constrained throughout our grammar. We haven't said much about STOP-GAP values, however, except to say that they are non-empty in two places: on the head daughter of a head-filler phrase, and in the lexical entries for adjectives like (*easy*). In addition, the defeasible constraint given in (36) above and repeated here ensures that all other lexical entries are [STOP-GAP $\langle\ \rangle$]:

(44) $$\textit{lexeme} : \begin{bmatrix}\text{STOP-GAP} & / \langle\ \rangle\end{bmatrix}$$

Since we want the STOP-GAP values given on lexemes to be reflected in the word-structures they license, we need to make sure that all lexical rules preserve that information. We do that by adding the following non-defeasible constraint to the type *l-rule*:

$$\textit{l-rule} : \begin{bmatrix} \text{INPUT} & \left\langle \text{X}\ ,\ [\text{STOP-GAP}\ \ \boxed{\text{A}}] \right\rangle \\ \text{OUTPUT} & \left\langle \text{Y}\ ,\ [\text{STOP-GAP}\ \ \boxed{\text{A}}] \right\rangle \end{bmatrix}$$

When STOP-GAP is non-empty, the GAP Principle subtracts the relevant element from the GAP list being passed 'up' the tree. It follows that we want to ensure that STOP-GAP is empty when there is no gap-filling going on. Gaps are never filled in head-specifier or head-modifier phrases, so we constrain the head daughters of the Head-Specifier and Head-Modifier Rules to be [STOP-GAP $\langle\ \rangle$]:

(45) Head-Specifier Rule (Revised Version)

$$\begin{bmatrix} \textit{phrase} & \\ \text{SPR} & \langle\ \rangle \end{bmatrix} \rightarrow \boxed{1} \quad \mathbf{H}\begin{bmatrix} \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{STOP-GAP} & \langle\ \rangle \end{bmatrix}$$

(46) Head-Modifier Rule (Revised Version)

$$[\textit{phrase}] \rightarrow \mathbf{H}\boxed{1}\begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{STOP-GAP} & \langle\ \rangle \end{bmatrix} \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle \boxed{1} \rangle \end{bmatrix}$$

Gap-filling sometimes occurs in head-complement phrases (in particular, when the head is an adjective like *easy*), so we do not want to constrain the head daughter of the Head-Complement Rule to be [STOP-GAP $\langle\ \rangle$]. However, since the head daughter of this rule is always a *word*, the STOP-GAP value will be appropriately constrained by the lexical entries.

This completes our discussion of complement gaps.[7]

14.5 Subject Gaps

We have covered only the basic cases of long-distance dependencies. There are many additional complexities. For example, we have not discussed cases in which the gaps are not complements, but rather subjects or modifiers. In addition, we have not discussed the distribution of *wh*-words (such as *who, what, which*, etc.) in questions and relative clauses, nor the obligatory inverted order of subject and auxiliary verb in many *wh*-questions. There is a rich literature investigating these and many other questions associated with LDDs, but such matters are beyond the scope of this text. In this section we sketch the basics of an account of what is subject extraction – that is LDDs in which the gaps are in subject position.

Our present account does not yet deal with examples like (47):

(47) a. Which candidates do you think like oysters on the half-shell?
b. That candidate, I think likes oysters on the half-shell.

This is because of an interaction between the ARP and the constraints (including the SHAC, inherited from *infl-lxm*) that all verb lexemes have SPR lists of length one. Together, these constraints require that the first member of a verb's ARG-ST list must appear on its SPR list. It may not belong to the rest of the list – i.e. to the list of elements that can appear on either COMPS or GAP, according to the ARP.

Rather than attempt to revise the ARP to handle these cases, we will treat them in terms of a post-inflectional lexical rule which provides [SPR ⟨ ⟩] lexical sequences for verbs, and puts the right information into the GAP list:

(48) Subject Extraction Lexical Rule

$$\begin{bmatrix} \textit{pi-rule} & \\ \text{INPUT} & \left\langle \text{X}\, , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{fin} \end{bmatrix} \\ \text{VAL} & [\text{SPR} \;\; \langle \text{Z} \rangle] \end{bmatrix} \\ \text{ARG-ST} & \boxed{\text{A}} \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \text{Y}\, , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & [\text{SPR} \;\; \langle\ \rangle] \\ \text{GAP} & \langle \boxed{1} \rangle \end{bmatrix} \\ \text{ARG-ST} & \boxed{\text{A}} \langle \boxed{1}\, , \ldots \rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

This rule maps any finite verb form into a word with an empty SPR list and a GAP list containing an element identified with the first argument – the subject of the verb. The

[7]There are further constraints governing complement gaps that we will not treat here. For example, an ADV$_{pol}$ like *not* or accented SO, which were analyzed as complements in Chapter 13, cannot serve as a topicalization filler:

(i) *Not, Kim will go to the store.
(ii) *So, Kim will go to the store.

This contrasts with the behavior of adverbial modifiers (left untreated in this text), which may be topicalized:

(iii) Tomorrow, (I think) Kim will go to the store __ .

lexical sequences that are the outputs of this rule are illustrated by the description in (49):

(49)
$$\left\langle \text{likes}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{fin} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle \boxed{2} \rangle \end{bmatrix} \\ \text{GAP} & \left\langle \boxed{1} \begin{bmatrix} \text{CASE} & \text{nom} \\ \text{AGR} & \textit{3sing} \end{bmatrix} \right\rangle \\ \text{STOP-GAP} & \langle\ \rangle \end{bmatrix} \\ \text{ARG-ST} & \langle \boxed{1}\ ,\ \boxed{2}\text{NP[acc]} \rangle \end{bmatrix} \right\rangle$$

Note that the ARP (inherited from the type *word*) is satisfied in (49): the SPR list is empty, and the rest of the ARG-ST list (i.e. the whole ARG-ST list) is appropriately related to the list values of COMPS and GAP. That is, the COMPS value (⟨NP[acc]⟩) is just the ARG-ST value (50a) minus the GAP value (50b):

(50) a.
$$\left\langle \begin{bmatrix} \text{CASE} & \text{nom} \\ \text{AGR} & \textit{3sing} \end{bmatrix},\ \begin{matrix} \text{NP} \\ [\text{CASE acc}] \end{matrix} \right\rangle$$

b.
$$\left\langle \begin{bmatrix} \text{CASE} & \text{nom} \\ \text{AGR} & \textit{3sing} \end{bmatrix} \right\rangle$$

14.6 The Coordinate Structure Constraint

One of the most discussed topics related to LDDs concerns restrictions on possible filler/gap associations. Although the position of filler and gap may be arbitrarily far apart, there are certain configurations that do not permit LDDs. Such configurations are known as 'islands' (a term due to Ross (1967)), and a major goal of syntactic research over the past three decades has been to understand where and why islands occur. In this section, we will look at one type of island and show how our grammar correctly predicts its existence and its properties.

The following examples illustrate what Ross called the 'Coordinate Structure Constraint':

(51) a.*Here is the student that [the principal suspended [__ and Sandy]].
b.*Here is the student that [the principal suspended [Sandy and __]].

(52) a.*Here is the student that [[the principal suspended __] and [the student council passed new rules]].
b.*Here is the student that [[the student council passed new rules] and [the principal suspended __]].

(53) a.*Apple bagels, I can assure you that [[Leslie likes __] and [Sandy hates lox]].
b.*Apple bagels, I can assure you that [[Leslie likes lox] and [Sandy hates __]].

Translating Ross's transformation-based formulation of the constraint into the language of fillers and gaps that we have been using, it can be stated as follows:

(54) Coordinate Structure Constraint (CSC)
In a coordinate structure,
(a) no conjunct can be a gap,
(b) nor can a gap be contained in a conjunct if its filler is outside of that conjunct.

(54a) is often referred to as the Conjunct Constraint, while (54b) is sometimes called the Element Constraint.

Ross also noticed a systematic class of exceptions to the Element Constraint, illustrated by (55):

(55) a. This is the dancer that we bought [[a portrait of ___] and [two photos of ___]].
b. Here is the student that [[the school suspended ___] and [we defended ___]].
c. Apple bagels, I can assure you that [[Leslie likes ___] and [Sandy hates ___]].

To handle examples like these, he appended an additional clause to the constraint, which we can formulate as follows:

(56) 'Across-the-Board' Exception (addendum to CSC):
. . . unless each conjunct properly contains a gap paired with the same filler.

As presented, the Coordinate Structure Constraint seems quite arbitrary, and the Across-the-Board Exception is just an added complication. And most analyses of these phenomena – specifically those that handle LDDs transformationally – have never come to grips with the full range of facts, let alone derived them from general principles.

Note first of all that the Conjunct Constraint is already explained by our grammar. Examples like (51) are ungrammatical for the simple reason that the elements on GAP lists must also be on ARG-ST lists, and coordinate conjunctions like *and* have empty ARG-ST lists. Unlike many other analyses (in particular, transformational approaches) our grammar does not employ empty elements (usually referred to as 'traces') to occupy the position of the gap in the syntactic structure. Since there are no empty NPs in our analysis, there is no empty element that could serve as a conjunct in a coordinate structure. That is, the Conjunct Constraint follows directly from the decision to treat the bottoms of LDDs in terms of an unrealized argument, rather than the presence of an empty element.

Now reconsider the grammar rule for coordination last updated in Chapter 8:

(57) Coordination Rule (Chapter 8 Version)

$$\begin{bmatrix}\text{FORM} & \boxed{1}\\ \text{VAL} & \boxed{0}\\ \text{IND} & s_0\end{bmatrix} \rightarrow$$

$$\begin{bmatrix}\text{FORM} & \boxed{1}\\ \text{VAL} & \boxed{0}\\ \text{IND} & s_1\end{bmatrix} \ldots \begin{bmatrix}\text{FORM} & \boxed{1}\\ \text{VAL} & \boxed{0}\\ \text{IND} & s_{n-1}\end{bmatrix} \begin{bmatrix}\text{HEAD} & \textit{conj}\\ \text{IND} & s_0\\ \text{RESTR} & \left\langle\begin{bmatrix}\text{ARGS}\ \langle s_1 \ldots s_n\rangle\end{bmatrix}\right\rangle\end{bmatrix} \begin{bmatrix}\text{FORM} & \boxed{1}\\ \text{VAL} & \boxed{0}\\ \text{IND} & s_n\end{bmatrix}$$

As stated, this rule doesn't say anything about the GAP values of the conjuncts or of the mother. (Note that the GAP Principle doesn't apply to subtrees licensed by this rule, as it is not a headed rule.) In our discussions of coordination so far, we have seen that some features must be identified across conjuncts (and with the mother) in coordination and that others should not. The Element Constraint examples cited above in (52) and (53) show that GAP is one of the features that must be identified. We thus modify our Coordination Rule slightly to add this constraint:

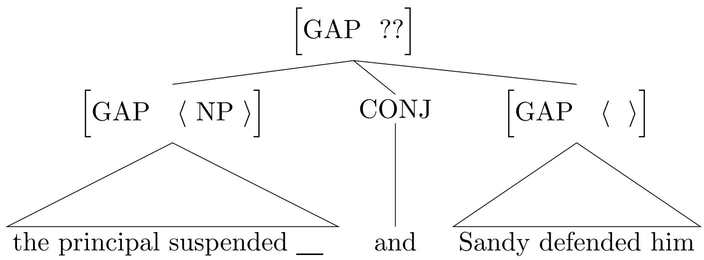
(58) Coordination Rule (Final Version)

$$\begin{bmatrix}\text{FORM} & \boxed{1}\\ \text{VAL} & \boxed{0}\\ \text{GAP} & \boxed{\text{A}}\\ \text{IND} & s_0\end{bmatrix} \rightarrow$$

$$\begin{bmatrix}\text{FORM} & \boxed{1}\\ \text{VAL} & \boxed{0}\\ \text{GAP} & \boxed{\text{A}}\\ \text{IND} & s_1\end{bmatrix} \ldots \begin{bmatrix}\text{FORM} & \boxed{1}\\ \text{VAL} & \boxed{0}\\ \text{GAP} & \boxed{\text{A}}\\ \text{IND} & s_{n-1}\end{bmatrix} \begin{bmatrix}\text{HEAD} & \textit{conj}\\ \text{IND} & s_0\\ \text{RESTR} & \left\langle\begin{bmatrix}\text{ARGS}\ \langle s_1 \ldots s_n\rangle\end{bmatrix}\right\rangle\end{bmatrix} \begin{bmatrix}\text{FORM} & \boxed{1}\\ \text{VAL} & \boxed{0}\\ \text{GAP} & \boxed{\text{A}}\\ \text{IND} & s_n\end{bmatrix}$$

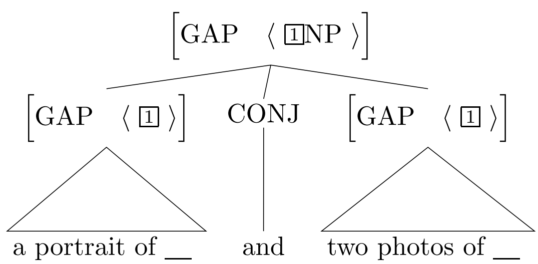
This revision guarantees that two conjuncts in a coordinate structure cannot differ in their GAP value. If one has an empty GAP list and the other has a nonempty GAP list (as in (51)–(53)), then the structure is not licensed. The GAP values that must be identical cannot be, as shown in (59):

(59)



On the other hand, it is possible for conjuncts to have nonempty GAP lists if they are all nonempty and all share the same value. This is what is illustrated in (55), whose structure is as shown in (60):

(60)



In short, both the Element Constraint and the Across-the-Board exceptions to it are treated properly in this revision of our analysis of coordination.

We close this discussion with one final observation about LDDs and coordinate structures. There is an exception to (56), illustrated by (61):

(61)*Which rock legend would it be ridiculous to compare [[__] and [__]]?

Our statements of the generalizations in (54) and (56), like Ross's original formulations of them, would in fact permit (61), whose deviance should have a syntactic (rather than a semantic) explanation, it would appear, because the meaning of this putative sentence could certainly be expressed as (62):

(62) Which rock legend would it be ridiculous to compare __ with himself?

But our analysis correctly rules out any sentences in which a gap constitutes a full conjunct. As noted above, this is because nonempty GAP values in the lexicon are licensed by the Argument Realization Principle, which allows ARG-ST elements not to be expressed as complements, rather than allowing them to appear as a phonetically empty element, or 'trace'. The difference is subtle, but the predictions are quite striking: our traceless analysis of gaps provides an immediate account of the deviance of (61) as well as an explanation of the examples in (51)–(53), which motivated Ross's Conjunct Constraint. The Coordinate Structure Constraint and its exceptions are thus properly accounted for in the analysis of coordination we have developed. Many alternative approaches – particularly those involving movement transformations to account for LDDs – have been unable to account for them at all.

14.7 Summary

Deducing the Conjunct Constraint from the interaction of our analyses of coordination and LDDs is an elegant result, providing significant support for our general approach to syntax. We also showed that we could extend our account of coordination in order to account for the Element Constraint as well.[8]

We will not examine other island constraints in this text. As with the Coordinate Structure Constraint, linguists have not been content to catalog the environments in which filler-gap pairings are impossible. Rather, a great deal of effort has gone into the search for explanations of syntactic islands, either in terms of the interaction of independently motivated elements of the theory (as in the example given above), or in terms of such factors as the architecture of the human language-processing mechanisms. This is a fertile area of research, in which definitive answers have not yet been found.

14.8 Changes to the Grammar

In this chapter, we developed an analysis of long-distance dependencies involving 'fillers' and unrealized elements, or 'gaps'. Our analysis involved two new features, GAP and STOP-GAP, both appropriate for feature structures of type *syn-cat*:

$$\textit{syn-cat}: \begin{bmatrix} \text{HEAD} & \textit{pos} \\ \text{VAL} & \textit{val-cat} \\ \text{GAP} & \textit{list}(\textit{expression}) \\ \text{STOP-GAP} & \textit{list}(\textit{expression}) \end{bmatrix}$$

[8]Essentially this account was first developed by Gazdar (1981), within the framework of Generalized Phrase Structure Grammar.

We treated the introduction of gaps at the bottom of LDDs in terms of the following modification of the Argument Realization Principle:

Argument Realization Principle:

$$\textit{word}: \left[\begin{array}{ll} \text{SYN} & \left[\begin{array}{ll} \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \boxed{\text{A}} \\ \text{COMPS} & \boxed{\text{B}} \ominus \boxed{\text{C}} \end{array}\right] \\ \text{GAP} & \boxed{\text{C}} \end{array}\right] \\ \text{ARG-ST} & \boxed{\text{A}} \oplus \boxed{\text{B}} \end{array}\right]$$

To introduce subject gaps, we created the following lexical rule:

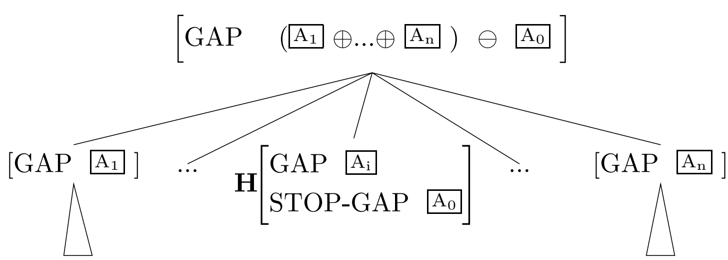
Subject Extraction Lexical Rule

$$\left[\begin{array}{ll} \textit{pi-rule} & \\ \text{INPUT} & \left\langle \text{X}, \left[\begin{array}{ll} \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll} \textit{verb} & \\ \text{FORM} & \text{fin} \end{array}\right] \\ \text{VAL} & \left[\text{SPR} \;\; \langle \text{ Z } \rangle\right] \end{array}\right] \\ \text{ARG-ST} & \boxed{\text{A}} \end{array}\right] \right\rangle \\ \text{OUTPUT} & \left\langle \text{Y}, \left[\begin{array}{ll} \text{SYN} & \left[\begin{array}{ll} \text{VAL} & \left[\text{SPR} \;\; \langle \;\rangle\right] \\ \text{GAP} & \langle \boxed{1} \rangle \end{array}\right] \\ \text{ARG-ST} & \boxed{\text{A}} \left\langle \boxed{1}, \ldots \right\rangle \end{array}\right] \right\rangle \end{array}\right]$$

We also introduced a new principle, which has the effect of passing GAP specifications from daughter to mother within headed phrase structures, while subtracting out any GAP elements that are bound within the phrase:

The GAP Principle

A local subtree Φ satisfies the GAP Principle with respect to a headed rule ρ if and only if Φ satisfies:



The value of STOP-GAP is assigned in the lexicon by the following defeasible constraint that is overridden by the lexical entries for adjectives like *easy* and *hard*:

$$\textit{lexeme} : \begin{bmatrix}\text{STOP-GAP} & / \langle\ \rangle\end{bmatrix}$$

$$\left\langle \text{easy}\ ,\ \begin{bmatrix}\textit{adj-lxm} & \\ \text{SYN} & \begin{bmatrix}\text{STOP-GAP} & \langle\ \boxed{1}\ \rangle\end{bmatrix} \\ \text{ARG-ST} & \left\langle \text{NP}_i\ ,\ \overset{\text{VP}}{\begin{bmatrix}\text{INF} & + \\ \text{GAP} & \langle\ \boxed{1}\text{NP}_i\ , ...\ \rangle\end{bmatrix}} \right\rangle\end{bmatrix} \right\rangle$$

And we added the following nondefeasible constraint on the type *l-rule*:

$$\textit{l-rule} : \begin{bmatrix}\text{INPUT} & \left\langle \text{X}\ ,\ \begin{bmatrix}\text{STOP-GAP} & \boxed{A}\end{bmatrix}\right\rangle \\ \text{OUTPUT} & \left\langle \text{Y}\ ,\ \begin{bmatrix}\text{STOP-GAP} & \boxed{A}\end{bmatrix}\right\rangle\end{bmatrix}$$

To handle the top of LDDs, we introduced lexical entries for *easy*-type adjectives and we also introduced one new grammar rule – the Head-Filler Rule, which licenses sentences with a 'topicalized' initial element:

Head-Filler Rule

$$[\textit{phrase}] \ \rightarrow\ \boxed{1}\begin{bmatrix}\text{GAP} & \langle\ \rangle\end{bmatrix} \quad \mathbf{H}\begin{bmatrix}\text{HEAD} & \begin{bmatrix}\textit{verb} & \\ \text{FORM} & \text{fin}\end{bmatrix} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix} \\ \text{STOP-GAP} & \langle\ \boxed{1}\ \rangle \\ \text{GAP} & \langle\ \boxed{1}\ \rangle\end{bmatrix}$$

To properly constrain the values of GAP and STOP-GAP throughout our trees, we made minor revisions to the 'initial symbol' clause of the definition of well-formed tree structure and to the Head-Specifier, Head-Modifier and Imperative Rules:

Φ is a Well-Formed Tree Structure according to G if and only if:

...

the label of Φ's root node satisfies the constraint:

$$\begin{bmatrix}\textit{phrase} & \\ \text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\textit{verb} & \\ \text{FORM} & \text{fin}\end{bmatrix} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix} \\ \text{GAP} & \langle\ \rangle\end{bmatrix}\end{bmatrix}$$

Head-Specifier Rule

$$\begin{bmatrix} \textit{phrase} & \\ \text{SPR} & \langle\ \rangle \end{bmatrix} \rightarrow \boxed{1} \quad \mathbf{H}\begin{bmatrix} \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{STOP-GAP} & \langle\ \rangle \end{bmatrix}$$

Head-Modifier Rule

$$[\textit{phrase}] \rightarrow \mathbf{H}\boxed{1}\begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{STOP-GAP} & \langle\ \rangle \end{bmatrix} \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle \boxed{1} \rangle \end{bmatrix}$$

Imperative Rule

$$\begin{bmatrix} \textit{phrase} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \end{bmatrix} \\ \text{GAP} & \boxed{A} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{dir} \\ \text{INDEX} & s \end{bmatrix} \end{bmatrix} \rightarrow \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{INF} & - \\ \text{FORM} & \text{base} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \text{NP}[\text{PER}\ \ \text{2nd}] \right\rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \\ \text{GAP} & \boxed{A} \end{bmatrix} \\ \text{SEM} & [\text{INDEX}\ \ s\ ] \end{bmatrix}$$

Finally, to deal with the Element Constraint (part of Ross's Coordinate Structure Constraint), we modified the Coordination Rule as follows:

Coordination Rule

$$\begin{bmatrix} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{GAP} & \boxed{A} \\ \text{IND} & s_0 \end{bmatrix} \rightarrow$$

$$\begin{bmatrix} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{GAP} & \boxed{A} \\ \text{IND} & s_1 \end{bmatrix} \ldots \begin{bmatrix} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{GAP} & \boxed{A} \\ \text{IND} & s_{n-1} \end{bmatrix} \begin{bmatrix} \text{HEAD} & \textit{conj} \\ \text{IND} & s_0 \\ \text{RESTR} & \langle[\text{ARGS}\ \langle s_1 \ldots s_n \rangle]\rangle \end{bmatrix} \begin{bmatrix} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{GAP} & \boxed{A} \\ \text{IND} & s_n \end{bmatrix}$$

14.9 Further Reading

Ross 1967 is probably the most influential work to date on the topic of long-distance dependencies. Chomsky (1973, 1977, 1986a) developed one of the most influential approaches to analyzing these constructions, using transformations. The treatment presented here is based loosely on that developed in Pollard and Sag 1994, which is compared with transformational approaches in Levine and Sag 2003. This analysis is unusual in not positing an empty category (a trace) in the position of the gap. Arguments for such a traceless analysis are discussed by Sag and Fodor (1994). Other nontransformational treatments are presented in Gazdar 1981, Kaplan and Zaenen 1989, Steedman 2000, and Bouma et al. 2001.

14.10 Problems

Problem 1: A Tree with a Gap
Draw a tree for (9b). Use abbreviations for the node labels, and show the value of GAP on all nodes. Show the value of STOP-GAP on any node where it is non-empty.

⚠ **Problem 2: Blocking Filled Gaps**
Examples (i) and (ii) are well-formed, but example (iii) is ungrammatical:

(i) Pat thinks that I rely on some sort of trick.
(ii) This mnemonic, Pat thinks that I rely on.
(iii) *This mnemonic, Pat thinks that I rely on some sort of trick.

Explain in detail why the mechanisms that license (i) and (ii) do not also permit (iii).

Problem 3: Subject Gaps
This problem is to make sure you understand how our analysis accounts for examples like (47).

A. Sketch the family of lexical sequences for *likes* that is the input to the Subject Extraction Lexical Rule.
B. Sketch the family of lexical sequences for *likes* that is the corresponding output to the Subject Extraction Lexical Rule.
C. Sketch the tree for the sentence in (47b). Use abbreviations for node labels, but show the value of GAP on all nodes and the value of STOP-GAP on any node where it is non-empty. You may abbreviate the structure over the NP *oysters on the half-shell* with a triangle.
D. Does our analysis correctly predict the contrast between (47b) and (i)?
(i) *Those candidates, I think likes oysters on the half-shell.
Explain why or why not.

Problem 4: Irish Complementizers
Consider the following example that shows the typical word order pattern of Modern Irish (data from McCloskey 1979):

(i) Shíl mé goN mbeadh sé ann.
thought I COMP would-be he there
'I thought that he would be there.'

Irish is a VSO language. One way of analyzing such languages is to posit a Head-Specifier-Complement Rule that introduces both kinds of dependents as sisters of the lexical head. In addition, we'll need a Head-Complement Rule that realizes only complements, and requires the head daughter to be [SPR $\langle\ \rangle$].

A. Formulate these two rules and show the structure for sentence (i). You may use abbreviations such as NP, S, etc., but be sure to show the value of COMPS and SPR on each node.

Now consider some further Irish data:

(ii) Dúirt mé gurL shíl mé goN mbeadh sé ann.
said I goN.PAST thought I COMP would-be he there
'I said that I thought that he would be there.'

(iii) an fear aL shíl mé aL bheadh ann
the man COMP thought I COMP would-be __ there
'the man that I thought would be there'

(iv) an fear aL dúirt mé aL shíl mé aL
the man COMP said I COMP thought I COMP
bheadh ann
would-be __ there
'the man that I said I thought would be there'

(v) an fear aL shíl goN mbeadh sé ann
[the man]$_j$ COMP thought __ COMP would-be he$_j$ there
'[the man]$_j$ that thought he$_j$ would be there'

(vi) an fear aL dúirt sé aL shíl goN
the man COMP said he COMP thought __ COMP
mbeadh sé ann
would-be he there
'the man that he said thought he would be there'

The complementizers *goN* and *aL* are in complementary distribution. That is, wherever *goN* is possible in these examples, *aL* is not, and vice versa.[9] Assume that both these elements are heads of CPs similar to those headed by *that* complementizers in English. If we then make the further assumption that LDDs in Irish work much as they do in English, we have all the tools we need to analyze the contrasts in (i)–(vi).[10]

B. Provide lexical entries for these two complementizers. [*Note: You may assume for purposes of this problem that the type comp-lxm which we proposed for English is applicable to Irish as well.*]

C. Show how your analysis successfully explains the distributional differences between the two complementizers. Be sure to cite the data given in the problem.

---

[9] For the purposes of this problem, you should ignore the difference between *gurL* and *goN*.

[10] Examples (iii)–(vi) involve relative clauses, which we have not discussed in much detail. Assume that the complementizers are the same whether they appear in relative clauses or in CP complements to verbs.

Problem 5: Nested Dependencies
We have made GAP a list-valued feature, which leaves open the possibility of multiple GAPs. This problem considers sentences which instantiate this possibility, such as (i) and (ii):

(i) Problems this involved, my friends on the East Coast are hard to talk to __ about __ .

(ii) Violins this well crafted, these sonatas are easy to play __ on __ .

A. Indicate which NP is interpreted as the filler for each of the gaps in (i) and (ii).

B. Draw a tree for sentence (i), indicating the value of GAP and STOP-GAP on every node. You do not need to include other features, though you should have a node label (e.g. VP, PP, etc.) on each node, and use tags and coindexing as appropriate. You may abbreviate the structure of the NPs *problems this involved* and *my friends on the East Coast* with triangles.

The PP complements of *talk* can actually appear in either order:

(iii) Dana tried to talk about it to everyone in the building.

(iv) Dana tried to talk to Leslie about this problem.

For the sake of this problem, we will assume that this is dealt with by allowing two distinct ARG-ST lists for *talk*: ⟨ NP , PP[about] , PP[to] ⟩ and ⟨ NP , PP[to] , PP[about] ⟩.

However, when we switch the order of the PPs in the multiple gap example, we get a sentence with a bizarre meaning (in which someone is talking to problems about people):

(v) Problems this involved, my friends on the East Coast are hard to talk about __ to __ .

C. Is this predicted by our analysis of LDDs? Why or why not?
[*Hint: Remember that the value of GAP is a list, and the order of the GAP list on phrasal nodes is determined by the GAP Principle.*]

Problem 6: Binding and LDDs
Assuming that reciprocals are [MODE ana], does our analysis of LDDs interact with the Binding Theory to predict that (i) should be grammatical? Why or why not?

(i) [Those people]$_i$ I tend to believe will tell [each other]$_i$ everything.

15

Variation in the English Auxiliary System

15.1 Introduction

English auxiliaries constitute a particularly interesting syntactic system, involving a small set of words that exhibit many intricate interactions and some fascinating idiosyncrasies. This system is peculiar to English, though many other languages have elements with intriguingly parallel properties.

English auxiliaries have changed considerably over the last thousand years or so, and their evolution has been well documented, making them a natural domain for studying syntactic change.[1] Change begins with variation, and the auxiliary system is also the locus of some fascinating differences among varieties of English. Variation is interesting in its own right, but studying it also helps us to ascertain which properties of our grammar we should formulate as or deduce from general principles, and which ones we should treat as essentially accidental.

In this chapter, we provide two examples of variation in English auxiliaries. The first example concerns the behavior of *have*, whose syntax is that of an auxiliary verb in some instances but not in others. The second example deals with a much studied phenomenon in the dialect known as African American Vernacular English (AAVE, for short). In both cases, we will explore how the variation might be handled within our theory. Our examples and discussion are intended only as samples of how data on variation might be relevant to work on syntactic theory. Syntactic variation is a topic worthy of a textbook in its own right, and we make no pretense of doing it justice here.

15.2 Auxiliary Behavior in the Main Verb *Have*

Our first example of variation in the English auxiliary system is one that occurs in a number of varieties of English, and can even vary for individual speakers, depending on the formality of the context: In some circumstances, certain uses of *have* as a main verb exhibit auxiliary-like behavior. The examples in (1) were judged acceptable by the speakers of British English we consulted: (Speakers of other varieties may well have heard such examples, even if they wouldn't produce them.)[2]

[1] Discussion of the history of the English auxiliary system, though an exciting topic, would take us too far afield.

[2] Our small survey on these data also included some speakers from Australia, New Zealand and India. These sentences were most uniformly accepted by the British speakers, however, and it is their judgments

(1) a. Have you any idea of the time?
b. I haven't a clue.
c. They said we had a problem, and we have.

This is not the first time we have seen a verb that is not a 'helping verb' – that is, one that is the sole verb in its clause – exhibiting some of the NICE properties. According to our analyses, the lexeme *be* must belong to the type *auxv-lxm*: finite forms of *be* (e.g. *is*, *were*) undergo the ADV_{pol}-Addition, Inversion, and Contraction Lexical Rules, and finite and non-finite forms of *be* undergo the Ellipsis Lexical Rule.[3]

In our grammar, the feature [AUX +] (associated with all instances of the type *auxv-lxm*) is associated with a certain range of properties, notably (i) the NICE properties, and (ii) not being a possible complement to the auxiliary *do*. In the dialects that allow main verb *have* to exhibit NICE properties, we can posit a lexical entry for *have* that takes an NP complement (rather than the VP complement of the auxiliary *have*), but which is still [AUX +]. That is, we posit an entry along the following lines:

(2)
$$\left\langle \text{have}\ , \begin{bmatrix} \textit{stv-lxm} & \\ \text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{AUX} & +\end{bmatrix}\end{bmatrix} \\ \text{ARG-ST} & \langle\ \text{NP}_i\ ,\ \text{NP}_j\ \rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{possess} \\ \text{SIT} & s \\ \text{POSSESSOR} & i \\ \text{POSSESSED} & j \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Many of the speakers who accept examples like (1) also permit their counterparts in (3):

(3) a. Do you have any idea of the time?
b. I don't have a clue.
c. They said we had a problem, and we do.

we will focus on here. We are grateful to Tim Baldwin, David Beaver, Eve Clark, Ann Copestake, Cathryn Donohue, Edward Flemming, Neal Glew, Alex Lascarides, Devyani Sharma, and Peter Sells for sharing their judgments.

[3]There are a few exceptions, which we will not attempt to deal with here. When *be* is used to denote activities over which the subject has clear voluntary control (meaning something close to 'act'), it can co-occur with the auxiliary *do*, as in Bob Dole's statement to supporters during the 1996 presidential campaign:

(i) You're not gonna get that tax break if you don't be quiet.

In addition, as first noted by Akmajian and Wasow (1974), *being* appears to be an exception to the Ellipsis Lexical Rule (for reasons that remain obscure):

(ii) They said we were being followed by a blue car, and we were (*being).

This suggests that, for these speakers, the entry in (2) is not quite right. Instead of stipulating [AUX +], it should allow either [AUX +] or [AUX −].[4] This will permit the verb to occur where [AUX +] is required (i.e. in the lexical rules involved in the NICE phenomena) as well as where [AUX −] is required (in the complement of auxiliary *do*). Thus, the differing behavior of main verb *have* across dialects can be captured with different specifications of one feature.

The situation is actually a bit more complex than this. Not all uses of *have* exhibit the NICE properties, even in the dialects that accept (1). Some examples are given in (4) and (5):

(4) a. They had a fit.
b.*They had not a fit.
c.*Had they a fit?
d.*They hadn't a fit.
e.*I said they would have a fit, and they had.

(5) a. Every day at lunch, Lou had a martini.
b.*Every day at lunch, Lou had not a martini.
c.*Every day at lunch, had Lou a martini?
d.*Every day at lunch, if Fran had a martini, Lou had, too.

There seems to be a semantic difference between those uses of *have* that exhibit NICE properties in the dialect in question and those that do not. Roughly, *have* means 'possess' in the NICE uses, but it assumes more specialized meanings in (4) and (5). To capture the correlation of syntactic behavior with different semantic interpretations, we need to postulate multiple lexical entries for *have*. In particular, the entry in (1) specifies that the semantic relation is possession. The main verb entries for *have* required for (4) and (5) specify different relations, and they are [AUX −].

Although the postulation of multiple entries for *have* might seem a priori undesirable, the fact that it has so many meanings would require separate entries, even if the meanings were not correlated with syntactic differences.

15.3 African American Vernacular English

The next case of variation we will examine concerns a dialect we will refer to as African American Vernacular English (which is also known as Black English or African American English). We will treat AAVE as a homogeneous dialect, just as we have been assuming

[4]We have been slightly evasive about what formal mechanism to use in allowing both values for AUX. The most parsimonious way to do this would be simply to underspecify the entry for this feature. Such an analysis would be consistent with everything in the text to this point. However, it would only work if *verb-lxm* had no default specification for the feature AUX, and this, in turn, would require that every nonauxiliary verb entry in the lexicon (that is, the overwhelming majority of verb entries) be individually marked [AUX −]. It seems more natural, therefore, to stay with the formulation given in Chapter 13, where there is a default [AUX / −] specification on the type *verb-lxm*, which is overridden by the specification on *auxv-lxm*. If we do this, we need some way for (2) to override the default and still allow both values for AUX. We could do this by modifying (2) to say either [AUX + |−] or ([AUX +]). Both of these options raise some technical questions for the formal theory of defaults in inheritance hierarchies, but these are beyond the scope of this text.

for simplicity that there is a homogeneous standard American English. There is, in fact, considerable variation within AAVE, but there are also many properties that are quite general to AAVE and legitimize treating it as one dialect.[5]

Before delving into the data, we need to digress briefly, to clear up some common misconceptions about AAVE. In the autumn of 1996, as we were writing the first draft of the present chapter, the school board in Oakland, California passed a resolution calling for the use of 'Ebonics' (another name for AAVE, favored particularly by afrocentric activists who wish to deny that AAVE is a variety of English) as a tool in the teaching of standard English to African American children. This event unleashed a storm of public outrage across the country. Much of the criticism was directed at the unfortunate wording of the Oakland resolution, and some of it concerned specific classroom practices; neither of these is relevant to the present text. A great deal of the outcry, however, concerned the linguistic status of AAVE, revealing widespread misunderstanding about the nature of language and grammar, even among many highly educated and sophisticated commentators.

Eldridge Cleaver, the former Black Panther leader, wrote in the *Los Angeles Times*, 'When I was growing up, what is now euphemistically being called Ebonics was accurately called bad English.' This statement – like numerous others that appeared in the press – reveals the powerful influence of prescriptivism. It presupposes that some varieties of English are correct and that others are incorrect. But what is the basis for attributions of correctness or 'badness' to a language variety – or for that matter to a particular utterance? In this text, we have appealed to native speaker intuitions of well-formedness to determine what is grammatical and what is not. This is clearly not the standard of correctness Cleaver had in mind, nor is it the one that prescriptivists employ. Rather, prescriptivism singles out one linguistic variety as the standard, and labels deviations from that standard as incorrect. The choice of the standard is determined by a variety of historical, social, and political factors, and it can be quite different from the language people actually use on a day-to-day basis. Investigating the factors determining what forms of language become standards is a fascinating topic, but it is not directly relevant to our concerns here.

No language or dialect is inherently 'good' or 'bad.' They can only be evaluated with respect to some objective. Our perspective is the study of language as a natural phenomenon. To say that one language is 'better' than another *per se* makes no more

[5]No two humans speak exactly the same way, or share exactly the same intuitions about every sentence. Indeed, individual speakers change the way they speak over time and under different circumstances. Hence the concepts of 'language' and 'dialect' are idealizations, based on undefined notions of similarity. They are nevertheless indispensable starting points for linguistic analysis. Moreover, providing precise characterizations of similarities and differences across varieties of speech could provide a scientific foundation for some of the intuitions underlying the way people individuate languages.

Linguists place little stock in the language/dialect distinction. One might attempt to characterize two varieties of speech as dialects of the same language if they are mutually intelligible, but even this intuitive characterization is problematic. Thus varieties that we have other reasons to think of as distinct languages (e.g. Swedish and Danish) are to a very large extent mutually intelligible. Moreover the relation of mutual intelligibility is not transitive: given three varieties A, B, and C, mutual intelligibility between A and B and between B and C does not guarantee that A and C are mutually intelligible. Thus one can talk at best about 'complexes' of language varieties (Hockett 1958) among which certain patterns of mutual intelligibility exist. Fortunately, these terminological issues do not need to be resolved in order for linguistic analysis to proceed.

sense to us than saying one atomic particle is better than another.[6] For our purposes, AAVE is just as legitimate an object of study as any other language variety.

Another widespread and more serious misconception is exemplified in a column by Brent Staples in the *New York Times*, who described AAVE as 'broken English.' Similarly, in a column by *The Washington Post*'s William Raspberry, AAVE is described as 'a language that has....no discernible rules.' Raspberry's piece recounts a fictional conversation with a taxicab driver, who says, 'you can say pretty much what you please [in Ebonics], as long as you're careful to throw in a lot of 'bes' and leave off final consonants.' If such claims were true, it would make linguistic investigation of AAVE trivial and pointless. But they are in fact patently false. Careful investigation of AAVE by linguists over the past four decades has shown that it has as much structure and system to it as any other linguistic variety. The differences between it and other varieties of English are quite systematic. One of those differences will be discussed briefly in this chapter, but see Green 2002 for a much more detailed survey.[7]

15.3.1 Missing Forms of *Be*

Examples like (6), though ungrammatical in standard American English (SAE, for short), are well-formed sentences of AAVE:

(6) a. Chris at home.
b. We angry with you.
c. You a genius!
d. They askin for help.

SAE requires a form of the copula (that is, a form of *be*) in each of these sentences, but, as noted in Chapter 11, many other languages (e.g. Russian and Hungarian) are like AAVE in permitting analogues of (6) without a copula. In fact, Ferguson (1971) noted a number of similarities between the environments in which copulas may be absent across languages, including AAVE.

Labov (1995) presents a number of arguments for an analysis of examples like (6) that involves deletion of a form of *be*. More specifically, he claims that such 'zero copula' sentences involve an extension of SAE contraction: *is* and *are* reduce to *'s* and *'re* in all dialects of English, and AAVE carries the process one step further, deleting the remaining consonants. He concludes that 'the copula is present in the underlying grammar, and is an object of cognition' (p. 46).

This sort of deletion analysis does not fit comfortably with the surface-oriented, constraint-based, strongly lexicalist grammar we have been developing in this text. To

---

[6]Of course, if the objective for evaluation of language varieties is getting ahead economically, then there can be little doubt that in contemporary American society, AAVE fares poorly (outside of some pockets of the entertainment industry). For this reason, virtually all American educators – including the Oakland School Board – recognize the need to teach standard English in the schools. Unfortunately, this point of agreement seems to have been lost in much of the discussion of 'Ebonics'.

[7]Some of our data on AAVE are taken from published sources, cited below. In addition, we are grateful to John Baugh, Erica Denham, Stacy Fambro, Lisa Green, Andrea Kortenhoven, Sonja Lanehart, Zakiyyah Langford, and Tracey Weldon for giving us their judgments on some examples. Where those consulted did not agree in their judgments, we have reported the majority response. For a more detailed presentation of the data, see Bender 2001.

reiterate what we said in Chapter 9, our conception of grammar is surface-oriented in that it provides a structure that is directly associated with the string of words that constitute each sentence. The ancillary structure that has to be computed expresses information that is straightforwardly derivable from properties of the words in the string. There are no operations that destructively modify any representations. We have localized most grammatical and semantic information within lexical entries that correspond directly to the words present in the sentence.

The only way to accommodate Labov's treatment of the zero copula within our theory would be to posit phonological or phonetic processes that are exempt from the conditions just quoted. But such a move would greatly weaken the force of our claims, at least in the absence of some principled method for demarcating the boundaries between syntax and morphology on the one hand, and phonetics and phonology on the other. If deletions can be freely reformulated as a phonetic processes, then a prohibition against syntactic deletion has no substance.

It would be interesting, therefore, to explore how the AAVE zero copula might be accounted for in the theory we have been developing, and to compare such an analysis to a deletion account. We will consider three possible analyses: a revised initial symbol, a new phrase structure rule, and silent forms of the copula. Somewhat surprisingly, given our general approach, the silent copula analysis is the only one consistent with known data. It is also the most similar in spirit of the three to Labov's account in that it posits an 'underlying' copula. Unlike Labov's deletion analysis, it represents a minimal departure from strict surface-orientation. It also gives better coverage: as shown in the next subsection, Labov's analysis makes strong claims which are falsified by the data.

15.3.2 Labov's Deletion Account

Like every good linguistic analysis, Labov's deletion account makes testable claims. He formalizes the claim that copula deletion is a continuation of the process of contraction in terms of two rules. The first produces contracted forms by deleting the initial vowel of *is* or *are*. The second produces copulaless forms by deleting the remaining consonant. The first rule creates the conditions necessary for the second to apply, and in fact is the only means of creating those conditions. This means that Labov's analysis predicts that copula absence should only be possible if contraction is.

At first it looks like this prediction is borne out. Labov lists several environments that disallow both contraction and deletion. These include nonfinite contexts (7), imperatives (8), and tag questions (9):[8]

(7) a. You got to be good, Rednall!
b.*You got to ϕ good, Rednall!

(8) a. Be nice to your mother!
b.*ϕ nice to your mother!

(9) a. It ain't a flower show, is it?
b.*It ain't a flower show, 's it?
c.*It ain't a flower show, ϕ it?

[8] In these and following examples, we will mark the spot where the copula is expected with ϕ.

However, further investigation turns up some environments where copula absence is possible, but contraction is not. These include complement and subject extraction.

Labov actually listed complement extraction as an environment that disallows both copula absence and contraction, giving the examples in (10):[9]

(10) a. I don't care what you are.
b.*I don't care what you ϕ.
c.*I don't care what you're.

However the existence of examples like (11) is problematic for Labov's analysis:

(11) a. How old you think his baby is?
b. How old you think his baby ϕ?
c.*How old you think his baby's?

In non-inverted contexts like this, sentence-final deletion is possible, but contraction is not.

The minimal data set in (12) illustrates a second kind of environment where deletion is possible but contraction isn't:

(12) a. Tha's the man they say is in love.
b. Tha's the man they say ϕ in love.
c.*Tha's the man they say's in love.

(Note that (12c) is bad only if the copula is fully contracted; it is okay if *say's* is given two syllables. Labov's account requires deletion of the vowel – that is, full contraction – as a prerequisite for copula absence.)

In both (11) and (12), the a-sentences with the full form of the copula and the b-sentences with no form of the copula are fine. The c-sentences, with the contracted copula, however, are unacceptable in AAVE. Since Labov's analysis proposes to derive the b-sentences from the c-sentences, these minimal sets provide counterexamples to his claims.

It would be possible to preserve Labov's analysis of other cases of copula absence and find a different mechanism for the problematic sentences. However, any such attempt would lack generality. If one mechanism, particularly one that does not rely on destructive transformations of strings, can account for all cases of copula absence, it is clearly to be preferred. The next three subsections consider three different possible syntactic analyses.

15.3.3 Initial Symbol Analysis

An alternative analysis of the missing copula in AAVE involves allowing the complements of *be* (whether they are VPs, NPs, PPs, or APs) to combine directly with NP subjects to

[9] Notice, however, that other examples of complement extraction are fine:

(i) How old your baby?
(ii) How tall that basketball player?
(iii) Where your car?
(iv) When your birthday?
(v) I don't know how old his baby.

We haven't included the ϕ here since it is unclear where it should go. See note 14.

make sentential constituents. In fact, our analysis of *be* already requires that its second argument have a nonempty SPR list. Remember, the SPR of *be* has to be identical to the SPR of its complement, so the complement always must have something on its SPR list.

Hence, unless something is added specifically to exclude them, the following expressions are all generated by our grammar as phrases that are [SPR ⟨ ⟩] and [COMPS ⟨ ⟩], differing in their part of speech types:

(13) a. It wild.
b. You in trouble.
c. Leslie the boss.
d. Somebody coming to dinner.
e. Jean interviewed by a reporter.

These strings are not well-formed stand-alone sentences in SAE, because the initial symbol of our grammar is:

(14)
$$\begin{bmatrix} \textit{phrase} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{fin} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

As noted earlier (Chapter 8), this says that a stand-alone sentence (or what linguists sometimes call a 'root sentence') must be headed by a verb in a finite form. None of the examples in (13) contains a finite verb, so they are clearly ruled out as SAE sentences. Moreover, it is worth noting that there are constructions (which we will not analyze here) that permit strings similar to those in (13) to appear as subordinate clauses, for example:

(15)
$$\text{With the cat} \left\{ \begin{array}{l} \text{away} \\ \text{in the kitchen} \\ \text{a prisoner} \\ \text{sleeping} \\ \text{locked up} \end{array} \right\}, \text{the mice will play.}$$

However, SAE does not permit the strings in (16) as independent sentences:

(16)
$$\text{*The cat} \left\{ \begin{array}{l} \text{away} \\ \text{in the kitchen} \\ \text{a prisoner} \\ \text{sleeping} \\ \text{locked up} \end{array} \right\}.$$

Thus we want to allow strings like those in (13) as well-formed constituents in SAE, but we need to restrict their distribution. In particular, these examples do not occur as root sentences because of the initial symbol (14). Similarly, they do not occur as complements of the complementizer *that* or of the verb *think* because both select for a complement that must be headed by a finite verb.

Returning now to AAVE, we could try to account for examples like (11) by modifying the initial symbol, allowing a wider range of feature structures to be root sentences. In particular, we could say that AAVE allows another alternative in addition to (14), namely, (17):

(17) $$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{PRED} & + \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix}$$

However, this analysis isn't up to expressing the observed generalizations about the construction in question. If we assume AAVE and SAE share most of the other lexical and grammatical features disscussed in this book, then the analysis incorrectly predicts that the missing copula is possible only in root sentences.[10] But examples like (6) and (13) can easily be embedded; sentences like (18) are perfectly normal in AAVE:

(18) a. If you ϕ alone, watch out!
b. The man she ϕ lookin for ain't here.

Here the strings *you alone* and *she lookin for* are in positions in which SAE would require a clause headed by a finite verb (*you* **are** *alone* and *she* **is** *looking for*), but AAVE permits the copula to be missing. It seems that the grammars of SAE and AAVE differ in that, where the former permits a finite clause headed by *is* or *are*, the latter allows a clause without the copula.

One might try, with considerable redundancy, to make similar changes to every finite-clause selecting environment (initial symbol, COMPS feature of clause-embedding verbs, relative clauses, etc.) in the grammar of AAVE. However, even this analysis would provide no account of cases where the predicate is involved in a long distance dependency:

(19) How old you think his baby ϕ?

No matter how it is analyzed, the NP *his baby* doesn't look like the clausal complement *think* is expecting. Some additional structure is required to 'promote' it to clausal status. The next two subsections explore two kinds of additional structure: a new phrase-structure rule and a silent form of *be*.

15.3.4 Phrase Structure Rule Analysis

We can capture the informal generalization that AAVE generally allows copulaless clauses wherever SAE allows finite clauses more precisely by introducing a new phrase structure rule for AAVE, rather than an additional initial symbol. Specifically, we can handle the missing copula in AAVE with a grammar rule that says that a sequence of a nominative NP followed by another phrase can function as a finite clause. The following rule does just this:

---

[10]Ferguson (1971) notes that some languages allow the copula to be missing only in main clauses, suggesting that there may be languages with initial symbols like (17).

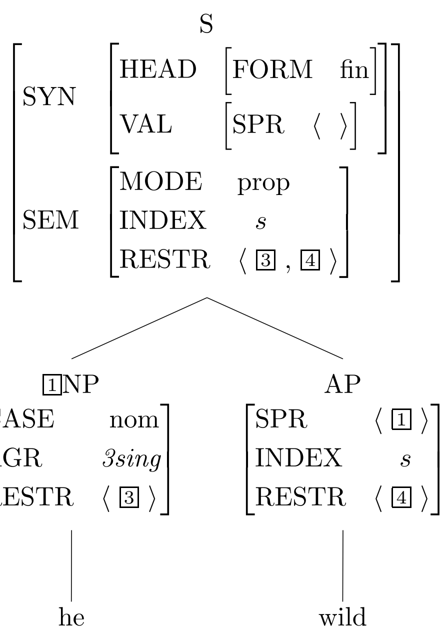
(20) Zero Copula Rule (AAVE)

$$\begin{bmatrix} \textit{phrase} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{fin} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & \boxed{2} \end{bmatrix} \end{bmatrix} \rightarrow \begin{array}{c} \boxed{1}\text{NP} \\ \begin{bmatrix} \text{CASE} & \text{nom} \\ \text{AGR} & \textit{non-1sing} \end{bmatrix} \end{array} \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{PRED} & + \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \boxed{1} \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & \boxed{2} \end{bmatrix} \end{bmatrix}$$

This rule says that wherever a finite clause is called for, AAVE allows a [PRED +] expression preceded by its nominative subject. It also says that the subject cannot be first-person singular, that is, *I* (see below). In addition, (20) fixes the INDEX and MODE values on the left hand side of the rule. (The Semantic Inheritance Principle applies only to headed rules.)

The Zero Copula Rule yields clauses that occur where finite clauses do, but which appear to be missing a verb. The resulting clauses will look as if they are missing a copula; and since the copula adds nothing semantically (except tense), they will also be interpreted as if they contained a copula. The effect of (20) is thus to license phrase structures like the following in AAVE:

(21)



We return now to the specification [AGR *non-1sing*] on the NP in the Zero Copula Rule. This was added because the first-person singular pronoun *I* cannot appear as the subject in the zero copula construction in AAVE:

(22) $*\text{I} \left\{ \begin{array}{l} \text{hungry} \\ \text{at home} \\ \text{the winner} \end{array} \right\}.$

This appears to be a rather singular exception: other related varieties with the zero copula construction (such as Jamaican Creole) do allow it with first-person singular subjects. It evidently requires a special stipulation on any available analysis of the phenomenon.

The type *non-1sing* that we posited for SAE in Chapter 4 has the subtypes *2sing* and *plural*. That is, it excludes both first- and third-person singular. This is clearly not the definition of the type that we want for AAVE, as sentences like (21), with third-person singular subjects and no copula, are perfectly acceptable. These data can all be accommodated by positing a hierarchy of agreement types for AAVE that diverges only slightly from its SAE conunterpart. The exact details of this alternative, however, are beyond the scope of this text.[11]

The Zero Copula Rule seems fairly promising so far, but, in fact, as we examine its empirical consequences in more detail, we see that it comes up short. The Zero Copula Rule predicts that copula absence is only possible when the subject is immediately adjacent to the predicate. This prediction is false: Long distance dependencies provide examples of copulaless clauses where the subject is not immediately adjacent to the predicate:

(23) a. How old they say his baby ϕ?
b. Tha's the man they say ϕ in love.

These examples cannot be generated through interaction of the Zero Copula Rule with the analyses of subject and complement extraction we presented in the previous chapter, because both crucially involve the selecting head. In the Zero Copula Rule analysis, there is no selecting head for the predicate. While the predicate is the selecting head for the subject, we would need a separate rule saying that a [PRED +] element with the same element on its GAP and SPR lists can serve as a finite clause. On the other hand, if there is a copula present (underlyingly) in sentences like (23), it is the selecting head for both the subject and the predicate, and the analyses of long-distance dependencies from Chapter 14 apply straightforwardly.

15.3.5 Silent Copula Analysis

Because our theory of grammar is based on the notion of constraint satisfaction, it does not allow any operations that destructively modify feature structures. So we can't posit a rule (syntactic or otherwise) that deletes the copula, nor would we want to. On the other hand, we can posit lexical sequences with the empty string as their phonology. Taking phonology as a sequence of sounds, one logical possibility is that the sequence is empty. Nothing in our theory so far rules out such a silent word.

Since the silent words in question are systematically related to other forms of *be*, we will write an inflectional rule to produce them. In essence, we treat the empty string as one more inflected form, alongside *am*, *is*, etc. This lexical rule is shown in (24):

[11] The marking on present tense verbs is variable in AAVE, but the variety arguably still shows a third-singular/non-third-singular distinction. A multiple inheritance hierarchy (see Chapter 16) could be used to provide top-level distinctions between and *non-1sing* and between and *non-3sing*.

(24) Silent *Be* Lexical Rule (AAVE)

$$\begin{bmatrix} \textit{i-rule} & \\ \text{INPUT} & \left\langle \text{be}\ ,\ \text{X} \right\rangle \\ \text{OUTPUT} & \left\langle \phi\ ,\ \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AGR} & \textit{non-1sing} \\ \text{FORM} & \text{fin} \\ \text{INV} & - \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

Since this rule is a subtype of *i-rule*, all of the information from the input (aside from the phonology) is preserved in the output. The agreement constraint looks particularly natural here, as adding agreement information is one common function of inflectional rules.

This rule, like the rule that produces the *am* form, applies to multiple lexemes which all share the same phonology. This is desirable, as *be* lexemes other than the copula can also appear in the silent form in AAVE. Examples include the *there* copula (25)[12] and the identity copula (26):

(25) a. There a car blocking my way.
b. I know there at least SOMEBODY happy about this.
c. There a big crowd of people outside.
d. There a book gone from my desk.

(26) A: Ain't Lew Alcindor Mohammed Ali?
B: You wrong. CASSIUS CLAY Mohammed Ali.

These facts are completely straightforward under the lexical rule account,[13] but they are problematic for the initial symbol and phrase structure rule accounts discussed above. The problem with (26) is that the second NP (*Mohammed Ali*) is not predicative (i.e. not [PRED +]). On (25), see Problem 2.

The lexical rule account can also account for the cases where copula absence is ungrammatical. The silent copula is [FORM fin], just like *is* (because the OUTPUT of the rule requires it to be). This means that the silent copula cannot be the complement of *to* (27) or serve as an imperative (28) (again, just like *is*):

(27) a.*You got to ϕ good, Rednall!
b.*You got to is good, Rednall!

(28) a.*ϕ nice to your mother!
b.*Is nice to your mother!

[12] For some AAVE speakers, the expletive subject in such sentences is *it*, rather than *there*, and a contracted form (*i's*) is required.

[13] On the other hand, the fact that some *be* lexemes don't undergo this rule is not problematic: Lexical exceptions to lexical rules are commonplace:

(i) *A good long sleep what you need most.
(ii) *It because of the cold weather.

Likewise, the specification [INV $-$] on the output of the rule ensures that the silent copula can't be used in tag questions (29) or other inversion environments like after *so* in (30):[14]

(29)*It ain't a flower show, ϕ it?

(30) a. I'm tired and so's my dog.
b.*I'm tired and so ϕ my dog.

15.3.6 Summary

In this section we have considered three different syntactic analyses of AAVE copula absence: a new initial symbol, a new phrase structure rule, and a lexical rule which produces lexical sequences with empty phonology. The data clearly support the third analysis, and so we are led to the conclusion that silent words exist. We do not make this move lightly, as silent words do complicate the relationship between the outwardly observable strings of syllables the structures we posit for them: In principle, a silent word could appear anywhere in the string, and a parser (human or computer) must always allow for that possibility. As discussed in Chapter 9, surface-orientation is an important aspect of the theory presented here, and one that contributes to making it a plausible competence grammar. We therefore depart from strict surface-orientation only when faced with data that admit no other analysis. Notice that in making this minimal departure, we are also preserving the two other characteristics of the grammar: The silent copula analysis is both constraint-based and strongly lexicalist.

Jackendoff (2002:131–132), in articulating a theory of grammar rather similar to ours, argues that lexical entries can be 'defective' in various ways. Some, like the dummy *it* (see Chapter 11), have empty semantics. He argues that exclamations like *ouch* are syntactically defective, in that they cannot combine with other words into phrases. A silent word is phonologically defective. We would expect any given language to have relatively few silent words, just as we would not expect many words to share any one audible phonology.

15.4 Summary

There is much more that could be said about variation within the English auxiliary system. Our superficial look at two cases of linguistic variation was meant only to demonstrate that the theory of grammar developed here can provide a natural analysis of such variation. In the case of the main verb *have*, careful analysis revealed a close link between its variable syntactic and semantic properties. In the case of AAVE, we found that the best analysis of copulaless sentences was a lexical rule that produces the silent form.[15]

[14]At first glance, it appears that the silent copula is allowed in other inversion environments, such as matrix *wh*-questions (i):

(i) Where he going?

However, further investigation shows that this environment is not necessarily an inversion environment in AAVE, as illustrated in (ii):

(ii) What they found there?

For further discussion, see Bender 2001.

[15]Jackendoff (2002:258) tentatively arrives at a similar conclusion for languages like Russian and Hebrew.

Variation across dialects is often accompanied by variation within the speech of individuals, as is the case with the AAVE copula. This variation is usually related to a variety of social factors and provides fruitful ground for the study of sociolinguistics. By providing precise syntactic analyses of the variant forms in a constraint-based framework, we open the door to a more detailed inquiry investigating the interaction of social and grammatical constraints.

15.5 Further Reading

Much of the pioneering work on AAVE was done by William Labov and his collaborators and students. Among the most important works are Labov et al. 1968; Labov 1969, 1972; and Baugh 1983. For more on AAVE, see Rickford and Rickford 2000 and Green 2002 (two recent accessible books on AAVE) and Mufwene et al. 1998 (a collection of important theoretical papers). Labov (1995) offers a clear presentation of a different analysis of some of the same phenomena discussed in this chapter. The analysis presented here is developed more fully in Bender 2001, Chapter 3. An early discussion of the dialect variation with respect to the main verb *have* can be found in Chomsky 1975, which was written in 1955.

15.6 Problems

Problem 1: AAVE Silent Copula and LDDs
Give the word structure for the silent copula in each of the following sentences:

(i) I don't know how old his baby ϕ.
(ii) Tha's the man they say ϕ in love.

[*Hint: In (i) the complement of the silent copula is gapped and in (ii) its subject is gapped.*]

Problem 2: Silent Copula Existentials in AAVE
In Section 15.3.5 above, we state that sentences like (i)–(iv) are problematic for the initial symbol and phrase structure rule analyses.

(i) There a car blocking my way.
(ii) I know there at least SOMEBODY happy about this.
(iii) There a big crowd of people outside.
(iv) There a book gone from my desk.

A. Draw a phrase structure for (i) under the Silent *Be* Lexical Rule analysis. You may use abbreviations for the nodes.
B. Discuss why this example is problematic for the initial symbol analysis.
C. Discuss why this example is problematic for the phrase structure rule analysis.
D. Extra credit: Could either analysis be modified or extended to account for these examples? How or why not?

Problem 3: AAVE Invariant *Be*

The use of *be* in environments where SAE would require a finite form is one of the best known features of AAVE. (This is what Raspberry was ridiculing in the column quoted earlier in the chapter). This so-called 'invariant *be*' conveys a special meaning, usually characterized as habitual. Thus, for example, (i) means roughly that the students are usually or always in the hall when the bell rings.

(i) The students be in the hall when the bell ring. (from Green 1993)

The goal of this problem is to determine how this use of *be* should be analyzed in our theory. Specifically, provide answers to the following three questions, and give arguments for your answers:

A. Should the 'invariant *be*' be analyzed as a distinct lexical entry, or can it be assimilated to one of the entries for *be* we have already posited?

[*Hint: Consider the semantics.*]

B. Is the 'invariant *be*' a verb, and, if so, is it finite?

C. Consider the following additional data:

(ii) *The students be not in the hall when the bell ring.

(iii) *Be the students in the hall when the bell ring?

(iv) *The students ben't in the hall when the bell ring.

(v) *Bob be eating when the news come on, and Sue be, too.

What do examples in (ii)-(v) show about the AUX value of 'invariant *be*'?

16

Sign-Based Construction Grammar

16.1 Taking Stock

Let us examine the road we have traveled in the preceding chapters. We began with the inadequacy of simple context-free grammars, given their inability to express the notion of headedness, their inability to express cross-categorial generalizations, and the redundancy inherent in their descriptions of English in particular, and natural languages in general. We set out to address these inadequacies by modifying CFGs. First we introduced the notion of feature structure, to model not only syntactic categories but also a variety of other constructs that became necessary as we developed our analyses (lexemes, words, agr-cats, etc.). We refined our feature structure analysis by adding the notion of a linguistic type, which we used to declare which features are appropriate for particular kinds of linguistic entities. We observed that linguistic entities (and the constraints they obey) often come in families, that is, groupings of distinct entities that share common properties. Based on this observation, we organized our linguistic types into a hierarchy.

After examining and analyzing a host of grammatical phenomena, however, we find that our system has not strayed very far from its CFG origins. We continue to specify well-formed tree structures in essentially the CFG fashion: we specify a lexicon plus a set of grammar rules, and we combine the information from these two sources in terms of a third, namely, the set of conditions that constitute our definition of well-formed tree structures. The lexicon is more elaborate than it would be in a simple CFG, with its hierarchical organization and its lexical rules. Likewise, we have enriched the definition of phrasal licensing (part of our well-formedness definitions) to incorporate various general principles, such as the Head Feature Principle. The grammar rules, on the other hand, have become rather simple and schematic.

In this chapter, we consider further modifications to the theory of grammar we are working with, relating the ideas developed in Chapters 5–14 to the kind of grammar that is used in much current research. To get to there from here, we will make three main modifications: In Section 16.2, we will refine our type hierarchy to allow individual entities to be classified along multiple dimensions (e.g. part of speech and valence patterns). In Section 16.3, we will recast both words and phrases as *signs*, or pairings of form and meaning. In Section 16.4 we will recast our grammar rules as feature structures similar to the conception of lexical rules developed in Chapter 8. These will relate phrases to the expressions (words and phrases) that make them up. Treating the grammar rules as feature structures will allow us to arrange them into a hierarchy (just like lexeme types and lexical rules) and to express many of our general principles as constraints that apply to supertypes in that hierarchy.

16.2 Multiple Inheritance Hierarchies

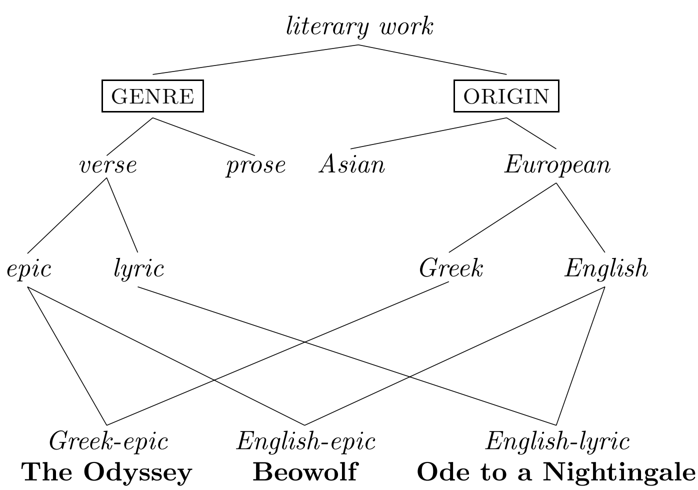
In Chapter 12, we introduced a distinction between raising and control elements. As we noted there, the distinction cuts across part-of-speech distinctions, for example, the distinction between verbs and adjectives. In fact, cross-cutting generalizations of this kind are not at all uncommon in human languages, and our theory of grammar should be able to express them.

However, the theory of lexical types we presented in Chapter 8 doesn't go far enough in expressing cross-cutting generalizations. Because the hierarchy of lexemes is one-dimensional (a tree), we posit distinct types (e.g. *adj-lxm* vs. *verb-lxm*) whenever a distinction must be made. But this means that if these types each have a subtype with a common property – for example, the type of subject-raising verb (*continue*) and that of the subject-raising adjective (*likely*) – then we must state the constraint that defines subject-raising twice – once on each of the distinct types. Our one-dimensional (or 'single-mother') inheritance hierarchies thus fail to express certain generalizations that we find in natural language.

There is a simple solution to this problem. Suppose that we allow a type to be simultaneously subdivided in two (or more) independent ways. That is, suppose we allow independent partitions of certain types into subtypes, so that each maximal (leaf) type must have a supertype in each partition. Intuitively, the partitions correspond to distinct informational dimensions. For example, we can distinguish a dimension built up from (types based on) part-of-speech distinctions from one that is based on distinctions of argument structure. This allows the contrast of, say, verb versus adjective to be independent of the contrast between strict-intransitive versus subject-raising versus subject-control, and so on.

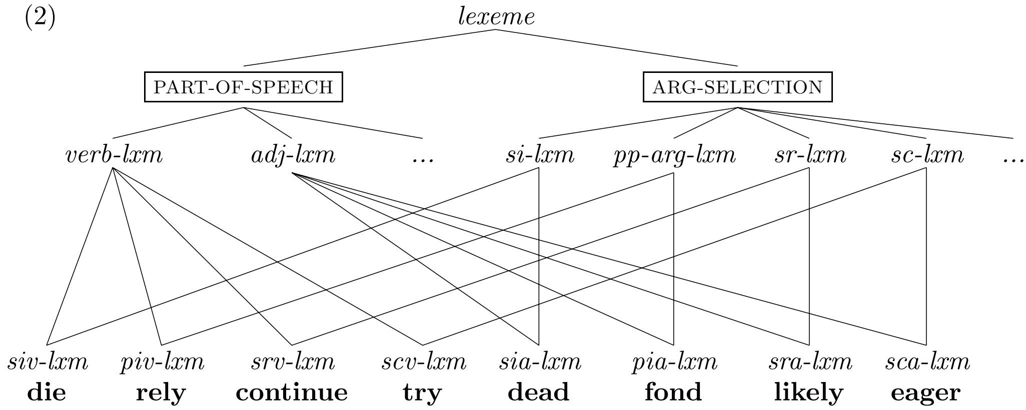
This kind of classification is called a MULTIPLE INHERITANCE HIERARCHY. Such hierarchies are commonly used in artificial intelligence and (more broadly within computer science) in the object-oriented programming paradigm to organize multiple dimensions of information about objects in particular knowledge domains. To take a non-AI example, literary works might be organized in terms of types that form two independent hierarchical dimensions of classification, as shown in (1):

(1)



In (2), we illustrate one way our lexeme hierarchy could be revised using multiple inheritance to express the cross-cutting generalizations our current hierarchy fails to capture:

(2)



Several comments about this hierarchy are in order. First, we are using some new abbreviations for type names, so we have spelled out all of the names in (3):

(3) a. *si-lxm*: *strict-intransitive-lexeme*
b. *pp-arg-lxm*: *PP-argument-lexeme*
c. *sr-lxm*: *subject-raising-lexeme*
d. *sc-lxm*: *subject-control-lexeme*
e. *siv*: *strict-intransitive-verb-lexeme*
f. *piv*: *PP-intransitive-verb-lexeme*
g. *srv*: *subject-raising-verb-lexeme*
h. *scv*: *subject-control-verb-lexeme*
i. *sia*: *strict-intransitive-adjective-lexeme*
j. *pia*: *PP-intransitive-adjective-lexeme*
k. *sra*: *subject-raising-adjective-lexeme*
l. *sca*: *subject-control-adjective-lexeme*

Second, the ARG-SELECTION types (*si-lxm*, *pp-arg-lxm*, etc.) must be subject to type constraints that express the ARG-ST properties that are shared by elements of the particular verbal and adjectival subtypes. These constraints can be formulated as in (4):

(4) a. $\textit{si-lxm} : \begin{bmatrix}\text{ARG-ST} & \langle\ \text{X}\ \rangle\end{bmatrix}$

b. $\textit{pp-arg-lxm} : \begin{bmatrix}\text{ARG-ST} & \langle\ \text{X}\ ,\ \text{PP}\ \rangle\end{bmatrix}$

c. $\textit{sr-lxm} : \begin{bmatrix}\text{ARG-ST} & \left\langle \boxed{1}\ , \begin{bmatrix}\text{SPR} & \langle\ \boxed{1}\ \rangle\end{bmatrix}\right\rangle\end{bmatrix}$

d. $\textit{sc-lxm} : \begin{bmatrix}\text{ARG-ST} & \left\langle \text{NP}_i\ , \begin{bmatrix}\text{SPR} & \langle\ \text{NP}_i\ \rangle\end{bmatrix}\right\rangle\end{bmatrix}$

The PART-OF-SPEECH types can remain subject to our earlier constraints, namely the following:

(5) a.

$$\textit{verb-lxm} : \left[\begin{array}{ll} \text{SYN} & \left[\text{HEAD} \left[\begin{array}{ll} \textit{verb} & \\ \text{PRED} & - \\ \text{INF} & / - \\ \text{AUX} & / - \\ \text{POL} & - \end{array}\right]\right] \\ \text{ARG-ST} & \left\langle \left[\begin{array}{ll} \text{HEAD} & \textit{nominal} \\ \text{VAL} & \left[\begin{array}{ll}\text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle\end{array}\right]\end{array}\right], \ldots \right\rangle \\ \text{SEM} & \left[\text{MODE} \quad \text{prop}\right]\end{array}\right]$$

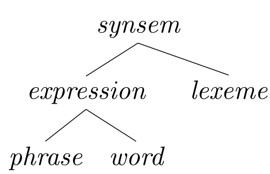
b.

$$\textit{adj-lxm} : \left[\begin{array}{ll} \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \textit{adj} \\ \text{VAL} & \left[\begin{array}{ll}\text{SPR} & \langle\ \text{X}\ \rangle \\ \text{MOD} & \langle\ [\text{HEAD} \quad \textit{noun}]\ \rangle\end{array}\right]\end{array}\right] \\ \text{ARG-ST} & \left\langle \left[\begin{array}{ll} \text{HEAD} & \textit{nominal} \\ \text{VAL} & \left[\begin{array}{ll}\text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle\end{array}\right]\end{array}\right], \ldots \right\rangle \\ \text{SEM} & \left[\text{MODE} \quad \text{prop}\right]\end{array}\right]$$

Given the similarity of these constraints, of course, we might want to further revise the system so that the specified ARG-ST and SEM constraints are associated with a common supertype. Our purpose here is not to make a definitive proposal for the analysis of the English lexeme hierarchy,[1] but rather to show that multiple inheritance hierarchies can be used to factor cross-cutting generalizations of words into succinct constraints that are associated with cross-classifying types.

We can use multiple inheritance hierarchies to solve another problem with our earlier analysis, where the high-level organization of the subtypes of *synsem* was assumed to be as shown in (6):

(6)



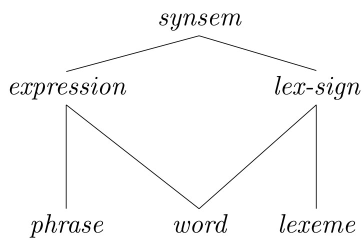
The types *phrase* and *word* are grouped together as subtypes of *expression*, i.e. the type of linguistic entity that can be a daughter of a phrase. On the other hand, *word* and *lexeme* share the property of being linguistic entities for which the feature ARG-ST is

[1]Such a proposal might also involve separating the distinction between constant and inflecting lexemes into a separate dimension.

appropriate. In the hierarchy in (6), however, there is no immediate supertype of *word* and *lexeme*. The grammar developed in Chapters 5–14 is thus missing the generalization that words and lexemes share common properties that phrases do not share.

This defect also can be remedied by organizing these types into a multiple inheritance hierarchy like (7):

(7)



All *synsem*s are specified for the features SYN and SEM, but only *lex-sign*s are specified for ARG-ST in addition. In the next section, we will revise our notions of *word*, *phrase* and *lexeme* slightly and see that they have one further feature in common.

16.3 Words and Phrases as Signs

The renowned Swiss linguist Ferdinand de Saussure (1857–1913) conceived of language as a system of what he called ‘signs’. A sign, according to Saussure, is a conventional association of a form with a meaning.[2] The sign-based conception of language is attractive in that it lets us put all linguistic expressions – words and phrases alike – on an equal footing.[3] Phrases like *love*, *Kim loves Sandy*, *Go away*, *to continue to like anchovies*, and so forth can all be viewed as associations of sound and meaning that are in accord with the constraints of English grammar, while *l’amour*, *Kim aime Sandy*, *Allez-vous en*, and *continuer d’aimer les anchois* are phrasal signs of French – different forms associated with the same meanings – and likewise for the corresponding phrasal signs of Inuktikut, Tongan, or any other natural language.

How, then, can we relate the grammar we have developed to the Saussurean conception? To begin with, observe that our current notion of a lexical entry is very much like a sign, in that lexical entries specify a form and a meaning. (Our lexical entries are richer than signs, though, in that they also specify syntactic information.) On the other hand, our current types *lexeme* and *word* are not signs: they don’t include any information about the phonological or orthographic form. In order to talk about (lexical) signs other than the basic lexical entries, we had to introduce the construct ‘lexical sequence’, whose sole purpose was to pair a form with a *lexeme* or *word*. We would have liked to say that inflectional rules relate ‘lexemes’ to ‘words’, but we could not, because lexical rules clearly modify phonological information, but neither lexemes nor words include phonological information in our grammar. So instead we ended up somewhat awkwardly speaking of inflectional rules as relating lexical sequences whose second member is of type *lexeme* to lexical sequences whose second member is of type *word*.

[2] Or, as Saussure would have it, an association of a mental representation of sound with a mental representation of meaning.

[3] Though Saussure did not. His conception of a linguistic system seems to have lacked the notions of recursivity and compositionality that modern work on grammatical theory takes for granted.

It would be a small but significant move to revise our notion of *word* along Saussurean lines, thereby eliminating the terminological tension between lexeme/word on the one hand and lexical sequence on the other. Suppose that we reorganized things so that the features appropriate for the type *word* were PHON(OLOGY) and SEM(ANTICS). We could then model the word *Kim* as follows:

(8)
$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \text{ Kim } \rangle \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{SIT} & s \\ \text{NAME} & \text{Kim} \\ \text{NAMED} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

This would be true to Saussure's conception, but it contains none of the SYN or ARG-ST information that has been central to our analyses of a variety of complex grammatical phenomena.

Of course, it requires only a small modification to include the syntactic information that is missing from (8). Introducing the SYN and ARG-ST attributes gives us an 'augmented' Saussurean sign that looks like (9):

(9)
$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \text{ Kim } \rangle \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \textit{3sing} \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \langle \ \rangle \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{SIT} & s \\ \text{NAME} & \text{Kim} \\ \text{NAMED} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

This is in essence the format in which the lexical analyses presented in this text have appeared in much of the published literature on sign-based grammar.

Notice that the PHON value in (9) is a list. This not only provides a uniform representation for words (recall our analysis of *kick the bucket* in Chapter 11), but also allows us to generalize the notion of sign to include phrases, where the PHON value will usually be a list of more than one form.

In treating phrases as signs, our grammar will characterize feature structures like the following, where PHON, SYN, and SEM (but not ARG-ST) values are specified:

(10)
$$\begin{bmatrix} \textit{phrase} & \\ \text{PHON} & \langle \text{ Kim , walks } \rangle \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{fin} \end{bmatrix} \\ \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{Kim} \\ \text{NAMED} & i \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{walk} \\ \text{SIT} & s \\ \text{WALKER} & i \end{bmatrix}, \ldots \right\rangle \end{bmatrix} \end{bmatrix}$$

This unfamiliar representation augments the Saussurean conception of sign (which he discussed only in terms of lexical expressions). Phrasal signs obey grammatical constraints that interact so as to correlate form sequences, e.g. ⟨*Kim, walks*⟩ with their meanings (e.g. the proposition that 'there is a walking situation *s* in which someone named *Kim* is walking, with additional temporal constraints contributed by the verb's tense').

In the next section, we will explore how the sign-based conception of grammar allows us to treat lexical rules and grammar rules in a parallel fashion, and further how it allows us to use type constraints, a key device we use for formulating lexical generalizations, to express cross-classifying generalizations about phrasal constructions, as well.

The ideas presented in this section are summarized in (11), where the type *synsem* has been renamed '*sign*', as *sign*s include phonological, as well as syntactic and semantic information:

(11)

TYPE	FEATURES/VALUE TYPES	IST
sign	$\begin{bmatrix} \text{PHON} & \textit{list(form)} \\ \text{SYN} & \textit{syn-cat} \\ \text{SEM} & \textit{sem-cat} \end{bmatrix}$	*feat-struc*
expression		*sign*
lex-sign	$\begin{bmatrix} \text{ARG-ST} & \textit{list(expression)} \end{bmatrix}$	*sign*
phrase		*expression*
word		*expression* & *lex-sign*
lexeme		*lex-sign*

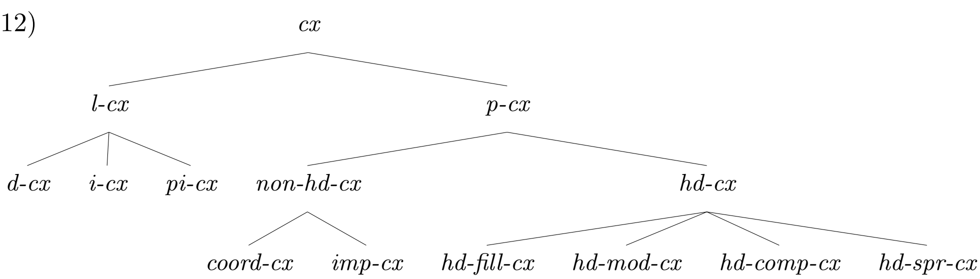
16.4 Constructions

Thus far, we have recast our notions of word, lexeme, and phrase in Saussurean terms by modeling his notion of 'sign' in terms of feature structures. Lexical rules are already expressed in terms of feature structures, and there is ample motivation for treating grammar rules the same way. In particular, there are common properties that many, but not all grammar rules share. If we model grammar rules as feature structures, we can organize them into a hierarchy that classifies them according to their various common properties. For example, each grammar rule is either a headed rule, in which case it obeys a particular set of constraints (the Head Feature Principle, the GAP Principle, etc.), or else it is a nonheaded rule (e.g. the Coordination Rule or the Imperative Rule). The headed rules in turn come in various subvarieties, each subject to its own particular constraints.

If we model grammar rules in terms of typed feature structures, then we can treat principles like the Semantic Compositionality Principle and the Head Feature Principle as type constraints (rather than as clauses in the definition of well-formed structure). In addition, we can use defeasible constraints of the sort deployed elsewhere to express the default nature of certain constraints on grammar rules. One such case is the Valence Principle, which requires that a head daughter's VAL specification be identified with that of its mother only when no specific rule contradicts it. For example, in the Head-Specifier Rule, which requires that the mother's and head daughter's SPR values be distinct, only the COMPS and MOD values of head daughter and mother are shared. Defeasible constraints on grammar rules thus work the same way as defeasible constraints on typed feature structures, and an adequate theory of grammar rules should be able to elucidate this fact.

Our goal, then, is to fold the grammar rules into our theory of feature structures, as we have already done in the case of lexical rules, which are modeled as feature structures (of type *l-rule*) specifying values for the features INPUT and OUTPUT. Suppose, however, that we discard the procedural metaphor these feature names embody. (After all, the well-formed structures are defined in terms of constraint satisfaction, not operations that destructively modify things.) We propose to replace the notion 'rule' with that of 'construction', using this for both lexical and phrasal analysis. That is, we may posit a type *construction* (*cx*) whose appropriate features are MOTHER and DTRS (DAUGHTERS) and whose immediate subtypes are *lexical-construction* (*l-cx*) and *phrasal-construction* (*p-cx*). All the rules we have discussed in this book can now be recast in terms of subtypes of these last two types, as sketched in (12):[4]

(12)



[4]Here we abbreviate as follows:

cx	*construction*	*l-cx*	*lexical-construction*
d-cx	*derivational-construction*	*i-cx*	*inflectional-construction*
pi-cx	*postinflectional-construction*	*p-cx*	*phrasal-construction*
non-hd-cx	*non-headed-construction*	*hd-cx*	*headed-construction*
coord-cx	*coordinate-construction*	*imp-cx*	*imperative-construction*
hd-fill-cx	*head-filler-construction*	*hd-comp-cx*	*head-complement-construction*
hd-spr-cx	*head-specifier-construction*	*hd-mod-cx*	*head-modifier-construction*

Before proceeding, we need to specify (just as we have done for all types we have considered in earlier chapters) which features are appropriate for each type, what kind of values these features take, and what constraints instances of these types must obey. This information is provided in (13) for the most general types in (12):

(13)

TYPE	FEATURES/VALUE TYPES	IST
cx	$\begin{bmatrix}\text{MOTHER} & \textit{sign} \\ \text{DTRS} & \textit{list}(\textit{sign})\end{bmatrix}$	*feat-struc*
l-cx	$\begin{bmatrix}\text{MOTHER} & \textit{lex-sign} \\ \text{DTRS} & \langle\ \textit{lex-sign}\ \rangle\end{bmatrix}$	*cx*
p-cx	$\begin{bmatrix}\text{MOTHER} & \textit{phrase} \\ \text{DTRS} & \textit{list}(\textit{expression})\end{bmatrix}$	*cx*

Very little has changed in the analysis of lexical constructions. The main difference is that we no longer need the awkward concept of lexical sequences. The mother (in the old terminology, output) and daughter (input) of a lexical construction are both *lex-sign*s. Similarly, the mother and daughters of phrasal constructions are all *expression*s. In the next section, we will discuss the details of phrasal constructions. First, however, we will revise the definition of well-formed structures.

In the grammar of Chapters 5–14, well-formed structures were trees, with syntactic and semantic information on each non-terminal node and phonological information on the leaves. In some sense, however, the only important information in each of these trees is the phonology on the leaves and the syntactic and semantic information on the mother node. The mother node's semantics represents the meaning of the whole expression (recall that the RESTR list of the mother will include all of the predications contributed by the daughters). Its syntax gives all the information necessary to determine what larger syntactic contexts the expression can occur in. The rest of the information (the *synsem*s labeling the intermediate nodes) is merely a record, on this view, of the pieces that were used to construct the whole.

The tree structure also served to connect the syntax and semantic information on the topmost node to the phonological information encoded in the leaves. Now that *phrase*s are *sign*s, and as such include phonological information, they can be cut free of the tree. This allows us to simplify the definition of well-formed structures considerably.

Before presenting the new definition, we need to introduce some terminological distinctions. As before, the components of our grammar are all descriptions, expressed now more uniformly as constraints. These descriptions systematize a set of fully specified feature structure models. In the grammar of Chapter 5-14, there were lexical rules and phrase structure rules (both descriptions) that helped license tree structures (models). Now we have lexical and phrasal CONSTRUCTIONS (descriptions) and INSTANTIATIONS of constructions (models). It will also be convenient to use the term CONSTRUCT to refer to the MOTHER value of any such instantiation.

With that background, here is the backbone of our revised well-formed structure definition:

(14) Φ is a Well-Formed Structure according to a grammar G if and only if:
1. there is some construction C in G, and
2. there is a feature structure I that is an instantiation of C, such that Φ is the value of the MOTHER feature of I.

As we will see in the next section, most of the principles of our grammar can be stated as constraints on construction types.[5] As such, they are implicitly included in the definition in (14).

The grammar thus licenses a feature structure of type *phrase* whose PHON value is ⟨ *ate* , *a* , *pizza* ⟩ because there is a feature structure instantiating the head-complement construction that has that feature structure as its MOTHER value. This phrasal construct satisfies the following description:

(15)
$$\begin{bmatrix} \textit{phrase} & \\ \text{PHON} & \langle \text{ ate , a , pizza } \rangle \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{fin} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \text{ NP } \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix} \\ \text{GAP} & \langle\ \rangle \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{eat} \\ \text{SIT} & s \\ \text{EATER} & i \\ \text{EATEN} & j \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{a} \\ \text{BV} & j \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{pizza} \\ \text{INST} & j \end{bmatrix}, \ldots \right\rangle \end{bmatrix} \end{bmatrix}$$

We may say that the head-complement construction allows this phrase as a construct. Similarly, the grammar licenses a feature structure of type *lexeme* whose PHON value is ⟨*driver*⟩, because there is a lexical construction (the one corresponding to the Agent Nominalization Lexical Rule in our earlier grammar) that has the feature structure as its MOTHER value. This lexical construct satisfies this description:

[5] The exception is the Binding Theory, which will require minor revision, given that the definition of ‘outrank’ in Chapter 7 makes reference to a tree configurational notion (domination). Once the notion of ‘exempt’ anaphor (see Problem 3 of Chapter 7) is properly incorporated into our grammar, however, ‘outrank’ can be defined without appeal to domination. We leave the details of this revision as an exercise for the reader.

(16)

$$\left[\begin{array}{ll} \textit{lexeme} & \\ \text{PHON} & \langle \text{driver} \rangle \\ \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{l} \textit{noun} \\ \text{AGR}\ [\text{PER} \quad \text{3rd}] \end{array}\right] \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \langle \text{DP} \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle \end{array}\right] \\ \text{GAP} & \langle\ \rangle \end{array}\right] \\ \text{SEM} & \left[\begin{array}{ll} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \left[\begin{array}{ll} \text{RELN} & \textbf{drive} \\ \text{SIT} & s \\ \text{DRIVER} & i \end{array}\right] \right\rangle \end{array}\right] \end{array}\right]$$

It is also convenient sometimes to speak loosely of the PHON value of a sign as being a construct, e.g. to speak of *driver* as a lexical construct and *ate a pizza* as a phrasal construct.

Finally, we need to distinguish those signs that can stand alone as utterances. That is we still need a 'Root Condition': the stand-alone structures licensed by G are those that satisfy (17) in addition to the constraints in (14):

(17)

$$\left[\begin{array}{ll} \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll} \textit{verb} & \\ \text{FORM} & \text{fin} \end{array}\right] \\ \text{VAL} & \left[\begin{array}{ll} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \end{array}\right] \\ \text{GAP} & \langle\ \rangle \end{array}\right] \end{array}\right]$$

16.5 Phrasal Constructions of Our Grammar

Let us now consider the various kinds of phrasal constructions in more detail. The headed constructions are those that make reference to a distinguished daughter, whose PHON value may be ordered first or last in the phrasal construct. This is easily modeled by declaring a new feature HEAD-DAUGHTER (HD-DTR) to be appropriate for feature structures of type *hd-cx*, as in (18):

(18)

TYPE	FEATURES/VALUE TYPES	IST
hd-cx	[HD-DTR *sign*]	*cx*

Particular subtypes of *hd-cx* must obey constraints identifying either the first daughter or the last as the head daughter. These constraints then interact with the following general principle that establishes the correspondence between the order of elements on the DTRS list and their linearization in the construct:

(19) Principle of Order

$$cx: \left[\begin{array}{ll} \text{MOTHER} & [\text{PHON}\ \boxed{\text{A1}} \oplus ... \oplus \boxed{\text{An}}\ ] \\ \text{DTRS} & \langle\ [\text{PHON}\ \boxed{\text{A1}}]\ ,\ ...\ ,\ [\text{PHON}\ \boxed{\text{An}}]\ \rangle \end{array}\right]$$

The Semantic Compositionality Principle can be reformulated in constructional terms as follows:

(20) Semantic Compositionality Principle

$$cx: \left[\begin{array}{ll} \text{MOTHER} & [\text{SEM}\ [\text{RESTR}\ \boxed{\text{A1}} \oplus ... \oplus \boxed{\text{An}}\ ]] \\ \text{DTRS} & \langle\ [\text{SEM}\ [\text{RESTR}\ \boxed{\text{A1}}]]\ ,\ ...\ ,\ [\text{SEM}\ [\text{RESTR}\ \boxed{\text{An}}]]\ \rangle \end{array}\right]$$

This is just a recasting of the version in Appendix A and is simplified in not taking constructional meaning into account.[6]

There are several of our grammatical principles that were part of the definition of phrasal licensing (see Appendix A), which can now be recast as constraints on the type *hd-cx*. These include the Head Feature Principle, the Semantic Inheritance Principle, the Valence Principle, and the GAP Principle, which may be formulated as follows:

(21) Head Feature Principle

$$hd\text{-}cx: \left[\begin{array}{ll} \text{MOTHER} & [\text{SYN}\ \ [\text{HEAD}\ \ \boxed{1}]] \\ \text{HD-DTR} & [\text{SYN}\ \ [\text{HEAD}\ \ \boxed{1}]] \end{array}\right]$$

(22) Semantic Inheritance Principle

$$hd\text{-}cx: \left[\begin{array}{ll} \text{MOTHER} & \left[\text{SEM}\ \left[\begin{array}{ll} \text{MODE} & \boxed{1} \\ \text{INDEX} & \boxed{2} \end{array}\right]\right] \\ \text{HD-DTR} & \left[\text{SEM}\ \left[\begin{array}{ll} \text{MODE} & \boxed{1} \\ \text{INDEX} & \boxed{2} \end{array}\right]\right] \end{array}\right]$$

(23) Valence Principle

$$hd\text{-}cx: \left[\begin{array}{ll} \text{MOTHER} & [\text{SYN}\ \ [\text{VAL}\ \ /\ \boxed{1}]] \\ \text{HD-DTR} & [\text{SYN}\ \ [\text{VAL}\ \ /\ \boxed{1}]] \end{array}\right]$$

(24) GAP Principle

$$hd\text{-}cx: \left[\begin{array}{ll} \text{MOTHER} & [\text{SYN}\ [\text{GAP}\ \ (\ \boxed{\text{A1}} \oplus ... \oplus \boxed{\text{An}}\ )\ \ominus\ \boxed{\text{A0}}\ ]\ ] \\ \text{HD-DTR} & [\text{SYN}\ [\text{STOP-GAP}\ \ \boxed{\text{A0}}\ ]] \\ \text{DTRS} & \langle\ [\text{SYN}\ [\text{GAP}\ \boxed{\text{A1}}\ ]]\ ,\ ...\ ,\ [\text{SYN}\ [\text{GAP}\ \boxed{\text{An}}\ ]]\ \rangle \end{array}\right]$$

[6]Another hallmark of construction grammar is its ability to accommodate the fact that the constructions themselves may contribute to the semantics of the phrases they license. This can be done by declaring another feature of constructions – CONSTRUCTIONAL-SEMANTICS (CX-SEM) – whose value is (possibly empty) list of predications. Such an analysis entails modifying the Semantic Compositionality Principle along the following lines:

Semantic Compositionality Principle (alternative version)

$$cx: \left[\begin{array}{ll} \text{MOTHER} & [\text{SEM}\ [\text{RESTR}\ \boxed{\text{A0}} \oplus \boxed{\text{A1}} \oplus ... \oplus \boxed{\text{An}}\ ]] \\ \text{DTRS} & \langle\ [\text{SEM}\ [\text{RESTR}\ \boxed{\text{A1}}]]\ ,\ ...\ ,\ [\text{SEM}\ [\text{RESTR}\ \boxed{\text{An}}]]\ \rangle \\ \text{CX-SEM} & \boxed{\text{A0}} \end{array}\right]$$

Nothing of analytic substance has changed here, but note that the Valence Principle is simplified in this formulation.

We are now ready to examine the maximal (that is, most specific) construction types. What we will see is that the constraints they must obey now follow from the interaction between the constraints on these types and those just presented. Moreover, this is just the same kind of constraint interaction that we have seen in previous chapters.

Consider first the head-complement construction, instances of which are subject to the following constraint:[7]

(25)

$$
\textit{hd-comp-cx}: \left[\begin{array}{ll}
\text{MOTHER} & [\text{SYN}\ [\text{VAL}\ [\text{COMPS}\ \langle\ \rangle\ ]\ ]] \\
\text{HD-DTR} & \boxed{0}\left[\begin{array}{l} \textit{word} \\ \text{SYN}\ [\text{VAL}\ [\text{COMPS}\ \boxed{\text{A}}\ ]] \end{array}\right] \\
\text{DTRS} & \langle\ \boxed{0}\ \rangle \oplus \boxed{\text{A}}\textit{nelist}
\end{array}\right]
$$

(25) identifies the HD-DTR with the first element of the DTRS list. This interacts with the Principle of Order in (19) to get head-initial order in head-complement constructs. (25) states explicitly that the mother's COMPS value differs from the head daughter's, hence overriding the Valence Principle (in part). All other constraints on superordinate types are compatible with (25). Hence these constraints interact to derive the following properties of head-complement constructions:

(26) (13) & (19) & (20)–(25):

$$
\left[\begin{array}{ll}
\text{MOTHER} & \left[\begin{array}{ll}
\text{PHON} & \boxed{\text{A1}} \oplus ... \oplus \boxed{\text{An}} \\
\text{SYN} & \left[\begin{array}{ll}
\text{HEAD} & \boxed{1} \\
\text{VAL} & \left[\begin{array}{ll}
\text{COMPS} & \langle\ \rangle \\
\text{SPR} & \boxed{\text{D}} \\
\text{MOD} & \boxed{\text{E}}
\end{array}\right]
\end{array}\right] \\
\text{SEM} & \left[\begin{array}{ll}
\text{MODE} & \boxed{2} \\
\text{INDEX} & \boxed{3} \\
\text{RESTR} & \boxed{\text{C1}} \oplus ... \oplus \boxed{\text{Cn}}
\end{array}\right]
\end{array}\right] \\
\text{HD-DTR}\ \boxed{4} & \left[\begin{array}{ll}
\textit{word} & \\
\text{SYN} & \left[\begin{array}{ll}
\text{HEAD} & \boxed{1} \\
\text{VAL} & \left[\begin{array}{ll}
\text{COMPS} & \langle\ \boxed{5}\ , ...,\ \boxed{\text{m}}\ \rangle \\
\text{SPR} & \boxed{\text{D}} \\
\text{MOD} & \boxed{\text{E}}
\end{array}\right]
\end{array}\right] \\
\text{SEM} & \left[\begin{array}{ll}
\text{MODE} & \boxed{2} \\
\text{INDEX} & \boxed{3}
\end{array}\right]
\end{array}\right] \\
\text{DTRS} & \left\langle \boxed{4}\left[\begin{array}{ll}\text{PHON} & \boxed{\text{A1}} \\ \text{RESTR} & \boxed{\text{C1}}\end{array}\right], \boxed{5}\left[\begin{array}{ll}\text{PHON} & \boxed{\text{A2}} \\ \text{RESTR} & \boxed{\text{C2}}\end{array}\right], ..., \boxed{\text{m}}\left[\begin{array}{ll}\text{PHON} & \boxed{\text{An}} \\ \text{RESTR} & \boxed{\text{Cn}}\end{array}\right] \right\rangle
\end{array}\right]
$$

[7] Note that *nelist* denotes a nonempty list.

These are the properties our grammar claims are true of all instantiations of the head-complement construction in English.[8] The head-complement constructs are those feature structures of type *phrase* that serve as the MOTHER value of some feature structure that instantiates (26). Two such feature structures are illustrated in (27) and (28):[9]

(27)
$$\left[\begin{array}{ll} \multicolumn{2}{l}{\textit{hd-comp-cx}} \\ \text{MOTHER} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{phrase}} \\ \text{PHON} & \langle \text{ talked , to , Kim } \rangle \\ \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \textit{verb} \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \boxed{\text{A}}\langle \text{ NP } \rangle \\ \text{COMPS} & \langle\ \rangle \end{array}\right] \end{array}\right] \\ \text{SEM} & [\ ...\ ] \end{array}\right] \\ \text{HD-DTR} & \boxed{0} \\ \text{DTRS} & \left\langle \boxed{0}\left[\begin{array}{ll} \multicolumn{2}{l}{\textit{word}} \\ \text{PHON} & \langle \text{ talked } \rangle \\ \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \textit{verb} \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \boxed{\text{A}} \\ \text{COMPS} & \langle \boxed{1} \rangle \end{array}\right] \end{array}\right] \\ \text{SEM} & [\ ...\ ] \end{array}\right], \boxed{1}\left[\begin{array}{ll} \multicolumn{2}{l}{\textit{phrase}} \\ \text{PHON} & \langle \text{ to , Kim } \rangle \\ \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \textit{prep} \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{array}\right] \end{array}\right] \\ \text{SEM} & [\ ...\ ] \end{array}\right] \right\rangle \end{array}\right]$$

(28)
$$\left[\begin{array}{ll} \multicolumn{2}{l}{\textit{hd-comp-cx}} \\ \text{MOTHER} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{phrase}} \\ \text{PHON} & \langle \text{ in , Seattle } \rangle \\ \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \textit{prep} \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \boxed{\text{A}} \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \boxed{\text{B}} \end{array}\right] \end{array}\right] \\ \text{SEM} & [\ ...\ ] \end{array}\right] \\ \text{HD-DTR} & \boxed{0} \\ \text{DTRS} & \left\langle \boxed{0}\left[\begin{array}{ll} \multicolumn{2}{l}{\textit{word}} \\ \text{PHON} & \langle \text{ in } \rangle \\ \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \textit{prep} \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \boxed{\text{A}} \\ \text{COMPS} & \langle \boxed{1} \rangle \\ \text{MOD} & \boxed{\text{B}} \end{array}\right] \end{array}\right] \\ \text{SEM} & [\ ...\ ] \end{array}\right], \boxed{1}\left[\begin{array}{ll} \multicolumn{2}{l}{\textit{word}} \\ \text{PHON} & \langle \text{ Seattle } \rangle \\ \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \textit{noun} \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{array}\right] \end{array}\right] \\ \text{SEM} & [\ ...\ ] \end{array}\right] \right\rangle \end{array}\right]$$

[8] Note that there must be at least one nonhead daughter in (26) in order to satisfy the *nelist* constraint on the type *hd-comp-cx*.

[9] The examples in (27) and (28) are, of course, abbreviated in that they do not show all of the values that the fully resolved feature structures are specified for.

The analysis of the other headed constructions is similar. Not many constraints are particular to maximal construction types, as sketched in (29)–(31):

(29)

$$\textit{hd-spr-cx}: \begin{bmatrix} \text{MOTHER} & \begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{SPR} & \langle\ \rangle\end{bmatrix}\end{bmatrix} \\ \text{HD-DTR} & \boxed{0}\begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{STOP-GAP} & \langle\ \rangle\end{bmatrix}\end{bmatrix} \\ \text{DTRS} & \langle \boxed{1}, \boxed{0} \rangle \end{bmatrix}$$

(30)

$$\textit{hd-mod-cx}: \begin{bmatrix} \text{HD-DTR} & \boxed{1}\begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{VAL} & \begin{bmatrix}\text{COMPS} \ \langle\ \rangle\end{bmatrix} \\ \text{STOP-GAP} & \langle\ \rangle\end{bmatrix}\end{bmatrix} \\ \text{DTRS} & \left\langle \boxed{1}, \begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{VAL} & \begin{bmatrix}\text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle \boxed{1} \rangle\end{bmatrix}\end{bmatrix}\end{bmatrix} \right\rangle \end{bmatrix}$$

(31)

$$\textit{hd-fill-cx}: \begin{bmatrix} \text{HD-DTR} & \boxed{0}\begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\textit{verb} & \\ \text{FORM} & \text{fin}\end{bmatrix} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix} \\ \text{GAP} & \langle \boxed{1} \rangle \\ \text{STOP-GAP} & \langle \boxed{1} \rangle\end{bmatrix}\end{bmatrix} \\ \text{DTRS} & \langle \boxed{1}[\text{GAP} \ \langle\ \rangle] , \boxed{0} \rangle \end{bmatrix}$$

(32) illustrates an instantiation of the head-filler construction (once again, we have only shown some of the feature specifications):

(32)
$$\begin{bmatrix} \textit{hd-fill-cx} \\ \text{MOTHER} & \begin{bmatrix} \text{PHON} & \langle \text{ Bagels , I , think , she , likes } \rangle \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \boxed{2}\begin{bmatrix} \textit{verb} \\ \text{FORM} & \text{fin} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \\ \text{GAP} & \langle\ \rangle \end{bmatrix} \\ \text{SEM} & [\ \ldots\ ] \end{bmatrix} \\ \text{HD-DTR} & \boxed{0} \\ \text{DTRS} & \left\langle \boxed{1}\begin{bmatrix} \text{PHON} \langle \text{ Bagels } \rangle \\ \text{SYN} \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} [\ \ldots\ ] \end{bmatrix} , \boxed{0}\begin{bmatrix} \text{PHON} \langle \text{ I , think , she , likes } \rangle \\ \text{SYN} \begin{bmatrix} \text{HEAD} & \boxed{2} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \\ \text{GAP} & \langle \boxed{1} \rangle \\ \text{STOP-GAP} & \langle \boxed{1} \rangle \end{bmatrix} \\ \text{SEM} [\ \ldots\ ] \end{bmatrix} \right\rangle \end{bmatrix}$$

Our nonheaded rules lend themselves to a similar reanalysis in terms of nonheaded constructions and type constraints. The Imperative Rule, for example, corresponds to the type constraint in (33):

(33)
$$\textit{imp-cx} : \begin{bmatrix} \text{MOTHER} & \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & [\text{SPR} \ \ \langle\ \rangle] \\ \text{GAP} & \boxed{A} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{dir} \\ \text{INDEX} & s \end{bmatrix} \end{bmatrix} \\ \text{DTRS} & \left\langle \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} \\ \text{INF} & - \\ \text{FORM} & \text{base} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \text{ NP[PER 2nd] } \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \\ \text{GAP} & \boxed{A} \end{bmatrix} \\ \text{SEM} & [\text{INDEX} \ \ s\ ] \end{bmatrix} \right\rangle \end{bmatrix}$$

This constraint correctly requires an imperative phrase to contain a nonhead daughter that is a base-form VP whose unexpressed subject is second person.

Finally, let us reconsider the Coordination Rule, which can also be recast in terms of a maximal construction type and an associated type constraint:

(34) *coord-cx* :

$$
\left[\begin{array}{ll}
\text{MOTHER} & \left[\begin{array}{ll}
\text{SYN} & \left[\begin{array}{ll}
\text{HEAD} & [\text{FORM}\ \boxed{1}] \\
\text{VAL} & \boxed{2} \\
\text{GAP} & \boxed{A}
\end{array}\right] \\
\text{SEM} & [\text{IND}\ \ s_0]
\end{array}\right] \\
\text{DTRS} & \left\langle \left[\begin{array}{ll}
\text{SYN} & \left[\begin{array}{ll}
\text{HEAD} & [\text{FORM}\ \boxed{1}] \\
\text{VAL} & \boxed{2} \\
\text{GAP} & \boxed{A}
\end{array}\right] \\
\text{SEM} & [\text{IND}\ \ s_1]
\end{array}\right], \ldots, \left[\begin{array}{ll}
\text{SYN} & \left[\begin{array}{ll}
\text{HEAD} & [\text{FORM}\ \boxed{1}] \\
\text{VAL} & \boxed{2} \\
\text{GAP} & \boxed{A}
\end{array}\right] \\
\text{SEM} & [\text{IND}\ \ s_{n-1}]
\end{array}\right], \right. \\
& \left. \left[\begin{array}{ll}
\text{HEAD} & \textit{conj} \\
\text{IND} & s_0 \\
\text{RESTR} & \left\langle [\text{ARGS}\ \langle s_1 \ldots s_n \rangle\ ] \right\rangle
\end{array}\right], \left[\begin{array}{ll}
\text{SYN} & \left[\begin{array}{ll}
\text{HEAD} & [\text{FORM}\ \boxed{1}] \\
\text{VAL} & \boxed{2} \\
\text{GAP} & \boxed{A}
\end{array}\right] \\
\text{SEM} & [\text{IND}\ \ s_n]
\end{array}\right] \right\rangle
\end{array}\right]
$$

An instantiation of this construction is sketched in (35):

(35) *coord-cx* :

$$
\left[\begin{array}{ll}
\text{MOTHER} & \left[\begin{array}{ll}
\text{PHON} & \langle\ \text{Kim , sleeps , and , Pat , works}\ \rangle \\
\text{SYN} & \left[\begin{array}{ll}
\text{HEAD} & \textit{verb} \\
\text{VAL} & \left[\begin{array}{ll}
\text{SPR} & \langle\ \rangle \\
\text{COMPS} & \langle\ \rangle
\end{array}\right]
\end{array}\right] \\
\text{SEM} & [\ \ldots\ ]
\end{array}\right] \\
\text{DTRS} & \left\langle \left[\begin{array}{ll}
\text{PHON} & \langle\ \text{Kim , sleeps}\ \rangle \\
\text{SYN} & \left[\begin{array}{ll}
\text{HEAD} & \textit{verb} \\
\text{VAL} & \left[\begin{array}{ll}
\text{SPR} & \langle\ \rangle \\
\text{COMPS} & \langle\ \rangle
\end{array}\right]
\end{array}\right] \\
\text{SEM} & [\ \ldots\ ]
\end{array}\right], \left[\begin{array}{ll}
\text{PHON} & \langle\ \text{and}\ \rangle \\
\text{SYN} & \left[\text{HEAD}\ \ \textit{conj}\right] \\
\text{SEM} & [\ \ldots\ ]
\end{array}\right], \right. \\
& \left. \left[\begin{array}{ll}
\text{PHON} & \langle\ \text{Pat , works}\ \rangle \\
\text{SYN} & \left[\begin{array}{ll}
\text{HEAD} & \textit{verb} \\
\text{VAL} & \left[\begin{array}{ll}
\text{SPR} & \langle\ \rangle \\
\text{COMPS} & \langle\ \rangle
\end{array}\right]
\end{array}\right] \\
\text{SEM} & [\ \ldots\ ]
\end{array}\right] \right\rangle
\end{array}\right]
$$

So we see that both headed and nonheaded phrasal constructions can be treated in terms of types and type constraints. The distinction between the two types of construction is drawn precisely to restrict the domain of application of certain constraints, for example, the Head Feature Principle and the Valence Principle, so that they apply only to headed constructions. Other principles, such as the Semantic Compositionality Principle are stated as constraints on the type *construction* (*cx*) and hence apply to headed and nonheaded constructions alike.

Once this system of construction types is in place, there are various ways it can be simplified using methods that are by now familiar. For example, it is possible to use multiple inheritance to express generalizations that would have been difficult to express in our previous grammar. Recall that in Chapter 14, we noted that topicalization was not the only example of a long-distance, filler-gap dependency. Relative clauses (e.g. (*the person*) *who Sandy thought that Pat visited*) and *wh*-questions (e.g. *Who did Sandy think that Pat visited?*) are two other kinds of clause that involve the same sort of long-distance dependency. To put it differently, these are two subvarieties of head-filler construction. As further investigation of clauses and other kinds of constructions reveals, this same pattern of cross-cutting generalizations holds true throughout the grammar.[10] There are many kinds of constructions in a language, yet they fall into groups that share many common properties.

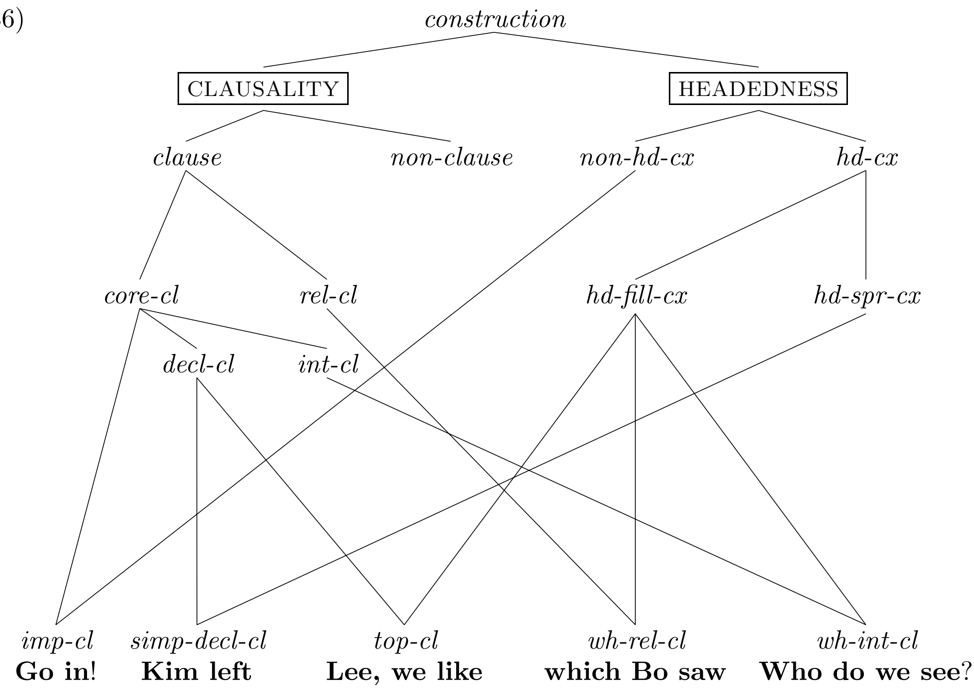
This fundamental fact about language – that constructions cluster into 'natural classes' with shared common properties – can be directly expressed by a multiple inheritance hierarchy. To do this, we partition the set of constructions into two dimensions: CLAUSALITY and HEADEDNESS. This leads to the following (incomplete) picture of construction types:[11]

[10]See Fillmore et al. forthcoming, Sag 1997, Ginzburg and Sag 2000.

[11]Here we use a number of new abbreviations:

imp-cl	*imperative-clause*	*decl-cl*	*declarative-clause*
simp-decl-cl	*simple-declarative-clause*	*top-cl*	*topicalized-clause*
wh-rel-cl	*wh-relative-clause*	*wh-int-cl*	*wh-interrogative-clause*
core-cl	*core-clause*		

(36)



Once more, we appeal to the superordinate types to serve as the locus of constraints that cut across the various subtypes. The constraints on the type *hd-fill-cx*, for example, will cut across topicalization and the various *wh*-constructions, though each kind of head-filler construction will have properties of its own (e.g. the inversion that is characteristic of the main clause *wh*-questions). Working out the details of this factorization of grammatical constraints in terms of superordinate types is the object of ongoing inquiry in a variety of constraint-based grammatical frameworks, including HPSG, Construction Grammar, and Word (Dependency) Grammar (see Appendix B).

16.6 Locality

This completes our sketch of how we can modify our framework to embrace Saussure's conception of language as a system of signs (one augmented to encompass recursivity, to be sure). With the revisions to the theoretical architecture that we have outlined – modeling words and phrases in terms of signs and eliminating grammatical rules in favor of constructions modeled as feature structures – our grammar attains a remarkable uniformity. All its devices have become constraints on feature structures. This includes lexical entries, syntactic and semantic principles, and now grammar rules.

This view of grammar comports well with the psycholinguistic results discussed in Chapter 9. We have retained the surface-oriented and strongly lexicalist character of our theory, which we argued for earlier. Formulating all the essential components of our theory as feature structure constraints renders the theory compatible with a wide range of processing strategies. Evidence like that cited in Chapter 9 suggests that distinct tasks or contexts might require the same linguistic knowledge to be employed diversely: in divergent processes and divergent orders. The psycholinguistic plausibility of the constraint-based lexicalist approach is preserved in the conceptually more elegant and uniform sign-based architecture developed in this chapter.

The notion of 'construction' has moved to center stage in our grammar, making the approach that we have arrived at a kind of 'Construction Grammar'. But notice that our system of constructions maintains the locality of our earlier system, i.e. the locality of CFGs. It makes no difference how a given phrase has been constructed. If it satisfies the appropriate conditions, then it can serve as a daughter in some other construction. There is no way for a word or construction to access the daughters of its daughters directly. CFGs allow selection between sisters; our CFG-like system has extended this notion of locality, allowing a restricted kind of 'aunt' selection, since the SPR value is passed up and discharged at the next higher level of structure. Consequently, under our current assumptions, we would be hard-pressed to write a grammar for a language where, for example, some verb selected for a complement that was an S whose VP head daughter contained an accusative NP. The prediction of locality in this sense is exactly right: there are no human languages where a verb selects for an S complement whose VP head daughter must contain an accusative NP. It is a striking fact about human languages that they don't exhibit dependencies of this kind, as far as we know.

Our CFG-like system can describe a nonlocal dependency only if some feature is introduced to carry the information about the accusative NP up to the VP and to the S it heads. Once the relevant information is encoded on the S node, it would become locally selectable. Our assumptions about what features are part of signs and how they are inherited thus constitute an hypothesis about what information is locally available for selection or locally accessible to a constructional constraint. For example, the GAP feature, taken together with our GAP Principle, makes information about missing elements locally accessible. But when an adjective like *hard* selects for a complement specified as [GAP ⟨NP ⟩], the gap may be one, two, or twenty-two levels 'deep' within the complement (not nonlocal to a fixed depth). That is the prediction made by our GAP analysis, a correct one in light of examples like the following:

(37) a. Sandy is hard to please __ .
b. Sandy is hard for us to continue to please __ .
c. Getting there early is hard for us to imagine them considering __ .

Moreover, the 'feature passing' method, which makes the GAP information accessible at every level between the filler and the gap position, is independently confirmed. As the Irish data presented in Problem 4 of Chapter 14 show, there are languages that have words that require a 'gappy' environment, which must be defined so as to include the entire domain between the filler and the gap. This is just as predicted by the feature-based theory of long-distance dependencies.

Stepping back from the analyses we have presented in this book, we can see that they embody the hypothesis that the 'nonlocal' syntactic information that is locally accessible to selectional or constructional constraints is of two kinds:

(38) a. the information (expressed in terms of HEAD and VAL specifications) that is passed up from a lexical head throughout the domain it projects (including subjects and modifiers) and

b. the unbounded information, e.g. GAP specifications inherited in virtue of the GAP Principle.

Though there is room for debate about what information should be encoded in a headed domain[12] or what kind of unbounded features should be assumed for a given language, the basic picture sketched in (38) is one that reflects a consensus in the field of grammatical theory. The framework we have presented here for expressing this consensus, however, is considerably more rigorous and perspicuous than most, and psycholinguistically more plausible. This has the value of allowing grammatical descriptions to 'scale up' at the same time that the consequences of alternate analytic hypotheses can be precisely assessed, both in the context of new data sets and with respect to new findings about human language processing and language use.

16.7 Summary

This final chapter has suggested a major reconceptualization of the grammatical theory developed in the preceding fifteen chapters. We have not worked out all of the details here, but we have presented enough to indicate that recasting various theoretical constructs as signs would have a number of advantages, including the following:

- *Parsimony* Unifying our formal account of lexical entries, lexical rules, grammar rules, and general principles of well-formedness in terms of constraints on feature structures makes the theory simpler and more elegant.
- *Constructions* By replacing phrase structure rules with an inheritance hierarchy of construction types, we can capture generalizations across different kinds of constructions that we had to treat as accidental in our earlier theory.
- *Interfaces* By including phonological, syntactic, and semantic information within the same data structures (feature structures), we clear the way for stating constraints that express the relationships among these. We have explored some of the syntax-semantics interface constraints in earlier chapters; the new architecture allows similar interactions between the phonology and the syntax and semantics, as is required in order to account for the interpretational effect of pitch accent, for example the distinct semantic/pragmatic effects of *KIM loves Sandy*, *Kim loves SANDY*, and *Kim LOVES Sandy*. A natural extension would be to include in our feature structures some information about the pragmatics – that is, about appropriate contexts of use; and the sign-based architecture would allow us to state constraints on the interactions of pragmatics with semantics, syntax, and phonology.
- *Process Independence* By providing purely declarative descriptions of the relationships among different pieces of information, the sign-based grammar, like the one summarized in Appendix A, can be used in comprehension models, production models, or many kinds of computational applications.

[12]For instance there is ongoing debate about whether information about a verb's subject should be available at a level higher than where the subject actually appears.

Investigation of sign-based grammars is a robust area of current research. Our remarks in this chapter have only scratched the surface. Indeed, the same could be said on a more general level: syntactic theory is a multifaceted field of study that is rapidly changing; our introduction to it has been necessarily very partial and somewhat simplified.

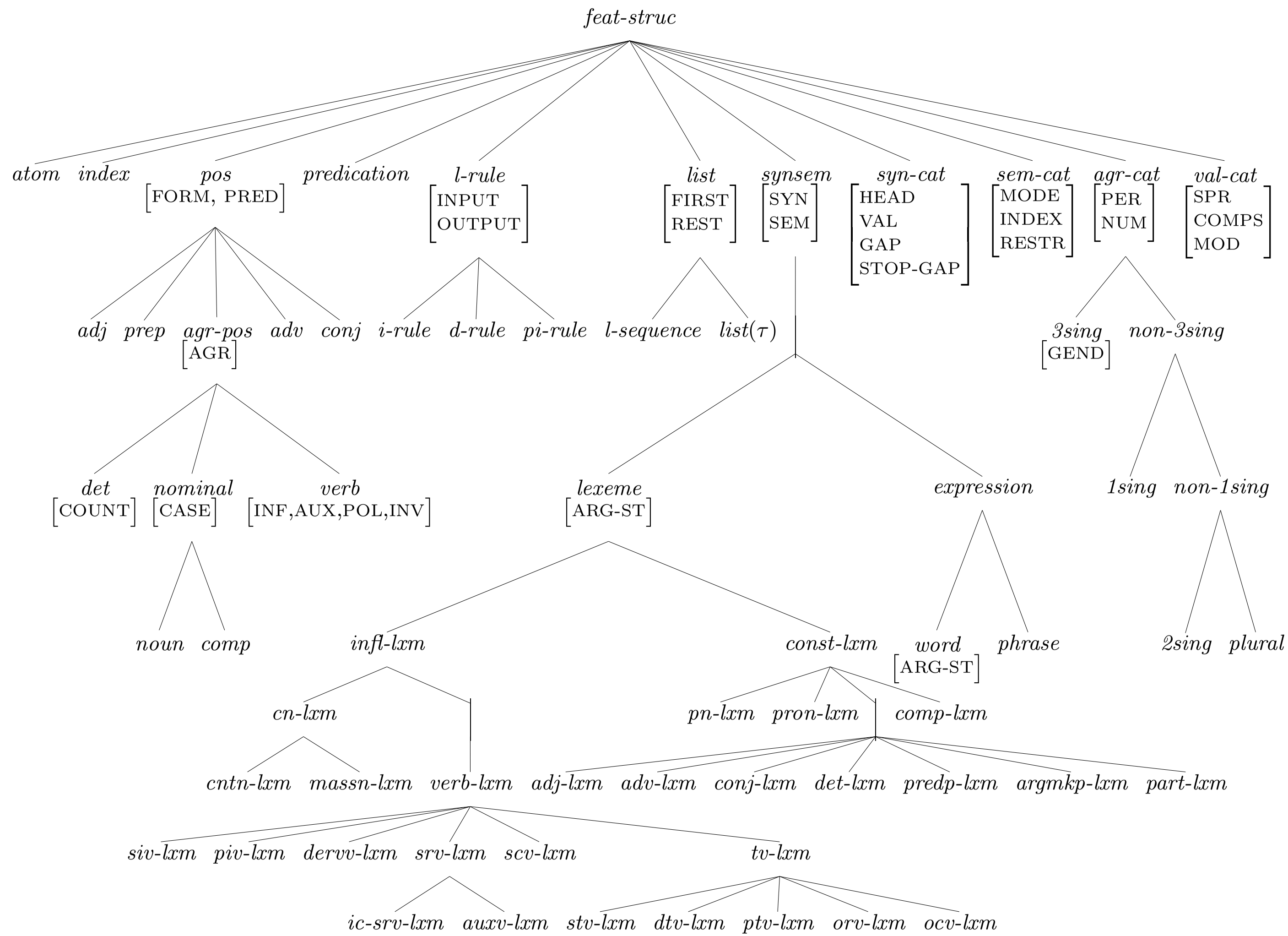
It is impossible to predict with any confidence how the field of syntax will develop in the future. No doubt many of the specific analyses put forward in this text will eventually need to be revised – as has been the case with most specific analyses in the generative tradition over the past several decades. But each reanalysis deepens our understanding of the phenomena under investigation and lays the groundwork for more refined accounts. The grammar developed here builds on the insights of three generations of generative grammarians, as well as much earlier work, and we hope linguists of the future will find our proposals equally useful.

As a textbook in a rapidly changing field, the present work has been as concerned with process as with product. Working through and comparing detailed analyses of specific phenomena is the best way to illustrate how syntactic theorizing is done, and hence how progress in the field comes about. Our primary goals in this text have been to introduce the reader to modes of argumentation and to teach the value of explicit formalization of hypotheses, the need to take semantics seriously in syntactic analysis, and the value of paying attention to processing considerations. If we have achieved these goals, then we have accomplished what we set out to do.

Appendix A
Summary of the Grammar

A.1 The Type Hierarchy

The types that are used in the grammar we have developed in Chapters 3–15 are presented here in two distinct formats (as in the interim summary in Chapter 9). The first is a tree diagram indicating the hierarchical organization of all relevant types. The second is a list of particular types, with an indication of which features are appropriate for each, what type of value is appropriate for each such feature, and each type's immediate supertype (IST).


feat-struc
atom
index
pos [FORM, PRED]
predication
l-rule [INPUT OUTPUT]
list [FIRST REST]
synsem [SYN SEM]
syn-cat [HEAD VAL GAP STOP-GAP]
sem-cat [MODE INDEX RESTR]
agr-cat [PER NUM]
val-cat [SPR COMPS MOD]
adj
prep
agr-pos [AGR]
adv
conj
i-rule
d-rule
pi-rule
l-sequence
list(τ)
3sing [GEND]
non-3sing
det [COUNT]
nominal [CASE]
verb [INF,AUX,POL,INV]
lexeme [ARG-ST]
expression
1sing
non-1sing
noun
comp
infl-lxm
const-lxm
word [ARG-ST]
phrase
2sing
plural
cn-lxm
pn-lxm
pron-lxm
comp-lxm
cntn-lxm
massn-lxm
verb-lxm
adj-lxm
adv-lxm
conj-lxm
det-lxm
predp-lxm
argmkp-lxm
part-lxm
siv-lxm
piv-lxm
dervv-lxm
srv-lxm
scv-lxm
tv-lxm
ic-srv-lxm
auxv-lxm
stv-lxm
dtv-lxm
ptv-lxm
orv-lxm
ocv-lxm

A.2 Feature Declarations and Type Constraints

GENERAL TYPES		
TYPE	FEATURES/CONSTRAINTS	IST
feat-struc		
atom, index		*feat-struc*
l-rule	$\left[\begin{array}{ll} \text{INPUT} & \textit{l-sequence}\left\langle \text{X} , \left[\begin{array}{ll} \text{SEM} & / \boxed{2} \\ \text{SYN} & \left[\text{STOP-GAP} \quad \boxed{\text{A}}\right] \end{array}\right] \right\rangle \\ \text{OUTPUT} & \textit{l-sequence}\left\langle \text{Y} , \left[\begin{array}{ll} \text{SEM} & / \boxed{2} \\ \text{SYN} & \left[\text{STOP-GAP} \quad \boxed{\text{A}}\right] \end{array}\right] \right\rangle \end{array}\right]$	*feat-struc*
i-rule	$\left[\begin{array}{ll} \text{INPUT} & \left\langle \text{X} , \left[\begin{array}{ll} \textit{lexeme} & \\ \text{SYN} & \boxed{3} \\ \text{ARG-ST} & \boxed{\text{A}} \end{array}\right] \right\rangle \\ \text{OUTPUT} & \left\langle \text{Y} , \left[\begin{array}{ll} \textit{word} & \\ \text{SYN} & \boxed{3} \\ \text{ARG-ST} & \boxed{\text{A}} \end{array}\right] \right\rangle \end{array}\right]$	*l-rule*
d-rule	$\left[\begin{array}{ll} \text{INPUT} & \left\langle \text{X} , \left[\begin{array}{ll} \textit{lexeme} & \\ \text{SYN} & / \boxed{3} \end{array}\right] \right\rangle \\ \text{OUTPUT} & \left\langle \text{Y} , \left[\begin{array}{ll} \textit{lexeme} & \\ \text{SYN} & / \boxed{3} \end{array}\right] \right\rangle \end{array}\right]$	*l-rule*
pi-rule	$\left[\begin{array}{ll} \text{INPUT} & \left\langle / \boxed{0} , \left[\begin{array}{ll} \textit{word} & \\ \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & / \boxed{1} \\ \text{VAL} & \left[\text{MOD} \ \boxed{\text{A}}\right] \end{array}\right] \end{array}\right] \right\rangle \\ \text{OUTPUT} & \left\langle / \boxed{0} , \left[\begin{array}{ll} \textit{word} & \\ \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & / \boxed{1} \\ \text{VAL} & \left[\text{MOD} \ \boxed{\text{A}}\right] \end{array}\right] \end{array}\right] \right\rangle \end{array}\right]$	*l-rule*

GENERAL TYPES (CONTINUED)		
TYPE	FEATURES/CONSTRAINTS	IST
list		*feat-struc*
list(τ)	$\begin{bmatrix} \text{FIRST} & \tau \\ \text{REST} & \textit{list}(\tau) \end{bmatrix}$	*list*
l-sequence	$\begin{bmatrix} \text{FIRST} & \textit{atom} \\ \text{REST} & \langle \textit{word} \rangle \mid \langle \textit{lexeme} \rangle \end{bmatrix}$	*list*
synsem	$\begin{bmatrix} \text{SYN} & \textit{syn-cat} \\ \text{SEM} & \textit{sem-cat} \end{bmatrix}$	*feat-struc*
syn-cat	$\begin{bmatrix} \text{HEAD} & \textit{pos} \\ \text{VAL} & \textit{val-cat} \\ \text{GAP} & \textit{list}(\textit{expression}) \\ \text{STOP-GAP} & \textit{list}(\textit{expression}) \end{bmatrix}$	*feat-struc*
sem-cat	$\begin{bmatrix} \text{MODE} & \left\{\text{prop, ques, dir, ref, ana, none}\right\} \\ \text{INDEX} & \left\{\textit{index}\text{, none}\right\} \\ \text{RESTR} & \textit{list}(\textit{predication}) \end{bmatrix}$	*feat-struc*
val-cat	$\begin{bmatrix} \text{SPR} & \textit{list}(\textit{expression}) \\ \text{COMPS} & \textit{list}(\textit{expression}) \\ \text{MOD} & \textit{list}(\textit{expression}) \end{bmatrix}$	*feat-struc*
expression		*synsem*
phrase		*expression*
word	$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \boxed{\text{A}} \\ \text{COMPS} & \boxed{\text{B}} \ominus \boxed{\text{C}} \end{bmatrix} \\ \text{GAP} & \boxed{\text{C}} \end{bmatrix} \\ \text{ARG-ST} & \boxed{\text{A}} \oplus \boxed{\text{B}} \end{bmatrix}$	*expression*

LEXEME TYPES		
TYPE	FEATURES/CONSTRAINTS	IST
lexeme	$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix}\text{MOD} & / \langle\ \rangle\end{bmatrix} \\ \text{STOP-GAP} & / \langle\ \rangle \end{bmatrix} \\ \text{ARG-ST} & \textit{list}(\textit{expression}) \end{bmatrix}$	*synsem*
infl-lxm	$\begin{bmatrix} \text{HEAD} & \begin{bmatrix}\text{AGR} & \boxed{1}\end{bmatrix} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle \begin{bmatrix}\text{AGR} & \boxed{1}\end{bmatrix} \rangle\end{bmatrix} \end{bmatrix}$	*lexeme*
const-lxm		*lexeme*
cn-lxm	$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \begin{bmatrix}\text{PER 3rd}\end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & / \text{ ref} \\ \text{INDEX} & i \end{bmatrix} \\ \text{ARG-ST} & \begin{bmatrix} \text{FIRST} & \text{DP}_i \\ \text{REST} & / \langle\ \rangle \end{bmatrix} \end{bmatrix}$	*infl-lxm*
verb-lxm	$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{PRED} & - \\ \text{INF} & / - \\ \text{AUX} & / - \\ \text{POL} & - \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{MODE} & \text{prop}\end{bmatrix} \\ \text{ARG-ST} & \left\langle \begin{bmatrix} \text{HEAD} & \textit{nominal} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix}, \ldots \right\rangle \end{bmatrix}$	*infl-lxm*
cntn-lxm	$\begin{bmatrix}\text{ARG-ST} & \langle\ [\text{COUNT } +]\ , \ldots \rangle\end{bmatrix}$	*cn-lxm*
massn-lxm	$\begin{bmatrix}\text{ARG-ST} & \langle\ [\text{COUNT } -]\ , \ldots \rangle\end{bmatrix}$	*cn-lxm*

LEXEME TYPES (CONTINUED)		
TYPE	FEATURES/CONSTRAINTS	IST
siv-lxm	$\begin{bmatrix}\text{ARG-ST} & \langle\ \text{X}\ \rangle\end{bmatrix}$	*verb-lxm*
piv-lxm	$\begin{bmatrix}\text{ARG-ST} & \langle\ \text{X}\ ,\ \text{PP}\ \rangle\end{bmatrix}$	*verb-lxm*
srv-lxm	$\begin{bmatrix}\text{ARG-ST} & \left\langle \boxed{1}\ ,\ \begin{bmatrix}\text{SPR} & \langle\ \boxed{1}\ \rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix}\right\rangle\end{bmatrix}$	*verb-lxm*
ic-srv-lxm	$\begin{bmatrix}\text{SEM} & \begin{bmatrix}\text{RESTR} & \left\langle\begin{bmatrix}\text{ARG} & s\end{bmatrix}\right\rangle\end{bmatrix} \\ \text{ARG-ST} & \left\langle \text{X}\ ,\ \overset{\text{VP}}{\begin{bmatrix}\text{INF} & + \\ \text{INDEX} & s\end{bmatrix}}\right\rangle\end{bmatrix}$	*srv-lxm*
auxv-lxm	$\begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{AUX} & +\end{bmatrix}\end{bmatrix}\end{bmatrix}$	*srv-lxm*
scv-lxm	$\begin{bmatrix}\text{SEM} & \begin{bmatrix}\text{RESTR} & \left\langle\begin{bmatrix}\text{ARG} & s\end{bmatrix}\right\rangle\end{bmatrix} \\ \text{ARG-ST} & \left\langle \text{NP}_i\ ,\ \begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \text{NP}_i\ \rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s\end{bmatrix}\end{bmatrix}\right\rangle\end{bmatrix}$	*verb-lxm*

LEXEME TYPES (CONTINUED)		
TYPE	FEATURES/CONSTRAINTS	IST
tv-lxm	$\begin{bmatrix}\text{ARG-ST} & \left\langle \text{X}\ , \begin{bmatrix}\text{HEAD} & \textit{nominal} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix}\end{bmatrix}, \ldots \right\rangle\end{bmatrix}$	*verb-lxm*
stv-lxm	$\begin{bmatrix}\text{ARG-ST} & \langle\ \text{X}\ , \text{Y}\ \rangle\end{bmatrix}$	*tv-lxm*
dtv-lxm	$\begin{bmatrix}\text{ARG-ST} & \langle\ \text{X}\ , \text{Y}\ , \text{NP}\ \rangle\end{bmatrix}$	*tv-lxm*
ptv-lxm	$\begin{bmatrix}\text{ARG-ST} & \langle\ \text{X}\ , \text{Y}\ , \text{PP}\ \rangle\end{bmatrix}$	*tv-lxm*
orv-lxm	$\begin{bmatrix}\text{SEM} & \begin{bmatrix}\text{RESTR} & /\left\langle\begin{bmatrix}\text{ARG} & s\end{bmatrix}\right\rangle\end{bmatrix} \\ \text{ARG-ST} & \left\langle \text{NP}\ , \boxed{1}\ , \begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \boxed{1}\ \rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s\end{bmatrix}\end{bmatrix}\right\rangle\end{bmatrix}$	*tv-lxm*
ocv-lxm	$\begin{bmatrix}\text{SEM} & \begin{bmatrix}\text{RESTR} & \left\langle\begin{bmatrix}\text{ARG} & s\end{bmatrix}\right\rangle\end{bmatrix} \\ \text{ARG-ST} & \left\langle \text{NP}\ , \text{NP}_i\ , \begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \text{NP}_i\ \rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s\end{bmatrix}\end{bmatrix}\right\rangle\end{bmatrix}$	*tv-lxm*
dervv-lxm		*verb-lxm*

LEXEME TYPES (CONTINUED)		
TYPE	FEATURES/CONSTRAINTS	IST
pn-lxm	$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \begin{bmatrix} \text{PER} & \text{3rd} \\ \text{NUM} & /\ \text{sg} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \end{bmatrix} \\ \text{ARG-ST} & /\ \langle\ \rangle \end{bmatrix}$	*const-lxm*
pron-lxm	$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{noun} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & /\ \text{ref} \end{bmatrix} \\ \text{ARG-ST} & \langle\ \rangle \end{bmatrix}$	*const-lxm*
conj-lxm	$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{conj} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{none} \end{bmatrix} \\ \text{ARG-ST} & \langle\ \rangle \end{bmatrix}$	*const-lxm*
adj-lxm	$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{adj} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \text{X}\ \rangle \\ \text{MOD} & \langle \begin{bmatrix} \text{HEAD}\ \textit{noun} \end{bmatrix} \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \begin{bmatrix} \text{HEAD} & \textit{nominal} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix}, \ldots \right\rangle \end{bmatrix}$	*const-lxm*
adv-lxm	$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{adv} \\ \text{VAL} & \begin{bmatrix} \text{MOD} & \langle \begin{bmatrix} \text{HEAD}\ \textit{verb} \end{bmatrix} \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{none} \end{bmatrix} \end{bmatrix}$	*const-lxm*
det-lxm	$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{det} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & /\ \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{none} \end{bmatrix} \end{bmatrix}$	*const-lxm*

LEXEME TYPES (CONTINUED)		
predp-lxm	$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{prep} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \text{X} \rangle \\ \text{MOD} & \langle \text{Y} \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \\ \text{RESTR} & \langle \text{Z} \rangle \end{bmatrix} \\ \text{ARG-ST} & \langle \text{NP} , \text{NP} \rangle \end{bmatrix}$	*const-lxm*
argmkp-lxm	$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{prep} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \boxed{1} \\ \text{INDEX} & \boxed{2} \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \\ \text{ARG-ST} & \left\langle \begin{array}{c} \text{NP} \\ \begin{bmatrix} \text{MODE} & \boxed{1} \\ \text{INDEX} & \boxed{2} \end{bmatrix} \end{array} \right\rangle \end{bmatrix}$	*const-lxm*
comp-lxm	$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{comp} & \\ \text{AGR} & \textit{3sing} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \begin{array}{c} \text{S} \\ \begin{bmatrix} \text{INDEX} & s \end{bmatrix} \end{array} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \end{bmatrix}$	*const-lxm*
part-lxm		*const-lxm*

OTHER GRAMMATICAL TYPES		
TYPE	FEATURES/CONSTRAINTS	IST
pos	$\begin{bmatrix} \text{FORM} & \left\{ \begin{array}{l} \text{fin, base, prp, psp, pass,} \\ \text{to, nform, aform, cform, ...} \end{array} \right\} \\ \text{PRED} & \{+, -\} \end{bmatrix}$	*feat-struc*
agr-pos	$[\text{AGR} \quad \textit{agr-cat}]$	*pos*
verb	$\begin{bmatrix} \text{INF} & \{+, -\} \\ \text{AUX} & \{+, -\} \\ \text{POL} & \{+, -\} \\ \text{INV} & \{+, -\} \end{bmatrix}$	*agr-pos*
nominal	$[\text{CASE} \quad \{\text{nom, acc}\}]$	*agr-pos*
noun	$[\text{FORM} \quad / \text{ nform}]$	*nominal*
comp	$[\text{FORM} \quad \text{cform}]$	*nominal*
det	$[\text{COUNT} \quad \{+, -\}]$	*agr-pos*
adj	$[\text{FORM} \quad \text{aform}]$	*pos*
prep, *adv*, *conj*		*pos*
agr-cat	$\begin{bmatrix} \text{PER} & \{\text{1st, 2nd, 3rd}\} \\ \text{NUM} & \{\text{sg, pl}\} \end{bmatrix}$	*feat-struc*
3sing	$\begin{bmatrix} \text{PER} & \text{3rd} \\ \text{NUM} & \text{sg} \\ \text{GEND} & \{\text{fem, masc, neut}\} \end{bmatrix}$	*agr-cat*
non-3sing		*agr-cat*
1sing	$\begin{bmatrix} \text{PER} & \text{1st} \\ \text{NUM} & \text{sg} \end{bmatrix}$	*non-3sing*
non-1sing		*non-3sing*
2sing	$\begin{bmatrix} \text{PER} & \text{2nd} \\ \text{NUM} & \text{sg} \end{bmatrix}$	*non-1sing*
plural	$[\text{NUM} \quad \text{pl}]$	*non-1sing*
predication	$\begin{bmatrix} \text{RELN} & \{\textbf{love}, \textbf{walk}, ...\} \\ \ldots & \end{bmatrix}$	*feat-struc*

A.3 Abbreviations

$$\mathrm{S} = \begin{bmatrix} \mathrm{SYN} & \begin{bmatrix} \mathrm{HEAD} & \textit{verb} \\ \mathrm{VAL} & \begin{bmatrix} \mathrm{COMPS} & \langle\ \rangle \\ \mathrm{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \qquad \mathrm{NP}_i = \begin{bmatrix} \mathrm{SYN} & \begin{bmatrix} \mathrm{HEAD} & \textit{noun} \\ \mathrm{VAL} & \begin{bmatrix} \mathrm{COMPS} & \langle\ \rangle \\ \mathrm{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \mathrm{SEM} & \begin{bmatrix} \mathrm{INDEX} & i \end{bmatrix} \end{bmatrix}$$

$$\mathrm{VP} = \begin{bmatrix} \mathrm{SYN} & \begin{bmatrix} \mathrm{HEAD} & \textit{verb} \\ \mathrm{VAL} & \begin{bmatrix} \mathrm{COMPS} & \langle\ \rangle \\ \mathrm{SPR} & \langle\ \mathrm{X}\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \qquad \mathrm{NOM} = \begin{bmatrix} \mathrm{SYN} & \begin{bmatrix} \mathrm{HEAD} & \textit{noun} \\ \mathrm{VAL} & \begin{bmatrix} \mathrm{COMPS} & \langle\ \rangle \\ \mathrm{SPR} & \langle\ \mathrm{X}\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\mathrm{V} = \begin{bmatrix} \textit{word} & \\ \mathrm{SYN} & \begin{bmatrix} \mathrm{HEAD} & \textit{verb} \end{bmatrix} \end{bmatrix} \qquad \mathrm{N} = \begin{bmatrix} \textit{word} & \\ \mathrm{SYN} & \begin{bmatrix} \mathrm{HEAD} & \textit{noun} \end{bmatrix} \end{bmatrix}$$

$$\mathrm{PP} = \begin{bmatrix} \mathrm{SYN} & \begin{bmatrix} \mathrm{HEAD} & \textit{prep} \\ \mathrm{VAL} & \begin{bmatrix} \mathrm{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \qquad \mathrm{CP} = \begin{bmatrix} \mathrm{SYN} & \begin{bmatrix} \mathrm{HEAD} & \textit{comp} \\ \mathrm{VAL} & \begin{bmatrix} \mathrm{COMPS} & \langle\ \rangle \\ \mathrm{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\mathrm{P} = \begin{bmatrix} \textit{word} & \\ \mathrm{SYN} & \begin{bmatrix} \mathrm{HEAD} & \textit{prep} \end{bmatrix} \end{bmatrix} \qquad \mathrm{C} = \begin{bmatrix} \textit{word} & \\ \mathrm{SYN} & \begin{bmatrix} \mathrm{HEAD} & \textit{comp} \end{bmatrix} \end{bmatrix}$$

$$\mathrm{AP} = \begin{bmatrix} \mathrm{SYN} & \begin{bmatrix} \mathrm{HEAD} & \textit{adj} \\ \mathrm{VAL} & \begin{bmatrix} \mathrm{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \qquad \mathrm{DP} = \begin{bmatrix} \mathrm{SYN} & \begin{bmatrix} \mathrm{HEAD} & \textit{det} \\ \mathrm{VAL} & \begin{bmatrix} \mathrm{COMPS} & \langle\ \rangle \\ \mathrm{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\mathrm{A} = \begin{bmatrix} \textit{word} & \\ \mathrm{SYN} & \begin{bmatrix} \mathrm{HEAD} & \textit{adj} \end{bmatrix} \end{bmatrix} \qquad \mathrm{ADV}_{pol} = \begin{bmatrix} \mathrm{SYN} & \begin{bmatrix} \mathrm{HEAD} & \textit{adv} \end{bmatrix} \\ \mathrm{SEM} & \begin{bmatrix} \mathrm{RESTR}\ \langle\ [\mathrm{RELN}\ \textbf{not} \mid \textbf{reaffirm}]\ \rangle \end{bmatrix} \end{bmatrix}$$

A.4 The Grammar Rules

(All daughters in our grammar rules are expressions, i.e. of type *word* or *phrase*; never of type *lexeme*).

(1) Head-Specifier Rule

$$\begin{bmatrix} \textit{phrase} & \\ \mathrm{SPR} & \langle\ \rangle \end{bmatrix} \rightarrow \boxed{1} \quad \mathbf{H}\begin{bmatrix} \mathrm{VAL} & \begin{bmatrix} \mathrm{SPR} & \langle\ \boxed{1}\ \rangle \\ \mathrm{COMPS} & \langle\ \rangle \end{bmatrix} \\ \mathrm{STOP\text{-}GAP} & \langle\ \rangle \end{bmatrix}$$

A phrase can consist of a (lexical or phrasal) head preceded by its specifier.

(2) Head-Complement Rule

$$\begin{bmatrix} \textit{phrase} & \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \rightarrow \mathbf{H}\begin{bmatrix} \textit{word} & \\ \text{COMPS} & \langle \boxed{1}, ..., \boxed{n} \rangle \end{bmatrix} \boxed{1} ... \boxed{n}$$

A phrase can consist of a lexical head followed by all its complements.

(3) Head-Modifier Rule

$$[\textit{phrase}] \rightarrow \mathbf{H}\boxed{1}\begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{STOP-GAP} & \langle\ \rangle \end{bmatrix} \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle \boxed{1} \rangle \end{bmatrix}$$

A phrase can consist of a (lexical or phrasal) head followed by a compatible modifier phrase.

(4) Coordination Rule

$$\begin{bmatrix} \text{FORM } \boxed{1} \\ \text{VAL } \boxed{0} \\ \text{GAP } \boxed{A} \\ \text{SEM [IND } s_0] \end{bmatrix} \rightarrow \begin{bmatrix} \text{FORM } \boxed{1} \\ \text{VAL } \boxed{0} \\ \text{GAP } \boxed{A} \\ \text{SEM [IND } s_1] \end{bmatrix} \ldots \begin{bmatrix} \text{FORM } \boxed{1} \\ \text{VAL } \boxed{0} \\ \text{GAP } \boxed{A} \\ \text{SEM [IND } s_{n-1}] \end{bmatrix}$$

$$\begin{bmatrix} \text{HEAD} & \textit{conj} \\ \text{IND} & s_0 \\ \text{RESTR} & \langle [\text{ARGS } \langle s_1, \ldots, s_n \rangle] \rangle \end{bmatrix} \begin{bmatrix} \text{FORM } \boxed{1} \\ \text{VAL } \boxed{0} \\ \text{GAP } \boxed{A} \\ \text{SEM [IND } s_n] \end{bmatrix}$$

Any number of elements with matching VAL, FORM and GAP specifications can form a coordinate phrase with identical VAL, FORM and GAP specifications.

(5) Imperative Rule

$$\begin{bmatrix} \textit{phrase} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \end{bmatrix} \\ \text{GAP} & \boxed{A} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{dir} \\ \text{INDEX} & s \end{bmatrix} \end{bmatrix} \rightarrow \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{INF} & - \\ \text{FORM} & \text{base} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \text{NP}\begin{bmatrix} \text{PER} & \text{2nd} \end{bmatrix} \right\rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \\ \text{GAP} & \boxed{A} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \end{bmatrix} \end{bmatrix}$$

An imperative phrase can consist of a (lexical or phrasal) VP whose FORM value is base and whose unexpressed subject is 2nd person.

(6) Head-Filler Rule

$$[\textit{phrase}] \;\rightarrow\; \boxed{1}\begin{bmatrix}\text{GAP} & \langle\ \rangle\end{bmatrix} \quad \text{H}\begin{bmatrix}\text{HEAD} & \begin{bmatrix}\textit{verb} & \\ \text{FORM} & \text{fin}\end{bmatrix} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix} \\ \text{STOP-GAP} & \langle\ \boxed{1}\ \rangle \\ \text{GAP} & \langle\ \boxed{1}\ \rangle\end{bmatrix}$$

A phrase can consist of a head with a gap preceded by an expression that meets whatever requirements the head places on that gap.

A.5 Lexical Rules

The following lexical rules obey the constraints provided earlier for feature structures of type *i-rule*, *d-rule*, and *pi-rule*:

(7) Singular Noun Lexical Rule

$$\begin{bmatrix}\textit{i-rule} & \\ \text{INPUT} & \left\langle \boxed{1}\ ,\ \textit{cn-lxm} \right\rangle \\ \text{OUTPUT} & \left\langle \boxed{1}\ ,\ \begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{AGR} & \begin{bmatrix}\text{NUM sg}\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix} \right\rangle\end{bmatrix}$$

(8) Plural Noun Lexical Rule

$$\begin{bmatrix}\textit{i-rule} & \\ \text{INPUT} & \left\langle \boxed{1}\ ,\ \textit{cntn-lxm} \right\rangle \\ \text{OUTPUT} & \left\langle \text{F}_{NPL}(\boxed{1})\ ,\ \begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{AGR} & \begin{bmatrix}\text{NUM} & \text{pl}\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix} \right\rangle\end{bmatrix}$$

(9) 3rd-Singular Verb Lexical Rule

$$\begin{bmatrix}\textit{i-rule} & \\ \text{INPUT} & \left\langle \boxed{3}\ ,\ \begin{bmatrix}\textit{verb-lxm} & \\ \text{SEM} & \begin{bmatrix}\text{RESTR} & \boxed{\text{A}}\end{bmatrix}\end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \text{F}_{3SG}(\boxed{3})\ ,\ \begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{FORM} & \text{fin} \\ \text{AGR} & \textit{3sing}\end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{RESTR} & \boxed{\text{A}} \oplus \ldots\end{bmatrix} \\ \text{ARG-ST} & \left\langle \begin{bmatrix}\text{CASE} & \text{nom}\end{bmatrix}, \ldots \right\rangle\end{bmatrix} \right\rangle\end{bmatrix}$$

(10) Non-3rd-Singular Verb Lexical Rule

$$\left[\begin{array}{ll}
\textit{i-rule} & \\
\text{INPUT} & \left\langle \boxed{1}\,, \left[\begin{array}{ll} \textit{verb-lxm} & \\ \text{SEM} & \left[\text{RESTR} \quad \boxed{A}\right] \end{array}\right] \right\rangle \\
\text{OUTPUT} & \left\langle \boxed{1}\,, \left[\begin{array}{ll} \text{SYN} & \left[\text{HEAD} \quad \left[\begin{array}{ll} \text{FORM} & \text{fin} \\ \text{AGR} & \textit{non-3sing} \end{array}\right]\right] \\ \text{SEM} & \left[\text{RESTR} \quad \boxed{A} \oplus ...\right] \\ \text{ARG-ST} & \langle \left[\text{CASE} \quad \text{nom}\right], ... \rangle \end{array}\right] \right\rangle
\end{array}\right]$$

(11) Past-Tense Verb Lexical Rule

$$\left[\begin{array}{ll}
\textit{i-rule} & \\
\text{INPUT} & \left\langle \boxed{3}\,, \left[\begin{array}{ll} \textit{verb-lxm} & \\ \text{SEM} & \left[\text{RESTR} \quad \boxed{A}\right] \end{array}\right] \right\rangle \\
\text{OUTPUT} & \left\langle \text{F}_{PAST}(\boxed{3})\,, \left[\begin{array}{ll} \text{SYN} & \left[\text{HEAD} \quad \left[\text{FORM} \quad \text{fin}\right]\right] \\ \text{SEM} & \left[\text{RESTR} \quad \boxed{A} \oplus ...\right] \\ \text{ARG-ST} & \left\langle \left[\text{CASE} \quad \text{nom}\right], ... \right\rangle \end{array}\right] \right\rangle
\end{array}\right]$$

(12) Base Form Lexical Rule

$$\left[\begin{array}{ll}
\textit{i-rule} & \\
\text{INPUT} & \left\langle \boxed{1}\,, \textit{verb-lxm} \right\rangle \\
\text{OUTPUT} & \left\langle \boxed{1}\,, \left[\text{SYN} \quad \left[\text{HEAD} \quad \left[\text{FORM} \quad \text{base}\right]\right]\right] \right\rangle
\end{array}\right]$$

(13) Constant Lexeme Lexical Rule

$$\left[\begin{array}{ll}
\textit{i-rule} & \\
\text{INPUT} & \left\langle \boxed{1}\,, \textit{const-lxm} \right\rangle \\
\text{OUTPUT} & \left[\text{FIRST} \quad \boxed{1}\right]
\end{array}\right]$$

(14) Present Participle Lexical Rule

$$
\begin{bmatrix}
\textit{d-rule} & \\
\text{INPUT} & \left\langle \boxed{3}\ , \begin{bmatrix} \textit{verb-lxm} & \\ \text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{PRED} & -\end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{RESTR} & \boxed{A}\end{bmatrix} \\ \text{ARG-ST} & \boxed{B} \end{bmatrix} \right\rangle \\
\text{OUTPUT} & \left\langle \text{F}_{PRP}(\boxed{3})\ , \begin{bmatrix} \textit{part-lxm} & \\ \text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{FORM} & \text{prp} \\ \text{PRED} & + \end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{RESTR} & \boxed{A} \oplus ...\end{bmatrix} \\ \text{ARG-ST} & \boxed{B} \end{bmatrix} \right\rangle
\end{bmatrix}
$$

(15) Past Participle Lexical Rule

$$
\begin{bmatrix}
\textit{d-rule} & \\
\text{INPUT} & \left\langle \boxed{3}\ , \begin{bmatrix} \textit{verb-lxm} & \\ \text{SEM} & \begin{bmatrix}\text{RESTR} & \boxed{A}\end{bmatrix} \\ \text{ARG-ST} & \boxed{B} \end{bmatrix} \right\rangle \\
\text{OUTPUT} & \left\langle \text{F}_{PSP}(\boxed{3})\ , \begin{bmatrix} \textit{part-lxm} & \\ \text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{FORM} & \text{psp}\end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{RESTR} & \boxed{A} \oplus ...\end{bmatrix} \\ \text{ARG-ST} & \boxed{B} \end{bmatrix} \right\rangle
\end{bmatrix}
$$

(16) Passive Lexical Rule

$$\begin{bmatrix} \textit{d-rule} & \\ \text{INPUT} & \left\langle \boxed{1}\ , \begin{bmatrix} \textit{tv-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{PRED} & - \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \langle\ [\text{INDEX } i]\ \rangle \oplus \boxed{\text{A}} \end{bmatrix} \right\rangle \\ \text{OUPUT} & \left\langle \text{F}_{PSP}(\boxed{1})\ , \begin{bmatrix} \textit{part-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{pass} \\ \text{PRED} & + \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \boxed{\text{A}} \oplus \left\langle \left(\begin{matrix} \text{PP} \\ \begin{bmatrix} \text{FORM} & \text{by} \\ \text{INDEX} & i \end{bmatrix} \end{matrix} \right) \right\rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

(17) Agent Nominalization Lexical Rule

$$\begin{bmatrix} \textit{d-rule} & \\ \text{INPUT} & \left\langle \boxed{2}\ , \begin{bmatrix} \textit{verb-lxm} & \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \end{bmatrix} \\ \text{ARG-ST} & \langle\ \text{X}_i\ , \text{NP}_j\ \rangle \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \text{F}_{-er}(\boxed{2})\ , \begin{bmatrix} \textit{cntn-lxm} & \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & i \end{bmatrix} \\ \text{ARG-ST} & \left\langle \text{Y} \left(, \begin{matrix} \text{PP}_j \\ \begin{bmatrix} \text{FORM} & \text{of} \end{bmatrix} \end{matrix} \right) \right\rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

(18) Extraposition Lexical Rule

$$\begin{bmatrix} \textit{pi-rule} & \\ \text{INPUT} & \left\langle \text{X}\ , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \boxed{2}\text{CP}\ \rangle \\ \text{COMPS} & \boxed{\text{A}} \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \text{Y}\ , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \text{NP[FORM it]}\ \rangle \\ \text{COMPS} & \boxed{\text{A}} \oplus \langle \boxed{2} \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

(19) ADV$_{pol}$-Addition Lexical Rule

$$
\left[\begin{array}{ll}
\textit{pi-rule} & \\
\text{INPUT} & \left\langle \text{X}\ ,\ \left[\begin{array}{ll}
\text{SYN} & \left[\begin{array}{ll}\text{HEAD} & \left[\begin{array}{ll}\textit{verb} & \\ \text{FORM} & \text{fin} \\ \text{POL} & - \\ \text{AUX} & + \end{array}\right]\end{array}\right] \\
\text{ARG-ST} & \langle\ \boxed{1}\ \rangle \oplus \boxed{\text{A}} \\
\text{SEM} & \left[\begin{array}{ll}\text{INDEX} & s_1\end{array}\right]
\end{array}\right] \right\rangle \\
\text{OUTPUT} & \left\langle \text{Y}\ ,\ \left[\begin{array}{ll}
\text{SYN} & \left[\begin{array}{ll}\text{HEAD} & \left[\begin{array}{ll}\text{POL} & +\end{array}\right] \\ \text{VAL} & \left[\begin{array}{ll}\text{SPR} & \langle\ \text{Z}\ \rangle\end{array}\right]\end{array}\right] \\
\text{ARG-ST} & \langle\ \boxed{1}\ \rangle \oplus \left\langle \overset{\text{ADV}_{pol}}{\left[\begin{array}{ll}\text{INDEX} & s_2 \\ \text{RESTR} & \left\langle \left[\begin{array}{ll}\text{ARG} & s_1\end{array}\right] \right\rangle\end{array}\right]} \right\rangle \oplus \boxed{\text{A}} \\
\text{SEM} & \left[\begin{array}{ll}\text{INDEX} & s_2\end{array}\right]
\end{array}\right] \right\rangle
\end{array}\right]
$$

(20) Inversion Lexical Rule

$$
\left[\begin{array}{ll}
\textit{pi-rule} & \\
\text{INPUT} & \left\langle \text{X}\ ,\ \left[\begin{array}{ll}
\text{SYN} & \left[\begin{array}{ll}\text{HEAD} & \left[\begin{array}{ll}\textit{verb} & \\ \text{FORM} & \text{fin} \\ \text{AUX} & + \end{array}\right]\end{array}\right] \\
\text{ARG-ST} & \boxed{\text{A}} \\
\text{SEM} & \left[\begin{array}{ll}\text{MODE} & \text{prop}\end{array}\right]
\end{array}\right] \right\rangle \\
\text{OUTPUT} & \left\langle \text{Y}\ ,\ \left[\begin{array}{ll}
\text{SYN} & \left[\begin{array}{ll}\text{HEAD} & \left[\begin{array}{ll}\text{INV} & +\end{array}\right] \\ \text{VAL} & \left[\begin{array}{ll}\text{SPR} & \langle\ \rangle\end{array}\right]\end{array}\right] \\
\text{ARG-ST} & \boxed{\text{A}} \\
\text{SEM} & \left[\begin{array}{ll}\text{MODE} & \text{ques}\end{array}\right]
\end{array}\right] \right\rangle
\end{array}\right]
$$

(21) Contraction Lexical Rule

$$\begin{bmatrix} \textit{pi-rule} & \\ \text{INPUT} & \left\langle \boxed{2}\ , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{fin} \\ \text{AUX} & + \\ \text{POL} & - \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \boxed{\text{B}} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s_1 \\ \text{RESTR} & \boxed{\text{A}} \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \text{F}_{NEG}(\boxed{2})\ , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{POL} & + \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \text{X}\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \boxed{\text{B}} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s_2 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{not} \\ \text{SIT} & s_2 \\ \text{ARG} & s_1 \end{bmatrix} \right\rangle \oplus \boxed{\text{A}} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

(22) Ellipsis Lexical Rule

$$\begin{bmatrix} \textit{d-rule} & \\ \text{INPUT} & \left\langle \boxed{1}\ , \begin{bmatrix} \textit{auxv-lxm} & \\ \text{ARG-ST} & \langle\ \boxed{2}\ \rangle \ \oplus\ \boxed{\text{A}} \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \boxed{1}\ , \begin{bmatrix} \textit{dervv-lxm} & \\ \text{ARG-ST} & \langle\ \boxed{2}\ \rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

(23) Subject Extraction Lexical Rule

$$\begin{bmatrix} \textit{pi-rule} \\ \text{INPUT} & \left\langle \text{X}, \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} \\ \text{FORM} & \text{fin} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \text{ Z } \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \boxed{\text{A}} \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \text{Y}, \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \end{bmatrix} \\ \text{GAP} & \langle \boxed{1} \rangle \end{bmatrix} \\ \text{ARG-ST} & \boxed{\text{A}} \langle \boxed{1}, \ldots \rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

A.6 The Basic Lexicon

Here are some sample lexical entries that are part of the basic lexicon. Each entry is a pair consisting of (1) a description of a phonological form and (2) a description satisfiable by feature structures of (some maximal subtype of) *lexeme*. Lexical entries include only information that is not inherited from other types. As before, the notation '...' indicates things we haven't dealt with but which a complete grammar would have to.

A.6.1 Nouns

(24)

$$\left\langle \text{she}, \begin{bmatrix} \textit{pron-lxm} \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{CASE} & \text{nom} \\ \text{AGR} & \begin{bmatrix} \textit{3sing} \\ \text{GEND} & \text{fem} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{female} \\ \text{INST} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(25)

$$\left\langle \text{him}, \begin{bmatrix} \textit{pron-lxm} \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{CASE} & \text{acc} \\ \text{AGR} & \begin{bmatrix} \textit{3sing} \\ \text{GEND} & \text{masc} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{male} \\ \text{INST} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(26)
$$\left\langle \text{themselves}\ , \begin{bmatrix} \textit{pron-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{CASE} & \text{acc} \\ \text{AGR} & \begin{bmatrix} \textit{plural} & \\ \text{PER} & \text{3rd} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ana} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{group} \\ \text{INST} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(27)
$$\left\langle \text{Kim}\ , \begin{bmatrix} \textit{pn-lxm} & \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{Kim} \\ \text{NAMED} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(28)
$$\left\langle \text{book}\ , \begin{bmatrix} \textit{cntn-lxm} & \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{book} \\ \text{INST} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(29)
$$\left\langle \text{advantage}\ , \begin{bmatrix} \textit{massn-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{advantage} \\ \text{AGR} & \textit{3sing} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{none} \\ \text{INDEX} & \text{none} \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(30)
$$\left\langle \text{tabs}\ , \begin{bmatrix} \textit{cntn-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{tabs} \\ \text{AGR} & \begin{bmatrix} \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{none} \\ \text{INDEX} & \text{none} \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(31)

$$\left\langle \text{there}\ ,\ \begin{bmatrix} \textit{pron-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{there} \\ \text{AGR} & \begin{bmatrix} \text{PER 3rd} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{none} \\ \text{INDEX} & \text{none} \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(32)

$$\left\langle \text{it}\ ,\ \begin{bmatrix} \textit{pron-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{it} \\ \text{AGR} & \textit{3sing} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{none} \\ \text{INDEX} & \text{none} \\ \text{RESTR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

A.6.2 Verbs

(33)

$$\left\langle \text{die}\ ,\ \begin{bmatrix} \textit{siv-lxm} & \\ \text{ARG-ST} & \langle\ \text{NP}_i\ \rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{die} \\ \text{SIT} & s \\ \text{CORPSE} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(34)

$$\left\langle \text{love}\ ,\ \begin{bmatrix} \textit{stv-lxm} & \\ \text{ARG-ST} & \langle\ \text{NP}_i\ ,\ \text{Y}_j\ \rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{love} \\ \text{SIT} & s \\ \text{LOVER} & i \\ \text{LOVED} & j \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(35)
$$\left\langle \text{give}\ , \begin{bmatrix} \textit{dtv-lxm} & \\ \text{ARG-ST} & \langle\ \mathrm{X}_i\ ,\ \mathrm{Y}_j\ ,\ \mathrm{Z}_k\ \rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{give} \\ \text{SIT} & s \\ \text{GIVER} & i \\ \text{GIVEN} & j \\ \text{GIFT} & k \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(36)
$$\left\langle \text{give}\ , \begin{bmatrix} \textit{ptv-lxm} & \\ \text{ARG-ST} & \langle\ \mathrm{X}_i\ ,\ \mathrm{NP}_k\ ,\ \mathrm{Z}_j[\text{FORM to}]\ \rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{give} \\ \text{SIT} & s \\ \text{GIVER} & i \\ \text{GIVEN} & j \\ \text{GIFT} & k \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(37)
$$\left\langle \text{matter}\ , \begin{bmatrix} \textit{siv-lxm} & \\ \text{ARG-ST} & \left\langle \begin{bmatrix} \text{SEM [INDEX } \boxed{1}] \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{matter} \\ \text{SIT} & s \\ \text{MATTERING} & \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(38)
$$\left\langle \text{keep}\ , \begin{bmatrix} \textit{ptv-lxm} & \\ \text{ARG-ST} & \left\langle \mathrm{NP}_i\ , \begin{bmatrix} \text{FORM tabs} \end{bmatrix}, \begin{bmatrix} \text{FORM} & \text{on} \\ \text{INDEX} & j \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{observe} \\ \text{SIT} & s \\ \text{OBSERVER} & i \\ \text{OBSERVED} & j \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(39)
$$\left\langle \text{take}, \begin{bmatrix} \textit{ptv-lxm} & \\ \text{ARG-ST} & \left\langle \text{NP}_i , \begin{bmatrix}\text{FORM advantage}\end{bmatrix}, \begin{bmatrix}\text{FORM} & \text{of}\\ \text{INDEX} & j\end{bmatrix}\right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{exploit} \\ \text{SIT} & s \\ \text{EXPLOITER} & i \\ \text{EXPLOITED} & j\end{bmatrix}\right\rangle\end{bmatrix}\end{bmatrix}\right\rangle$$

(40)
$$\left\langle \text{to}, \begin{bmatrix} \textit{auxv-lxm} & \\ \text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{INF} & + \\ \text{FORM} & \text{base}\end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s \\ \text{RESTR} & \langle\ \rangle\end{bmatrix} \\ \text{ARG-ST} & \left\langle \text{X} , \begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{INF} & - \\ \text{FORM} & \text{base}\end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s\end{bmatrix}\end{bmatrix}\right\rangle \end{bmatrix}\right\rangle$$

(41)
$$\left\langle \text{continue}, \begin{bmatrix} \textit{ic-srv-lxm} & \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s \\ \text{RESTR} & \left\langle\begin{bmatrix}\text{RELN} & \textbf{continue} \\ \text{SIT} & s\end{bmatrix}\right\rangle\end{bmatrix}\end{bmatrix}\right\rangle$$

(42)
$$\left\langle \text{try}, \begin{bmatrix} \textit{scv-lxm} & \\ \text{ARG-ST} & \left\langle \text{NP}_i , \overset{\text{VP}}{\begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{INF} & +\end{bmatrix}\end{bmatrix}\end{bmatrix}}\right\rangle \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s \\ \text{RESTR} & \left\langle\begin{bmatrix}\text{RELN} & \textbf{try} \\ \text{SIT} & s \\ \text{TRIER} & i\end{bmatrix}\right\rangle\end{bmatrix}\end{bmatrix}\right\rangle$$

(43)
$$\left\langle \text{expect}\ ,\ \begin{bmatrix} \textit{orv-lxm} & \\ \text{ARG-ST} & \left\langle \text{NP}_j\ ,\ \text{X}\ ,\ \overset{\text{VP}}{\begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{INF} & +\end{bmatrix}\end{bmatrix}\end{bmatrix}} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix}\text{RELN} & \textbf{expect}\\ \text{SIT} & s \\ \text{EXPECTER} & j\end{bmatrix}\right\rangle\end{bmatrix}\end{bmatrix}\right\rangle$$

(44)
$$\left\langle \text{persuade}\ ,\ \begin{bmatrix} \textit{ocv-lxm} & \\ \text{ARG-ST} & \left\langle \text{NP}_j\ ,\ \text{NP}_i\ ,\ \overset{\text{VP}}{\begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{INF} & +\end{bmatrix}\end{bmatrix}\end{bmatrix}} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix}\text{RELN} & \textbf{persuade}\\ \text{SIT} & s \\ \text{PERSUADER} & j \\ \text{PERSUADEE} & i\end{bmatrix}\right\rangle\end{bmatrix}\end{bmatrix}\right\rangle$$

(45)
$$\left\langle \text{be}\ ,\ \begin{bmatrix} \textit{auxv-lxm} & \\ \text{ARG-ST} & \left\langle \text{X}\ ,\ \begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{PRED} & +\end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s\end{bmatrix}\end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \langle\ \rangle\end{bmatrix}\end{bmatrix}\right\rangle$$

(46)
$$\left\langle \text{be}\ ,\ \begin{bmatrix} \textit{orv-lxm} & \\ \text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{AUX} & +\end{bmatrix}\end{bmatrix} \\ \text{ARG-ST} & \left\langle \begin{bmatrix}\text{FORM} & \text{there}\end{bmatrix}, \text{X}\ ,\ \begin{bmatrix}\text{PRED} & + \\ \text{SEM} & \begin{bmatrix}\text{INDEX} & s\end{bmatrix}\end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \langle\ \rangle\end{bmatrix}\end{bmatrix}\right\rangle$$

(47)

$$\left\langle \text{have} , \begin{bmatrix} \textit{auxv-lxm} & \\ \text{ARG-ST} & \left\langle \text{X} , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{psp} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s_2 \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s_1 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{have} \\ \text{SIT} & s_1 \\ \text{ARG} & s_2 \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(48)

$$\left\langle \text{can} , \begin{bmatrix} \textit{auxv-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{fin} \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \text{X} , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{INF} & - \\ \text{FORM} & \text{base} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s_2 \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s_1 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{can} \\ \text{SIT} & s_1 \\ \text{ARG} & s_2 \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(49)

$$\left\langle \text{do} , \begin{bmatrix} \textit{auxv-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{fin} \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \text{X} , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{base} \\ \text{AUX} & - \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \langle \ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

A.6.3 Miscellaneous

(50)
$$\left\langle \text{the}\ ,\ \begin{bmatrix} \textit{det-lxm} & \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{the} \\ \text{BV} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(51)
$$\left\langle \text{few}\ ,\ \begin{bmatrix} \textit{det-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUM pl} \end{bmatrix} \\ \text{COUNT} & + \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{few} \\ \text{BV} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(52)
$$\left\langle \text{'s}\ ,\ \begin{bmatrix} \textit{det-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \text{NP}\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{the} \\ \text{BV} & i \end{bmatrix}, \ldots \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(53)
$$\left\langle \text{to}\ ,\ \begin{bmatrix} \textit{argmkp-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{to} \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle$$

(54)
$$\left\langle \text{in}\ ,\ \begin{bmatrix} \textit{predp-lxm} & \\ \text{ARG-ST} & \langle\ \text{NP}_i\ ,\ \text{NP}_j\ \rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{in} \\ \text{SIT} & s \\ \text{CONTAINER} & j \\ \text{CONTAINED} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(55)

$$\left\langle \text{and}\ , \begin{bmatrix} \textit{conj-lxm} & \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{and} \\ \text{SIT} & s \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(56)

$$\left\langle \text{today}\ , \begin{bmatrix} \textit{adv-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \left\langle \begin{array}{c} \text{VP} \\ \begin{bmatrix} \text{INDEX} & s \end{bmatrix} \end{array} \right\rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{today} \\ \text{ARG} & s \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(57)

$$\left\langle \text{not}\ , \begin{bmatrix} \textit{adv-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \left\langle \begin{array}{c} \text{VP} \\ \begin{bmatrix} \text{INDEX} & s \end{bmatrix} \end{array} \right\rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{today} \\ \text{ARG} & s \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(58)

$$\left\langle \text{that}\ , \begin{bmatrix} \textit{comp-lxm} & \\ \text{ARG-ST} & \langle \begin{bmatrix} \text{FORM fin} \end{bmatrix} \rangle \\ \text{SEM} & \begin{bmatrix} \text{MODE prop} \end{bmatrix} \end{bmatrix} \right\rangle$$

(59)

$$\left\langle \text{easy}\ , \begin{bmatrix} \textit{adj-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{STOP-GAP} & \langle \boxed{1} \rangle \end{bmatrix} \\ \text{ARG-ST} & \left\langle \text{NP}_i\ , \begin{array}{c} \text{VP} \\ \begin{bmatrix} \text{INF} & + \\ \text{GAP} & \langle \boxed{1}\text{NP}_i\ , \ldots \rangle \end{bmatrix} \end{array} \right\rangle \end{bmatrix} \right\rangle$$

A.7 Well-Formed Structures

In this section, we lay out more precisely the constructs of the theory whose effects we have been illustrating in the preceding chapters.

A.7.1 Preliminaries

According to our approach, a grammar G is defined by the following components:

- a finite set of features: $\mathcal{F} = \{\text{SYN, SEM, HEAD, AGR, } \ldots\}$,
- a finite set of primitive items:
  $\mathcal{A}_{atom} = \mathcal{A}_{pol} \cup \mathcal{A}_{gr.atom} \cup \mathcal{A}_{mode} \cup \mathcal{A}_{reln}$, where:
  1. $\mathcal{A}_{pol} = \{+, -\}$,
  2. (a set of ground atoms) $\mathcal{A}_{gr.atom} = \{\text{1st}, \text{2nd}, \text{3rd}, \text{sg}, \text{pl}, \ldots, \text{run}, \text{dog}, \ldots\}$,
  3. $\mathcal{A}_{mode} = \{\text{prop}, \text{ques}, \text{dir}, \text{ref}, \text{none}\}$, and
  4. $\mathcal{A}_{reln} = \{\textbf{walk}, \textbf{love}, \textbf{person}, \ldots\}$,
- a denumerably infinite set of primitive items: $\mathcal{A}_{index} = \mathcal{A}_{ind} \cup \mathcal{A}_{sit}$, where:
  1. $\mathcal{A}_{ind} = \{i, j, \ldots\}$ and
  2. $\mathcal{A}_{sit} = \{s_1, s_2, \ldots\}$,
- the distinguished element *elist* (*empty-list*),
- a finite set of types: $\mathcal{T} = \{$*noun*, *agr-pos*, *plural*, *expression*, ...$\}$,
- a type hierarchy with a tree structure associated with constraint inheritance (for instance, the type hierarchy represented by the tree in Section A.1 and the table in A.2).
- a set $\mathcal{LT} \subset \mathcal{T}$ called the *leaf types* (a type τ is a *leaf type* if it is associated with a leaf in the type hierarchy tree, i.e. if τ is one of the most specific types),
- a set of list types (if τ is a type, then *list*(τ) is a type),
- a set of grammar rules (like those in Section A.4),
- a set of principles,
- a lexicon (which is a finite set of lexical entries like those in Section A.6), and
- a set of lexical rules (like those in Section A.5).

Thus a grammar G comes with various primitives grouped into two sets: $\mathcal{A}_{atom}$ ($\mathcal{A}_{pol}$, $\mathcal{A}_{gr.atom}$, $\mathcal{A}_{mode}$, $\mathcal{A}_{reln}$) and $\mathcal{A}_{index}$ ($\mathcal{A}_{ind}$, $\mathcal{A}_{sit}$). G assigns the type *atom* to all elements of $\mathcal{A}_{atom}$. The elements of $\mathcal{A}_{index}$ are used by the grammar for describing individual objects and situations; they are associated with the leaf type *index*. We assume that no items in these sets of primitives can be further analyzed via grammatical features.

Our grammar appeals to several ancillary notions which we now explicate: *feature structure description*, *feature structure*, *satisfaction of a description*, and *tree structure*.

A.7.2 Feature Structure Descriptions

For expressing the constraints associated with the grammar rules, principles, types, and lexical entries, we introduce the notion of a feature structure description. The feature structure descriptions are given as attribute-value matrices, augmented with the connective '|', set descriptors ($\{\ldots\}$), list descriptions ($\langle\ldots\rangle$, attribute-value matrices with FIRST/REST, or two list descriptions connected by $\oplus$), and a set *Tags* of tags (labels represented by boxed integers or letters).

A.7.3 Feature Structures

The set of the feature structures $\mathcal{FS}$ is given by the following recursive definition:

(60) $\phi \in \mathcal{FS}$ (i.e. ϕ is a feature structure) iff

a. $\phi \in \mathcal{A}_{atom} \cup \mathcal{A}_{index}$, or

b. ϕ is a function from features to feature structures, $\phi : \mathcal{F} \longrightarrow \mathcal{FS}$ satisfying the following conditions

1. ϕ is of a leaf type τ;
2. $DOM(\phi) = \{F \mid \text{G declares } F \text{ appropriate for } \tau\} \cup$

$$\{F' \mid \exists\tau' \text{ such that } \tau' \text{ is a supertype of } \tau \text{ and}$$
$$\text{G declares } F' \text{ appropriate for } \tau'\},$$

i.e. ϕ is defined for any feature that is declared appropriate for τ or for any of τ's supertypes;

3. for each $F \in DOM(\phi)$, G defines the type of the value $\phi(F)$ (we call the value $\phi(F)$ of the function ϕ on F the value of the feature F); and
4. ϕ obeys all further constraints ('type constraints') that G associates with type τ (including those inherited by default from the supertypes τ' of τ), or

c. ϕ is of type *list*(τ), for some type τ, in which case either:

1. ϕ is the distinguished element *elist*, or else:
2. A. *DOM*(ϕ) is $\{\text{FIRST}, \text{REST}\}$,
   B. the type of ϕ(FIRST) is τ, and
   C. the type of ϕ(REST) is *list*(τ).

A.7.4 Satisfaction

We explain how feature structures satisfy descriptions indirectly – in terms of denotation, which we define as follows:

Denotation of Feature Structure Descriptions

The denotation of a feature structure description is specified in terms of a structure $\mathcal{M}$:

(61) $\mathcal{M} = \langle \mathcal{A}, \mathcal{F}, \mathcal{T}, \textbf{Type}, I \rangle$, where:

1. $\mathcal{A} = \mathcal{A}_{atom} \cup \mathcal{A}_{index} \cup \{elist\}$,
2. $\mathcal{F}$ is a finite set of features,
3. $\mathcal{T}$ is a finite set of types,

4. **Type** is a function mapping feature structures to types –
Type : $\mathcal{FS} \longrightarrow \mathcal{LT}$, where $\mathcal{LT}$ is the set of the leaf types, and
5. I is a function mapping feature names and atomic descriptors to features and atoms of the appropriate sort:

$$I = I_{\widetilde{\mathcal{F}}} \cup I_{\widetilde{\mathcal{A}}_{atom}} \cup I_{\widetilde{\mathcal{A}}_{ind}} \cup I_{\widetilde{\mathcal{A}}_{sit}} \cup \{\langle elist, elist \rangle\},$$

where
$I_{\widetilde{\mathcal{F}}} \in \mathcal{F}^{\widetilde{\mathcal{F}}}$, $I_{\widetilde{\mathcal{A}}_{atom}} \in \mathcal{A}_{atom}{}^{\widetilde{\mathcal{A}}_{atom}}$, $I_{\widetilde{\mathcal{A}}_{ind}} \in \mathcal{A}_{ind}{}^{\widetilde{\mathcal{A}}_{ind}}$, $I_{\widetilde{\mathcal{A}}_{sit}} \in \mathcal{A}_{sit}{}^{\widetilde{\mathcal{A}}_{sit}}$,

and $\widetilde{X}$ denotes the set of expressions that have denotations in the set X.

The function I is called an *interpretation function.* An *assignment function* is a function

$$g : Tags \longrightarrow \mathcal{FS}.$$

We say that a feature structure ϕ is of a type $\tau \in \mathcal{T}$ iff there is a (unique) leaf type $\tau' \in \mathcal{LT}$ such that:

(62)
1. τ' is a subtype of τ, and
2. $\mathbf{Type}(\phi) = \tau'$.

Given $\mathcal{M}$, the interpretation $[\![d]\!]^{\mathcal{M},g}$ of a feature structure description d with respect to an assignment function g is defined recursively as follows:

(63)
1. if $v \in \widetilde{\mathcal{F}} \cup \widetilde{\mathcal{A}}_{atom} \cup \widetilde{\mathcal{A}}_{index} \cup \{elist\}$, *then* $[\![v]\!]^{\mathcal{M},g} = I(v)$;
2. if τ is a type, i.e. $\tau \in \mathcal{T}$, then $[\![\tau]\!]^{\mathcal{M},g} = \{\phi \in \mathcal{FS} : \phi$ is of type $\tau\}$;
3. if $F \in \widetilde{\mathcal{F}}$, and d is a feature structure description, then $[\![[F\ d]]\!]^{\mathcal{M},g} =$
$\{\phi \in \mathcal{FS}$: there is some ϕ' such that $\phi' \in [\![d]\!]^{\mathcal{M},g}$ and $\langle I(F), \phi' \rangle \in \phi\}$;[1]
4. if $d = \begin{bmatrix} d_1 \\ \dots \\ d_n \end{bmatrix}$
where $n \geq 1$, and $d_1, \dots, d_n$ are feature structure descriptions, then
$$[\![d]\!]^{\mathcal{M},g} = \bigcap_{i=1}^{n} [\![d_i]\!]^{\mathcal{M},g};$$
5. if d is a set descriptor $\{d_1, \dots, d_n\}$, then
$$[\![d]\!]^{\mathcal{M},g} = \bigcup_{i=1}^{n} [\![d_i]\!]^{\mathcal{M},g}$$
$([\![\{\ \}]\!]^{\mathcal{M},g} = \emptyset)$;
6. $[\![d_1 \mid d_2]\!]^{\mathcal{M},g} = [\![d_1]\!]^{\mathcal{M},g} \cup [\![d_2]\!]^{\mathcal{M},g}$;
7. if $d \in Tags$, then $[\![d]\!]^{\mathcal{M},g} = g(d)$;
8. if $d \in Tags$ and d' is a feature structure description, then
$[\![d\ d']\!]^{\mathcal{M},g} = \{\phi \in \mathcal{FS} : g(d) = \phi$ and $\phi \in [\![d']\!]^{\mathcal{M},g}\}$;
(Note that tagging narrows the interpretation down to a singleton set.)

[1]Note that the definition of a feature structure in A.7.3, taken together with this clause, ensures that each element ϕ of the set $[\![[F\ d]]\!]^{\mathcal{M},g}$ is a proper feature structure.

9. List Addition:[2]

   a. $[\![\mathit{elist} \oplus d]\!]^{\mathcal{M},g} = [\![d]\!]^{\mathcal{M},g}$,

   b. if $d = \begin{bmatrix} \text{FIRST} & d_1 \\ \text{REST} & d_2 \end{bmatrix} \oplus d_3$,

   then $[\![d]\!]^{\mathcal{M},g} =$
   $\{\phi \in \mathcal{FS} : \phi(\text{FIRST}) \in [\![d_1]\!]^{\mathcal{M},g} \text{ and } \phi(\text{REST}) \in [\![d_2 \oplus d_3]\!]^{\mathcal{M},g}\}$.

Satisfaction of Feature Structure Descriptions

A feature structure $\phi \in \mathcal{FS}$ satisfies a feature structure description d iff there is some assignment function g such that $\phi \in [\![d]\!]^{\mathcal{M},g}$.

We make no attempt here to extend this definition to include the satisfaction of defeasible constraints. See footnote 5 of Chapter 9. For examples of feature structures that satisfy particular descriptions, see Section 6.3.4 of Chapter 6.

A.7.5 Tree Structures

Finally, we assume a notion of tree structure described informally as follows:

(64) A tree structure is a directed graph that satisfies a number of conditions:[3]

1. it has a unique root node,
2. each non-root node has exactly one mother,
3. sister nodes are ordered with respect to each other,
4. it has no crossing branches,
5. each nonterminal node is labeled by a feature structure, and
6. each terminal node is labeled by a phonological form (an atom).

A.7.6 Structures Defined by the Grammar

(65) **Well-Formed Tree Structure**:

Φ is a Well-Formed Tree Structure according to G if and only if:

1. Φ is a tree structure,
2. the label of Φ's root node satisfies the constraint:

$$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \mathit{verb} \\ \text{FORM} & \text{fin} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \end{bmatrix} \\ \text{GAP} & \langle\ \rangle \end{bmatrix} \end{bmatrix}$$

   and

3. each local subtree within Φ is either phrasally licensed or lexically licensed.

[2]Where no confusion should arise, we use 'FIRST', 'SYN', etc. to refer either to the appropriate feature (an element of $\mathcal{F}$) or to its name (an element of $\widetilde{\mathcal{F}}$).

[3]Again, we assume familiarity with notions such as *root*, *mother*, *terminal node*, *non-terminal node*, and *branches*. See footnote 16 of Chapter 6.

Lexical Licensing is defined in terms of lexical sequences that are legitimate outputs of lexical rules. The instances of the type *lexical-sequence* are defined as follows:

(66) **Lexical Sequences**:
$\langle \omega, \phi \rangle$ is a lexical sequence if and only if ω is a phonological form (an atom), ϕ is a feature structure, and either:

1. G contains some lexical entry $\langle d_1, d_2 \rangle$ such that ω satisfies d_1 and ϕ satisfies d_2, or
2. there is some lexical rule instantiation licensed by G (a feature structure of type *l-rule*) whose OUTPUT value is $\langle \omega, \phi \rangle$.

(67) **Lexical Licensing**:
A word structure of the form:

$$\begin{array}{c} \phi \\ | \\ \omega \end{array}$$

is licensed if and only if:

1. $\langle \omega, \phi \rangle$ is a lexical sequence, where ϕ is of type *word*,
2. (Case Constraint:) An outranked NP or CP is [CASE acc], and
3. ϕ satisfies the Binding Theory.

(68) **The Binding Theory**:

Principle A: A [MODE ana] expression must be outranked by a coindexed element.
Principle B: A [MODE ref] expression must not be outranked by a coindexed element;
where:

(i) If a node is coindexed with its daughter, their feature structures are of equal rank.
(ii) If there is an ARG-ST list on which A precedes B, then A outranks B.

(69) **Phrasal Licensing**:
A grammar rule $\rho = \mathrm{d}_0 \rightarrow \mathrm{d}_1 \ldots \mathrm{d}_n$ licenses a local subtree:

$$\Phi = \begin{array}{ccc} & \phi_0 & \\ \swarrow & & \searrow \\ \phi_1 & \ldots & \phi_n \end{array}$$

if and only if:

1. for each $i, 0 \leq i \leq n$, ϕ_i is of type *expression*,
2. there is some assignment function g under which the sequence $\langle \phi_0, \phi_1, ..., \phi_n \rangle$ satisfies the description sequence $\langle d_0, d_1, \ldots, d_n \rangle$,
3. Φ satisfies the Semantic Compositionality Principle and the Anaphoric Agreement Principle, and
4. if ρ is a headed rule, then Φ satisfies the Head Feature Principle, the Valence Principle, the Semantic Inheritance Principle, and the GAP Principle with respect to ρ.

(70) Φ satisfies the Semantic Compositionality Principle with respect to a grammar rule ρ if and only if Φ satisfies:

$$\begin{array}{c} \begin{bmatrix}\text{RESTR} & \boxed{\text{A}_1} \oplus ... \oplus \boxed{\text{A}_\text{n}}\end{bmatrix} \\ \begin{array}{ccc}[\text{RESTR}\ \boxed{\text{A}_1}] & \ldots & [\text{RESTR}\ \boxed{\text{A}_\text{n}}]\end{array}\end{array}$$

(71) **Anaphoric Agreement Principle**:
Coindexed NPs agree (i.e. their AGR values are identical).

(72) Φ satisfies the Head Feature Principle with respect to a headed rule ρ if and only if Φ satisfies:

$$\begin{array}{c} [\text{HEAD}\ \boxed{1}] \\ \begin{array}{ccc} \ldots & \begin{array}{c}\phi_h \\ \begin{bmatrix}\text{HEAD} & \boxed{1}\end{bmatrix}\end{array} & \ldots \end{array}\end{array}$$

where ϕ_h is the head daughter of Φ.

(73) Φ satisfies the Valence Principle with respect to a headed rule ρ if and only if, for any VAL feature F, Φ satisfies:

$$\begin{array}{c} \begin{bmatrix}\text{F} & \boxed{\text{A}}\end{bmatrix} \\ \begin{array}{ccc} \ldots & \begin{array}{c}\phi_h \\ \begin{bmatrix}\text{F} & \boxed{\text{A}}\end{bmatrix}\end{array} & \ldots \end{array}\end{array}$$

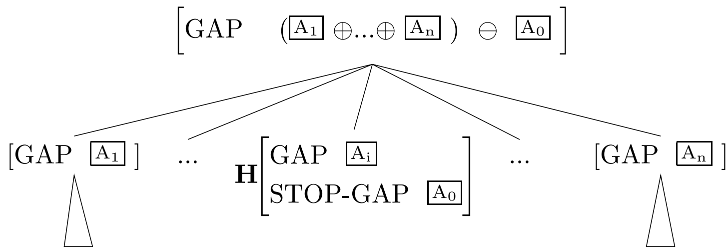
where ϕ_h is the head daughter of Φ and ρ does not specify incompatible F values for ϕ_h and ϕ_0.

(74) Φ satisfies the Semantic Inheritance Principle with respect to a headed rule ρ if and only if Φ satisfies:

$$\begin{array}{c} \begin{bmatrix}\text{MODE} & \boxed{4} \\ \text{INDEX} & \boxed{5}\end{bmatrix} \\ \begin{array}{ccc} \ldots & \begin{array}{c}\phi_h \\ \begin{bmatrix}\text{MODE} & \boxed{4} \\ \text{INDEX} & \boxed{5}\end{bmatrix}\end{array} & \ldots \end{array}\end{array}$$

where ϕ_h is the head daughter of Φ.

(75) Φ satisfies the GAP Principle with respect to a headed rule ρ if and only if Φ satisfies:



Appendix B
Related Grammatical Theories

As noted in Chapter 2, the theory of grammar developed in this text is most closely related to the framework known as 'Head-driven Phrase Structure Grammar', or HPSG. HPSG is one of a number of frameworks for grammatical analysis that have been developed within the Chomskyan paradigm, broadly conceived. The intellectual tradition our theory represents is eclectic in its orientation, synthesizing ideas from several approaches to the study of language. To clarify these connections, we provide here a brief and incomplete survey of related theories of grammar. We hope not only to provide some information about the intellectual roots of our approach to syntax, but also to explicate its relationship to other contemporary theories.

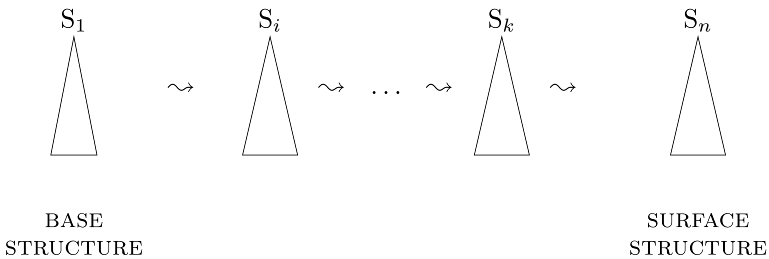
The basic concept of generative grammar is simply a system of rules that defines in a formally precise way (i.e. 'generates') a set of sequences (strings over some vocabulary of words or 'formatives') that represent the well-formed sentences of a given language.[1] Thus both of the systems considered in Chapter 2 – the regular expression (finite-state) and context-free phrase structure grammars – are generative grammars, as is the grammar summarized in Appendix A.

Generative syntax began in the 1950s when Noam Chomsky and others he influenced developed and formalized a theory of grammar based on the notion of 'transformation'.[2] The architecture of a transformational generative grammar defines sentence well-formedness indirectly: first, base (or 'underlying' or 'kernel') structures are gen-

[1]We follow here Chomsky's (1975,1966) original usage of this term. The term 'generative grammar' is sometimes also used to refer to 'generative-transformational grammar' (see below). There is no small irony in this usage, given the characteristic practice of transformational work of the last two decades. This body of literature, though technical in appearance, has systematically eschewed the development of consistent, broad coverage systems whose predictions can be verified empirically. This stands in marked contrast to the formal and analytic precision that is emblematic of most of the nontransformational generative approaches we survey below, whose mathematical properties (including their generative capacity, both stringset and structural) has in many cases been explored in detail, and whose empirical coverage has frequently been tested in terms of large-scale, computational grammars.

[2]Chomsky's notion of transformation grew out of the theory of linguistic transformations invented in the 1950s by Zellig Harris. A number of relevant papers are collected in Harris 1970. On Harris' often overlooked contribution to theoretical linguistics and related areas, see Nevin 2003 and Nevin and Johnson 2003.

erated via a system of phrase structure rules;[3] and then transformational rules apply successively to map these phrase structures into other phrase structures. The sentences of the language, then, are just those that can be derived by applying transformational rules to the base structures according to a particular regime, e.g. a regime of ordered transformations, at least some of which are obligatory. A transformational derivation thus involves a sequence of phrase structures, the first of which is a base structure and the last of which is a phrase structure (usually called a 'surface structure') whose word string corresponds to a sentence of the language:



Transformational generative grammar (which has dominated the mainstream of syntactic theory from the 1960s through to the present) has changed significantly over the years. Yet, despite considerable evolution within this framework, the notion of transformational derivation has been present in one guise or another in virtually every formulation of transformational grammar.[4] Similarly, other commonalities remain in the practice of transformational grammarians, such as the treatment of sublexical entities (e.g. inflectional affixes) as independent syntactic elements, that is, as syntactic primitives on a par with words.[5]

In contrast to the transformational tradition, there is another approach to generative grammar, equally committed (if not more so) to the original goal of developing precisely formulated grammars. This tradition has two distinctive properties:

(i) **Constraint-Based Architecture**: Grammars are based on the notion of constraint satisfaction, rather than transformational derivation.

(ii) **Strong Lexicalism**: Words, formed in accordance with an independent lexical theory (or 'module'), are the atoms of the syntax. Their internal structure is invisible to syntactic constraints.

These two design properties together form the basis of the 'constraint-based lexicalist' (CBL) approach to generative grammar. In CBL approaches, surface structures are

[3]In some versions of this approach, lexical insertion into the structures generated by the CFG is accomplished by a separate specialized mechanism.

[4]There are exceptions, though, e.g. Koster 1987 and Brody 1995. Chomsky has always maintained that it is 'not easy' to provide empirical (or theory-independent) evidence that would lead one to prefer a transformational theory over simpler alternatives. Despite this repeated claim (Chomsky (1981: 90f; 1995: 223f)), Chomsky has included transformational operations in every version of grammatical theory he has developed since the 1950s. In more recent work, Chomsky (2002) claims that transformations are a 'conceptual necessity'; see Levine and Sag 2003 for some further discussion.

[5]Hence Chomsky's introduction of the term (syntactic) 'formative' to encompass stems, noninflecting words, and inflectional affixes.

generated directly, though ancillary kinds of syntactic representation may be cogenerated (see below).

The principle of strict lexicalism has its origin in the pioneering work of Chomsky (1970), who challenged previous attempts to derive nominalizations (e.g. *the enemy's destruction of the city*) from clauses (e.g. *the enemy destroyed the city*) via syntactic transformations. In the mid and late 1970s, many alternatives to transformational analyses were developed. There are two particularly significant developments in this period. The first is Bresnan's 'Realistic' Transformational Grammar (widely circulated in unpublished form; a version was published as Bresnan 1978), which for the first time provided a cogent treatment of numerous phenomena (e.g. passivization) in lexical rather than transformational terms. Bresnan's dramatic first step inspired a number of people, notably Brame (1979) and Gazdar (1981) [first drafted in 1980], to take the further step of purging transformations from syntactic theory altogether. Second, the emergence of the framework of Montague Grammar provided new techniques for characterizing meanings directly in terms of surface structure, thereby eventually eliminating any semantic motivation for syntactic transformations. In many versions of transformational grammar, active and passive sentences were derived from a common underlying structure, leading to the (controversial) suggestion that many aspects of meaning are preserved by transformational derivations. With the advent of more sophisticated methods of semantic analysis, distinct surface structures could be assigned formally distinct but equivalent semantic interpretations, thus accounting for the semantics in a principled fashion without appeal to transformations.

'Realistic' Transformational Grammar and Montague Grammar together set the stage for the emergence of fully nontransformational generative frameworks in the late 1970s and early 1980s. Most notable among these are Lexical Functional Grammar (LFG), Generalized Phrase Structure Grammar (GPSG), Categorial Grammar (CG), and Dependency Grammar, each of which we summarize below. The subsequent history of CBL generative grammar witnessed not only considerable development in each of these frameworks, but also the introduction of other new approaches, notably, Construction Grammar (CxG). Of immediate relevance also is the evolution of GSPG, through the integration of ideas from various other frameworks, into the framework of HPSG, from which many analyses and the general orientation of the present text are directly drawn.

Not all influential theories of grammar fit comfortably into our dichotomy between transformational and CBL approaches. In particular, three frameworks that need to be mentioned are Relational Grammar (RG), Tree Adjoining Grammar (TAG), and Optimality Theory (OT). These are all discussed briefly following our survey of CBL theories.

The primary focus of this appendix is a survey of CBL approaches. However, most CBL frameworks grew out of early transformational work, and more recent developments within transformational grammar have continued to exert an influence upon CBL research. For this reason, our survey begins with a very brief historical sketch of relevant aspects of transformational generative grammar.

B.1 Historical Sketch of Transformational Grammar, ca. 1955 to the present

An early version of transformational generative grammar was presented in Chomsky's 1957 book, *Syntactic Structures*, which was a condensed version of portions of *The Logical Structure of Linguistic Theory*, a magnum opus completed in 1955, but not published until twenty years later. The analyses presented there and in other transformational works of the period included explicit formal statements of rules intended to license all and only the well-formed sentences of the language under discussion (usually English). This emphasis on the precise formulation of hypotheses is perhaps the greatest influence of early transformational grammar on the approach presented here (along with such tools as context-free grammar, which were also invented by Chomsky during this period).

As noted above, a key claim of transformational grammar (in all its versions) is that an empirically adequate grammar requires that sentences be associated not with a single tree structure, but with a sequence of trees, each related to the next by a transformation. The initial trees in Chomsky's *Syntactic Structures* theory were to be generated by a CFG. For example, passive sentences (such as *The cat was chased by the dog*) were derived from the same underlying structures as their active counterparts (*The dog chased the cat*) by means of a passivization transformation that permuted the order of the two NPs and inserted the words *be* and *by* in the appropriate places.

The most celebrated analysis in this theory is its treatment of the English auxiliary system (roughly, the material covered in Chapter 13 of this text). Chomsky (1957) proposed that tense was, in the underlying syntactic structure, a 'formative' separate from the verb on which it ultimately appears. A movement transformation was posited to account for inversion in questions (deriving, e.g. *Is the sun shining?* from the same underlying structure as *The sun is shining*); and an insertion transformation placed *not* in the appropriate position for sentence negation. Both these transformations in some instances have the effect of stranding tense – that is, leaving it in a position not adjacent to any verb. For these cases, Chomsky posited a transformation to insert *do* as a carrier of tense. Several other uses of auxiliary *do* (e.g. in ellipsis) were also treated on this view as instances of tense stranding. This unified account of apparently disparate uses of *do*, together with the formal explicitness of the presentation, won many converts to transformational grammar.

Katz and Postal (1964) and Chomsky (1965) introduced a number of major changes into transformational grammar, and Chomsky dubbed the resulting theory 'the Standard Theory'. It differed from earlier transformational grammar in several ways, some rather technical. Among the important innovations of this theory were the use of recursive phrase structure rules (allowing for the elimination of transformations that combined multiple trees into one) and the introduction of syntactic features to account for subcategorization (valence).

Perhaps the most important conceptual change was the addition of a semantic component to the theory of transformational grammar. In this theory, the initial tree in each sentence's derivation, known as its 'deep structure', transparently represented all the information necessary for semantic interpretation. In particular, it was claimed that there is a simple mapping between the semantic roles played by arguments to a verb (intuitively, who did what to whom) and the deep structure grammatical relations (subject, object, etc.). In the final tree of the derivation (the surface structure), the words and phrases were arranged as the sentence would actually be pronounced. On this theory, then, transformations were thought to be the primary link between sound and meaning in natural language.

The Standard Theory had great intuitive appeal and attracted much attention from neighboring disciplines. In particular, many philosophers were attracted by the idea that deep structures might provide something very much like the 'logical form' of sentences needed for precise analysis of their role in inference. Likewise, psychologists hoped that the transformational derivations were a first approximation to the mental processes involved in the production and comprehension of utterances. Initial experiments gave credibility to this idea, in that they showed a correlation between the psychological complexity of a sentence and the number of transformations posited in its derivation. Further research on this idea (usually referred to as the 'derivational theory of complexity') failed to support it, however, and by the early 1970s it had been largely abandoned (Fodor et al. 1974).

Most contemporary grammatical theories have preserved the most important innovations of the Standard Theory, namely, syntactic features, recursive phrase structure, and some sort of semantic component. On the other hand, no current theory maintains the centrality of transformations in mediating between sound and meaning.

The first major challenge to Chomsky's views within the generative paradigm was a movement known as 'Generative Semantics'; its leading figures included George Lakoff, James McCawley, Paul Postal, and John ('Haj') Ross. They carried the central idea of the Standard Theory to its logical conclusion, claiming that deep structures should themselves be viewed as representations of meaning, and denying that syntactic and semantic rules should be considered distinct components of a grammar. That is, on the Generative Semantics view, something was considered a possible input to the transformational rules just in case it represented a proposition that made sense. Hence all languages could be derived from the same underlying source, differing only in how the underlying representations get transformed into sounds.

The underlying trees of Generative Semantics were far larger and more elaborate than those of the Standard Theory (though the inventory of grammatical categories was much reduced). Virtually all the work involved in describing the relationships between form and meaning in language was done in this theory by transformations, though these rules were rarely formulated explicitly.

Generative Semantics enjoyed wide currency for a few years and served as the vehicle for the exploration of a wide range of fascinating phenomena in many languages. Although the theory itself had a short life span (for reasons that have been debated by historians of linguistics[6]), many of the constructions first discovered by generative semanticists continue to figure prominently in theoretical discussions. Moreover, some recent analyses have borne striking resemblances to earlier Generative Semantics proposals, as has often been observed.

Unlike the generative semanticists, Chomsky and some others (notably, Ray Jack-

[6] See Newmeyer 1986, Harris 1993, and Huck and Goldsmith 1995.

endoff) quickly abandoned the idea that pairs of sentences with identical deep structures must be synonymous. In particular, they argued that transformations that reordered quantified NPs could change the scopes of the quantifiers (e.g. *Many people read few books* was claimed to have a range of interpretations different from *Few books are read by many people*). Hence they claimed that structures other than deep structures must play a role in semantic interpretation.

Instead of the complex underlying trees and elaborate transformational derivations of Generative Semantics, the framework that Chomsky dubbed the 'Extended Standard Theory' (EST) posited a relatively impoverished theory of transformations; instead, it enriched other components of the theory to carry much of the descriptive burden. In addition to the new types of semantic rules alluded to above, schematization over phrase structure rules and an enriched conception of the lexicon – including lexical rules – were introduced. These innovations have been carried over into much contemporary work, including the theory developed in this text. The approach of EST led to a highly 'modular' theory of grammar, with a variety of distinct types of mechanisms to account for different kinds of empirical phenomena.

EST also saw the introduction of 'empty categories' – that is, elements that occupy positions in a tree but which have no phonetic realization. These included a type of null pronoun used in control constructions (e.g. the subject of *leave* in *We tried to leave*) and 'traces' of elements that have been moved.

A central concern of EST and much subsequent work has been to constrain the power of the theory – that is, to restrict the class of grammars that the theory makes available. The primary rationale for seeking such constraints has been to account for the possibility of language acquisition, which (as noted in Chapters 1 and 9) Chomsky regards as the central question for linguistics.

EST was superseded by what came to be known as Government and Binding Theory, or GB. GB was first laid out in Chomsky's (1981) book, *Lectures on Government and Binding*.[7] It develops the modular style of EST, dividing the theory of grammar into a set of subtheories, each with its own set of principles, assumed to be universal. Although GB still used transformational derivations to analyze sentences, it reduced the transformational component to a single rule (referred to as 'Move α'), which could move anything anywhere. The idea was that general principles would filter out most derivations, preventing the massive overgeneration that would otherwise result from such an underconstrained transformational operation.

Elaborating on earlier work in EST, GB analyses posited a taxonomy of empty categories. Binding Theory, which was a major topic of research within GB, was applied not only to overt pronouns, but also to empty categories. Movement was formulated so as to leave behind traces (a kind of empty category), which were bound by the moved element. Binding Theory thus attempted to find relations between constraints on movement and constraints on possible pronoun-antecedent relations. Since movement was used to deal with a wide range of phenomena (including filler-gap dependencies, the active-passive relation, raising, extraposition, and auxiliary inversion), linking all of these to the binding principles yielded a highly interconnected system.

[7]For an introductory presentation, see Haegeman 1994.

The primary focus in GB and subsequent transformational research has been the postulation of a theory of universal grammar. GB claimed that many of the principles that make up the theory of grammar are parameterized, in the sense that they vary within a narrow range.[8] Learning a language, on this view, consists of fixing a small set of parameters (plus learning vocabulary). That is, GB claimed that all languages are essentially alike, with only a few restricted parameters of possible variation.

Many linguists since the early 1980s have framed their grammatical studies in terms of this framework, yielding a large literature that represents analyses of a much wider range of languages and phenomena than any of the other theories listed here. But the analyses developed within GB are often inconsistent with one another. In addition, these analyses are seldom formulated with a precision comparable to that assumed in this text. For these reasons (and the further absence of any theory of what could count as a possible 'parameter'[9]), particular GB analyses and the general claims about cross-linguistic parametric variation are often quite difficult to evaluate. Nonetheless, it is clear that GB analyses tend to share certain noteworthy characteristics, including the following:

- Highly articulated phrase structures (linguistically significant distinctions and relations are encoded into tree configurations);
- Use of movement (that is, the transformation 'Move α');
- Extensive use of empty categories;
- A rich set of universal principles, some of which are parameterized;
- Avoidance of language-particular rules (properties specific to a language are to be expressed in terms of values of universally available parameters);
- Deductive structure (small changes in a grammar should have far-reaching consequences for the language, so that stipulation is minimized).

The theory we have presented here has been influenced by GB in a number of ways. These include very general goals, such as striving for a theory whose components interact deductively. They also include more specific design features, such as the general form of the Binding Theory (though not the detailed statement of the binding principles). Finally, there are specific points of our grammar that were first proposed within GB analyses, such as treating complementizers as heads that could take sentences as their complements.

As of this writing, the most recent incarnation of transformational grammar is the 'Minimalist Program'. As its name implies, MP is a program for research, rather than a theory of syntax. Further, it is a program that seeks to discover whether natural language grammar (and in particular syntax) is an 'optimal' system in the sense of requiring the fewest theoretical constructs. The general tactic is to lay out what appears to be an optimally simple theory of syntax and then test whether it can account for the data of natural languages.

The model that the MP is currently exploring is one in which the only information associated with formatives is information relevant to either the pronunciation (that is, the level of Phonological Form) or the meaning (that is, the level of Logical Form). A

[8] Another name for this approach to syntax is 'Principles and Parameters'.

[9] For a recent discussion of this issue, see Baker 2001 and especially Trask 2002.

small set of syntactic operations combine the formatives into syntactic structures and transform those structures along two paths, creating a Phonological Form and a Logical Form, which are meant to be interpreted by other cognitive systems. Constraints posited in the theory may refer to these end levels, but not any intermediate stages. In addition, competing derivations based on the same set of formatives can be compared and evaluated with respect to some economy metrics. The derivations that are preferred by the economy metrics yield grammatical strings, those that are dispreferred yield ungrammatical strings. More recent work has sought to localize such effects, applying economy constraints at each step of a transformational derivation. This conception of grammar, in which the properties of competing transformational derivations are crucial in determining sentence well-formedness,[10] represents a radical departure from the original goals and methodology of generative grammar and has no direct connection with the theoretical orientation of the present work.

The seminal work on MP is Chomsky 1995. For an elementary exposition of MP, see Radford 1997; also useful are Webelhuth 1995 and Epstein and Seely 2002. For an extensive comparison of MP with CBL approaches in general, see Johnson and Lappin 1999. For a critical perspective on recent Minimalist work, see Bender 2002. For a sociologically intriguing discussion of the evolution of MP from GB, see Lappin et al. 2000a, 2000b, 2001.

B.2 Constraint-Based Lexicalist Grammar

Some CBL theories of grammar were developed by transformational grammarians who began to question the reasons that had been given for positing transformations. Others have a rather different genealogy, growing out of work on the artificial languages of logic and computer science. Combining the precision and rigor of the latter tradition with the attention to empirical generalizations that dominates the former has been extremely fruitful, though by no means easy.

The following subsections provide very brief sketches of some of the most important CBL theories of grammar.[11]

B.2.1 Categorial Grammar (1974 to the present)

Categorial Grammar (CG) has a long history dating back to the 1930s, but it was developed primarily by mathematical logicians before the early 1970s. CG first received widespread attention from linguists when the logician Richard Montague used it as the syntactic framework to go with his new approach to analyzing natural language semantics.[12] Contemporary work on CG maintains Montague's strong commitment to semantic

[10] Generative semanticists in the early 1970s briefly discussed the need for what were then termed 'transderivational constraints', but the idea was not pursued for long.

[11] The dates given in parentheses are roughly the periods during which a substantial number of researchers have been or are still active in developing the theory.

[12] Montague's intensional logic and his precise framework for studying the relation between expressions and their meaning had a considerable influence on work in linguistics. His famous remark (Montague 1970:373) that 'there is in my opinion no important theoretical difference between natural languages and the artificial languages of logicians' is sometimes referred to (following Bach 1989) as 'Montague's Hypothesis'.

compositionality – that is, to the idea of building up the structures and interpretations of sentences at the same time, using coordinated syntactic and semantic rules (the 'Rule-to-Rule Hypothesis').

The central idea of CG is that the combinatory properties of grammatical categories can be directly encoded in their forms. A CG has a small number of basic categories, and a large (potentially infinite) class of categories defined in terms of how they combine with other categories. For example, we might designate S and NP as basic and define a category of expressions that combine with an NP on their left to form an S, which can be annotated NP\S. This category would be equivalent to our VP (including those consisting simply of an intransitive verb). Another category, annotated (NP\S)/NP, would be used for expressions that combine with an NP on their right to form an NP\S; this corresponds roughly to our category of strictly transitive verbs, but might also be used for more complex expressions like *let stand*, in sentences like *The decision let stand the lower court's ruling*. Because categories are defined in terms of their members' potential for combining with other constituents, CG is often seen as a variety of Dependency Grammar (q.v.).

The simplest forms of CG only combine categories by simple concatenation, with concomitant elimination of a 'slash' in the combined category. For example, two expressions of categories NP and NP\S can be concatenated to form a new expression of category S. Such a CG is provably equivalent to a simple context-free phrase-structure grammar. Richer versions of CG include other rules of category combination, for example allowing two expressions of categories A/B and B/C to combine to create an expression of category A/C (so-called 'composition'). They may also allow what is called 'type raising', for example, reanalyzing an expression of category A as belonging to a more complex category like B/(B/A) or B/(A\B). Such enrichments of the basic machinery of CG have made possible appealing analyses of a range of syntactic, semantic, and phonological phenomena in natural languages.

CG divides into two major schools of thought, known as 'type logical grammar' and 'combinatory categorial grammar'. The former treats the combination of categories as a system of logic, and the derivations of sentences are treated as a kind of mathematical proof. The latter focuses more on linguistic questions, positing operations that are motivated by natural language phenomena, even if they seem unnatural as logical operations.

The primary attractions of CG have always been its conceptual simplicity and the fact that it is well suited to the formulation of tightly linked syntactic and semantic analyses. It also provides particularly elegant and appealing accounts of coordination. In a number of works (notably Steedman 1996), Steedman has argued that CG is well-suited to explaining a range of facts about English prosody (that is, pitch accent and intonation) and about how people process sentences.

One characteristic of all but the simplest versions of CG has been variously cited as a strength and a weakness: CG typically allows many more different analyses of any given string than other theories of grammar. For example, in a theory with composition and type-raising, a simple transitive sentence like *Pat likes oatmeal* can be derived either by combining the verb with its object first, and then with its subject, or by combining the verb with its subject first, and then with its object. With longer sentences, the number of possible derivations grows extremely rapidly.

More information about CG is available online at: http://www.cs.man.ac.uk/ai/CG. For an introduction to categorial grammars, see Wood 1993 and Carpenter 1997. A brief overview is available online at http://cognet.mit.edu/MITECS/Entry/steedman. Steedman (1996, 2000) summarizes the combinatory categorial grammar tradition. For the competing 'type-logical' approach, see Morrill 1994 and Moortgat 1997.

The influence of CG on the theory developed in this text is quite clear. The valence features of HPSG do much the same work as complex categories do in CG. The nodes of trees in HPSG are labeled with feature structures that contain all the information in CG categories (plus quite a bit more). Our grammar rules cancel elements off our valence lists in a way analogous to the combinatory rules of CG. Hence many CG analyses can be translated fairly straightforwardly into HPSG.

B.2.2 Construction Grammar (1988 to the present)

Construction Grammar (CxG) is a label applied to a family of theories that take grammatical constructions as essential units out of which sentences are built. Constructions, on this view, encompass not only (descriptions of) words and phrases, but also idioms and other types of collocations, characterizable at various levels of generality. An important leading idea of work in CxG is that there is no important difference between 'core' and 'peripheral' constructions.[13] Construction grammarians argue that, by contrast, grammatical theory can and should include numerous principles whose domain of application includes both kinds of construction, no matter how that distinction is made. A common misconception about CxG is that it is just a theory of 'marginal', 'peripheral' constructions. To the contrary, one of its fundamental claims is that linguistic rules (constraints) operate at diverse levels of generality, without there being any theoretical difference (other than the scope of their application) between construction-specific constraints, constraints of intermediate grain (e.g. constraints on all headed constructions) and the most general principles of UG.

CxG researchers countenance neither empty categories nor transformations, two formal devices central to transformational work, finding them empirically unmotivated and descriptively unnecessary. As in HPSG (q.v.), CxG is based on the notion of constraint satisfaction and constraint inheritance is used to express generalizations that cut across diverse kinds of objects that pattern into clusters obeying a 'family resemblance'. Chapter 16 of this book outlines a formal version of CxG. As is clear from the discussion there, the central features of the theory developed in our first fifteen chapters translate naturally into a constructional framework. The framework outlined there is closest to that of Fillmore et al. (forthcoming). There are many other approaches that fall under the CxG umbrella, however, some of which eschew formal approaches to grammar.

There are two principal substantive motivations for CxG. The first is to account for grammatically determined, non-truth-conditional aspects of meaning – including such 'pragmatic' factors as conventional implicature and presupposition – in the formal rep-

[13] This is a distinction introduced (Chomsky 1986b) to distinguish constructions subject to UG from language-particular idiosyncrasies. Unfortunately, no criteria have (to our knowledge) ever been offered to distinguish between these two sorts of constructions, rendering the empirical content of the 'core-periphery' distinction, once again, quite difficult to evaluate.

resentations of sentences and of the grammatical constructions that license them. The second is to account for the full range of idiomatic and semi-idiomatic constructions, phenomena that are pervasive in the world's languages. Although every generative approach to the grammar of natural languages is committed in principle to full coverage of the facts of all languages, as well as in the extraction of intralanguage and interlanguage generalizations (the latter usually considered to constitute the stuff of universal grammar), varying approaches differ in their relative emphasis on the full coverage of language facts versus the development of a parsimonious theory of universal grammar. CxG falls at the end of this scale, emphasizing its concern for empirical coverage and the need base accounts of UG on well worked-out empirical descriptions.

Kay 1995 is a brief, acessible overview of the basics of CxG. For online information, see the official website at: http://www.constructiongrammar.org/; there is further information at: http://www.icsi.berkeley.edu/˜kay/bcg/ConGram.html. A version of CxG showing the influence of approaches to grammar that refer to themselves as 'cognitive' (e.g. Lakoff 1987, Langacker 1987) is developed by Goldberg (1995). The original, generative, view of CxG (Fillmore et al. 1988) is further developed in recent work by Michaelis and Lambrecht (1996), Kay and Fillmore (1999), Kay (2002), and Fillmore et al. (forthcoming).

B.2.3 Dependency Grammar (1959 to the present)

Work on transformational grammar rests on two crucial (but controversial) assumptions about sentence structure: that it is organized hierarchically into 'phrases' (hence 'phrase structure'), and that grammatical relations such as 'subject' and 'object' are derivative, to be defined in terms of phrase structure configurations. The assumption of phrase structure is a distinctively American contribution to linguistics, having been suggested by Bloomfield (1933). Bloomfield suggested that sentences should be analyzed by a process of segmentation and classification: segment the sentence into its main parts, classify these parts, then repeat the process for each part, and so on until the parts are 'morphemes', the indivisible atoms of grammar. Thus *Cool students write short essays* divides into the noun phrase *cool students* plus the verb phrase *write short essays*, which in turn divides into the verb *write* plus *short essays*, and so on. This contrasts with the European tradition (which dates back to classical Greece) in which the focus is on individual words and their relationships - for example, *cool* is an 'attributive modifier' of *students*, and *students* is the subject of *write*.

The attraction of phrase structure analysis is its formal clarity, which is revealed by the familiar phrase structure trees. Various linguists (mainly European) have attempted to develop the traditional approach in a similarly formal manner, with the emphasis on the relationships among words rather than on the groupings of words. One of the characteristics of these relationships is that the words concerned are generally not equal, in that one serves to modify the meaning of the other; so *cool students* denote certain students, and *students writing essays* denotes a kind of writing. The relationships are called 'dependencies', with the modifying word depending on the modified (so *cool* depends on *students*, and *students* on *write*), and the approach is called 'Dependency Grammar' (DG) to contrast it with phrase structure grammar.

There are several ways to represent DG analyses diagrammatically, including a system that has been widely used in American schools since the nineteenth century which is often called simply ‘sentence diagramming’. The first real attempt to build a theory of DG analysis was Tesnière 1959, but since then developments in the tradition of PSG[14] have been paralleled in DG theories. One of these which is particularly close in other respects to HPSG is ‘Word Grammar’ (Hudson 1984, 1990, 1998). Online information about DG is available at: http://ufal.mff.cuni.cz/dg/dg.html.

In some respects, HPSG bridges the gap between DG and PSG, for in HPSG all the parts of a phrase depend directly on its head word – phrases are ‘head-driven’, just as in DG. On the other hand, in HPSG the dependent parts are themselves phrases with their own internal structure consisting of a head word and its dependents.

B.2.4 Generalized Phrase Structure Grammar (1979–1987)

Generalized Phrase Structure Grammar, (known as GPSG) was initiated by Gerald Gazdar in a pair of papers (Gazdar 1981, 1982) that attracted the attention of numerous researchers in the field of syntax. The theory was further developed by him and a number of colleagues in the early 1980s and was codified in the 1985 book, *Generalized Phrase Structure Grammar* (Gazdar et al. 1985), which provides a detailed exposition of the theory.

The central idea of GPSG is that standard context-free phrase structure grammars can be enhanced in ways that do not enrich their generative capacity, but which do make them suitable for the description of natural language syntax. The implicit claim of work in GPSG was that the tenable arguments against CFG as a theory of syntax were arguments about efficiency or elegance of notation, and not about coverage in principle.

Among the important ideas that originated in GPSG are the separation of CFG rules into (i) rules of immediate dominance (‘ID rules’), which specify only which phrases can appear as daughters in a local syntactic tree, and (ii) rules of linear precedence (‘LP rules’), which specify general constraints determining the order of daughters in any local tree. This factorization of the two functions of traditional CFG rules is preserved in HPSG, though we have not employed it in the formulation of grammar rules in this text.

A second idea stemming from work in GPSG is the treatment of long-distance dependency constructions, including filler-gap constructions (such as topicalization, *wh*-questions, and relative clauses). The GPSG treatment of these phenomena involved locally encoding the absence of a given constituent via a feature specification. The remarkable result of the transformationless GPSG analysis of filler-gap dependencies was that it succeeded where transformational theories had failed, namely in deriving the Coordinate Structure Constraint and its ‘across-the-board’ exceptions (see Chapter 14). This feature-based analysis of filler-gap dependencies is preserved in HPSG, and we have carried it over virtually intact to the current text.

[14] We use the term ‘PSG’ to refer to research developing extensions and generalizations of Context-Free Phrase Structure Grammars, e.g. GPSG (q.v.) and HPSG (q.v.).

B.2.5 Head-Driven Phrase Structure Grammar (1984 to the present)

HPSG evolved directly from attempts to modify GPSG in the interdisciplinary environment of Stanford's Center for the Study of Language and Information (CSLI), the site of several experimental computational systems for language processing. From its inception, HPSG has been developed as a conscious effort to synthesize ideas from a variety of perspectives, including those of Situation Semantics (which originated at CSLI at about the same time as HPSG), data type theory, and a variety of other linguistic frameworks under development in the early and mid-1980s. The name 'Head-driven Phrase Structure Grammar' was chosen to reflect the increasingly recognized importance (as compared with, say, GPSG) of information encoded in the lexical heads of syntactic phrases. Dependency relations are lexically encoded, as they are in Dependency Grammar (q.v.), Categorial Grammar (q.v.) and LFG (q.v.). The theoretical aspects of HPSG have been developed in considerable detail in three books (Pollard and Sag 1987, Pollard and Sag 1994, and Ginzburg and Sag 2000) and a number of major articles.

Some of the key ideas of work in HPSG are: (1) a sign-based architecture (see Chapter 16); (2) the organization of linguistic information via types, type hierarchies, and constraint inheritance; (3) the projection of phrases via general principles from rich lexical information; (4) the organization of such lexical information via a system of lexical types; and (5) locality of selection, agreement, case assignment and semantic role assignment, as guaranteed by the organization of feature structures, e.g. into *synsem* objects that appear on ARG-ST (or SUBCAT) lists, and (6) the factorization of phrasal properties into construction-specific and more general constraints. These properties have all been discussed at various places in this text.

Since the inception of HPSG, researchers have been involved with its computational implementations. From 1980 until 1991, Hewlett-Packard Laboratories in Palo Alto, California supported one such project, which involved the first two authors of this text and a number of colleagues and students. It was with this project that many of us learned for the first time how far the rhetoric of theoretical linguistics can be from the reality of working grammars. At the time of this writing, implementations of HPSG and HPSG-like grammars are being developed at numerous universities and industrial research laboratories around the world, including sites in North America, Western and Eastern Europe, Japan, Korea, Taiwan, and Australia. The LinGO initiative, an ongoing collaboration that includes partners in the US, Europe, and Asia, makes available (without cost) a number of open-source, HPSG-related, computational resources. These resources, which include grammars, lexicons, and the LKB Grammar Engineering Platform,[15] are available online at: http://lingo.stanford.edu. Another HPSG-related implementation effort, one that includes collaborators from Europe, the US, and Canada, is the TRALE system (and its predecessor – the ConTroll-System, developed at the linguistics department at the University of Tübingen), which is described at http://www.sfs.uni-tuebingen.de/hpsg/sysen.html. The TRALE system is an extension of the Attribute-Logic Engine (ALE) system, which can be freely downloaded from http://www.cs.toronto.edu/˜gpenn/ale.html. For general information about HPSG, see http://hpsg.stanford.edu and http://www.ling.ohio-state.edu/research/hpsg/.

[15]For an accessible introduction to the LKB system and related issues, see Copestake 2002.

B.2.6 Lexical Functional Grammar (1979 to the present)

The theory of Lexical Functional Grammar, commonly referred to as 'LFG', shares with Relational Grammar (q.v.) the idea that relational concepts like 'subject' are of central importance and cannot be defined in terms of tree structures. But it also treats phrase structure as an essential part of grammatical description and has focussed on the development of a universal theory of how constituent structures are associated with grammatical relations.

In LFG, each phrase is associated with multiple structures of distinct types, with each structure expressing a different sort of information about the phrase. The two representations that have been the center of attention in most LFG literature are: the 'functional structure', which expresses the relational information that is analogous in certain respects to our ARG-ST and in other respects to the valence features SPR and COMPS; and the 'constituent structure', which is a tree diagram very similar to the surface structures of the Standard Theory. General principles and construction-specific constraints define the possible pairings of functional and constituent structures. In addition, LFG recognizes a number of further levels of representation. Perhaps most notable among these are σ-structure, which represents linguistically significant aspects of meaning, and a-structure, which serves to link syntactic arguments with aspects of their meanings. Thus the analogue of the HPSG sign presented in Chapter 16 is a tuple of LFG structures, possibly the four-tuple consisting of a sentence's c-structure, f-structure, a-structure, and σ-structure.

There are no transformations in LFG. Much of the descriptive work done in earlier theories by transformations is handled by an enriched lexicon, an idea pioneered by LFG researchers. For example, the active-passive relation in English is treated as a lexical relation between two forms of verbs. In early LFG, this was codified in terms of lexical rules similar to those presented in this text. Subsequent work has sought to develop a more abstract conception of lexical relations in terms of 'lexical mapping theory'. LMT provides for constraints on the relation between f-structures and a-structures, that is, constraints associated with particular arguments that partially determine their grammatical function. It also contains mechanisms whereby arguments can be suppressed in the course of lexical derivation. In LFG, information from lexical entries and phrasal annotations is unified to produce the functional structures of complex expressions.

As the above description makes clear, LFG and HPSG bear many resemblances. In particular, HPSG has been able to incorporate many insights from work in LFG, most notably: the significant use of dependency-based analyses of numerous phenomena (rather than accounts based on constituent structure) and the general constraint-based approach to grammatical description. There are crucial differences between LFG and HPSG as well, e.g. the use of types and type-based constraint inheritance, which plays no role in the LFG literature. Similarly, the idea of cancelling arguments from valence lists (which stems from Categorial Grammar (q.v.)) is absent from LFG; instead, dependencies are managed in terms of the notions of 'completeness' and 'coherence', which ensure that functional structures contain sufficient and consistent information. LFG's constraining equations provide a way of handling certain kinds of grammatical requirements that has no analog in HPSG. Conversely, the defeasible constraints that are found in some versions of HPSG have no direct counterpart in LFG.

The differences in practice between the HPSG and LFG communities can lead to rather different analyses of the same phenomena; yet these analyses are often compatible with either framework. For an overview of current developments in LFG, see Dalrymple et al. 1995, Bresnan 2001, Dalrymple 2001, and the LFG website at http://www-lfg.stanford.edu/lfg/. The Parallel Grammar Project, based at the Palo Alto Research Center (PARC), is an ongoing collaboration that has developed a variety of computational resources for NLP based on LFG. Information about the ParGram Project, which includes researchers at a number of sites in the US, Europe and Japan, can be found online at http://www2.parc.com/istl/groups/nltt/pargram/.

B.3 Three Other Grammatical Frameworks

B.3.1 Relational Grammar (1974 to the present)

We can now return to the second controversial claim that transformational grammar made about sentence structure, namely, that grammatical relations are derivative. In early theories of generative grammar, transformations were defined in terms of structural properties of tree diagrams. To the extent that traditional notions like 'subject' and 'direct object' were employed in these theories, they were regarded simply as shorthand for relations between linguistic elements definable in terms of the geometry of trees. Relational Grammar (RG), developed by Paul Postal, David Perlmutter, David Johnson and others, adopts primitives that are conceptually very close to the traditional relational notions of subject, direct object, and indirect object. In this respect there is a strong affinity between RG and Dependency Grammar (q.v.). The grammatical rules of RG are formulated in relational terms, replacing the earlier formulations based on tree configurations. For example, the passive rule is stated in terms of promoting the direct object to subject, rather than as a structural rearrangement of NPs.

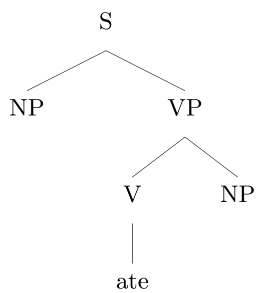
This approach allows rules to be given very general formulations that apply across languages. The characterization of passivization as promotion of the object does not depend on whether subjecthood and objecthood are indicated by word order or by other means, such as case marking on the nouns or some marking on the verb.

Although the influence of RG on the theory presented here may not be obvious, it is real. The notions of 'specifier' and 'complement' employed in this text are generalizations of 'subject' and 'object'. Languages use different grammatical devices to mark these relations (word order, case marking, agreement, etc.), and so a theory whose primitives are too closely linked to these devices would be unable to express cross-linguistic similarities. A number of contemporary theories, including LFG (q.v.) and HPSG (q.v.), have adopted this central insight of RG.

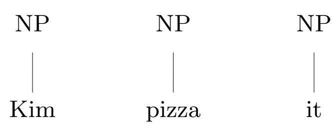
The RG framework has been applied to the description of a much wider variety of languages than were earlier generative theories (which tended to concentrate on the familiar European languages, East Asian languages, and a few others). For a brief online overview of the leading ideas of RG, see: http://cognet.mit.edu/MITECS/Entry/aissen. Various results of work in this framework are anthologized in Perlmutter 1983 and Postal and Joseph 1990. 'Arc Pair Grammar' (Johnson and Postal 1980) is an axiomatization and elaboration of many of the central ideas of Relational Grammar. Arc Pair Grammar was the first formalized constraint-based theory of grammar to be developed in the generative tradition. Although few details of the theory have been carried over into contemporary CBL theories, it pioneered many important innovations.[16]

B.3.2 Tree-Adjoining Grammar (1975 to the present)

Where other theories of grammar have posited representations of grammatical dependencies instead of or in addition to tree structure, Tree-Adjoining Grammar (TAG) (developed by Aravind Joshi, Leon S. Levy, Masako Takahashi, K. Vijay-Shanker, David Weir, and others, beginning in the early 1970s) has pursued the possibility of making trees into better representations of grammatical dependencies. The building blocks of a TAG are not rules and lexical entries, but elementary trees, anchored in lexical entries. In order to use trees to represent dependencies, TAG introduces an 'extended domain of locality'. In CFGs, the domain of locality is just one level in the tree, as each grammar rule can only describe a subtree of depth one. Trees of depth one can only represent some grammatical dependencies – e.g. between a verb and its object, but not between a verb and its subject. The elementary trees of a TAG are larger, such as this tree representing the verb *ate* and its dependents:[17]



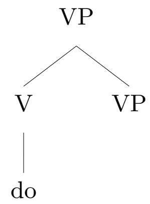
By the operation of 'substitution', the NP nodes in the elementary tree for *ate* could be replaced with any of the following trees:



With this extended domain of locality, TAG elementary trees can represent valence dependencies, but only if all arguments are realized locally. In order to capture non-local realization of dependents (e.g. raising, long-distance dependencies), TAG introduces another operation for combining elementary trees, called 'adjunction'. In adjunction, one elementary tree is spliced into another. Adjunction is defined so as to be possible only if the tree to be spliced in has a node on its frontier that is of the same category as its mother. An example of such a tree is the elementary tree for the raising verb *do*:

[16]For example, the graph-theoretic foundations that Johnson and Postal developed for Arc Pair Grammar are essentially the same as those employed in much work in HPSG (although the definition of feature structures as functions adopted in this book is a slightly different conception).

[17]Notice that the information in this tree is almost identical to that found in the ARG-ST (or SUBCAT) lists in HPSG analyses. The extension of the domain of locality to include subjects has been part of lexicalist theories since the invention of Categorial Grammar (q.v.) in the 1930s.



The operation of adjunction allows TAG to ‘factor recursion from the domain of dependencies’, while still using trees to represent both.

This presentation of the core ideas of TAG has been informal, but the formal properties of TAG and a number of its variants have been extremely well-studied. Indeed, research on TAG, its variants, and its relationship to other formalisms constitutes an important area of work in formal language theory. Further, there are broad-coverage implemented grammars developed in TAG, including the XTAG grammar (The XTAG Research Group 2000) and some implemented HPSG grammars have been compiled into TAGs (Kasper et al. 1995). Online information about the XTAG Project is available at http://www.cis.upenn.edu/~xtag/home.html. There is also work on modeling psycholinguistic results (including processing complexity) using TAG, including Joshi 1990 and Joshi et al. 2000. For an overview of TAG, see Joshi 2003. Other important work in the literature on TAG includes Joshi et al. 1975, Vijay-Shanker 1987, Weir 1987, and Abeillé and Rambow 2000.

B.3.3 Optimality Theory (1993 to the present)

Optimality Theory, or OT, was first developed as a phonological framework (Prince and Smolensky 1993), and has recently been adapted to syntactic analysis (see, for example, Barbosa et al. 1998, Bresnan 2000, Legendre et al. 2001, and Sells 2001). For an overview, see Kager 1999, Archangeli and Langendoen 1997, and the papers collected at the Rutgers Optimality Archive (http://roa.rutgers.edu/).

OT posits a universal set of defeasible constraints. The grammar of a language consists of a ranking of the constraints. Determining whether a given string of words is a well-formed sentence involves comparing it with other candidate expressions of the same proposition. The candidate whose highest-ranking constraint violation is lower than that of any other candidate is grammatical. For example, if constraint A outranks constraint B, which outranks constraint C, and if candidate sentence 1 violates A whereas candidate sentence 2 violates B and C, then sentence 2 is preferred over sentence 1, and sentence 1 is ungrammatical. If no other candidate sentence wins such a competition against sentence 2, then sentence 2 is licensed by the grammar.

The idea of constraints that can be violated is also incorporated in the theory presented in this book, since defeasible constraints specified in type hierarchies can be overridden. Moreover, a hierarchy of types with defeasible constraints defines a partial ordering on those constraints, with those introduced lower in the hierarchy taking precedence over those introduced at higher levels. Although there are substantive differences, certain central properties of OT can also be found in inheritance hierarchies with defeasible constraints.[18]

[18] Such hierarchies are explored in some detail in the artificial intelligence literature of the 1970s.

OT follows much earlier work in generative grammar in positing rich systems of universal grammar. However, the idea that determinations of well-formedness necessarily involve comparing structures or derivations is a break with past views, as we already noted in discussing the Minimalist Program (q.v.). Another common characteristic of MP and OT is the use of defeasible constraints. As noted above, such constraint mechanisms of various sorts have been proposed from time to time within some theories, including the theory presented in this book. This is not surprising, since idiosyncratic exceptions to general patterns are commonplace in natural languages. Defeasible constraint mechanisms are now employed fairly widely in various theories of syntax (though they are still controversial in certain circles). It remains to be seen whether a similar consensus will arise concerning the idea of defining well-formedness in terms of the outcome of some sort of competition.

B.4 Summary

In this appendix, we have surveyed a number of approaches to grammatical theory that have some relation to the analyses presented in this book. Our survey has been all too brief and has doubtless left out some approaches that certain readers will feel should have been included. Moreover, one thing we can say with certainty about the field of linguistics, at least over the last half century, is that theories of grammar have come and gone quite quickly. And this is likely to continue until the field evolves to a point where the convergent results of diverse kinds of psycholinguistic experiments and computational modeling converge with, and are generally taken to have direct bearing on, the construction of analytic hypotheses. Until that day, any survey of this sort is bound to be both incomplete and rapidly obsolescent.

For comparisons of OT with models that employ inheritance hierarchies with defeasible constraints, see Asudeh in press and Malouf in press.

Answers to Exercises

Chapter 1

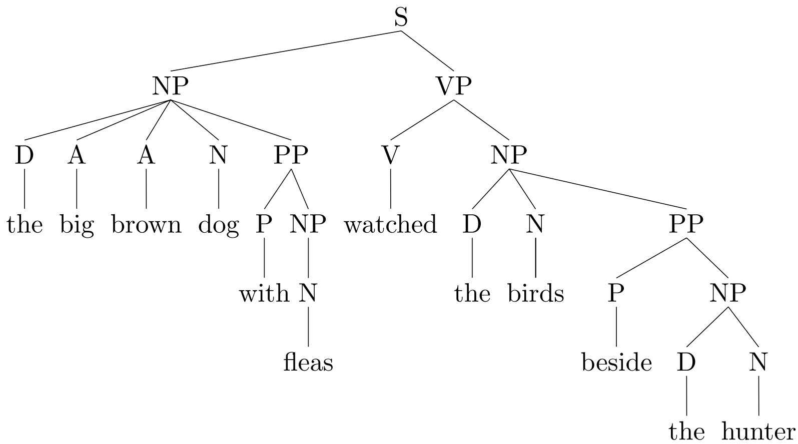
Exercise 1, page 4

A. In (i), *she* and *her* cannot refer to the same individual because, according to Hypothesis I, the nonreflexive pronoun *her* cannot be preceded by a coreferential expression. The occurrences of *she* and *herself* in (ii) are licensed by Hypothesis I, since the nonreflexive pronoun is not preceded by a coreferential expression, and the reflexive pronoun (namely, *herself*) IS preceded by a coreferential expression (namely *she*). The occurrence of *herself* in (iii) and the first occurrence of *herself* in (iv) both violate Hypothesis I, since neither one is preceded by a coreferential expression.

B. In (i)–(iv), all of the expressions that play a role in Hypothesis I are within the subordinate clauses. That is, the distribution of *herself*, *her*, and *she* in these sentences is exactly as it would be if the initial portion *We think that* were omitted. In (3), on the other hand, the coreferential expressions that Hypothesis I refers to are *we*, *us* (in (3a)), and *ourselves* (in (3b)); *we* is in the main clause, and *us* and *ourselves* are in the subordinate clause. This suggests that Hypothesis I will make the wrong predictions when the preceding coreferential expression is not in the same clause as the pronoun in question.

Chapter 2

Exercise 1, page 29



Exercise 2, page 40

A. (i) Terry wallows/*wallow in self-pity every morning.
 (ii) Pigs wallow/*wallows in mud.
 (iii) Kerry always remarks/*remark (that) it's getting late.
 (iv) They always remark/*remarks (that) it's getting late.

B. Categories:

PPV-SG	*wallows*
PPV-PL	*wallow*
SV-SG	*remarks*
SV-PL	*remark*

Rules:

VP-SG → PPV-SG PP
VP-PL → PPV-PL PP
VP-SG → SV-SG S
VP-PL → SV-PL S

Chapter 3

Exercise 1, page 58

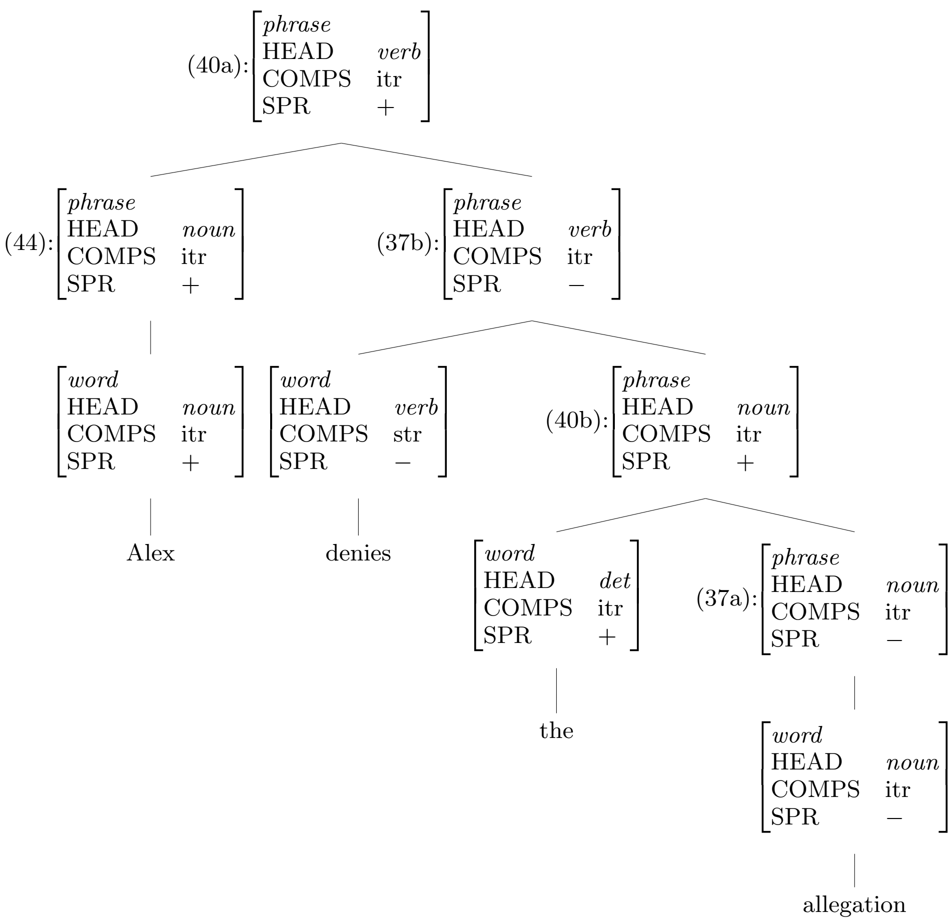
A. $\begin{bmatrix} \textit{department} & \\ \text{NAME} & \text{Metaphysics} \\ \text{TEL} & \text{650-723-4284} \end{bmatrix}$

B. $\begin{bmatrix} \text{TEL} & \boxed{23}\,\text{650-723-4284} \\ \text{CHAIR} & \begin{bmatrix}\text{TEL} & \boxed{23}\end{bmatrix} \end{bmatrix}$

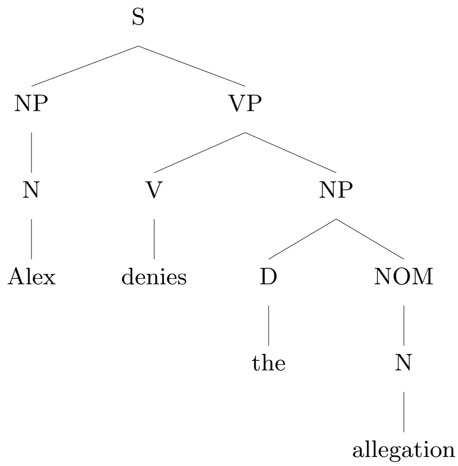
C. Constraints aren't compatible: The type of the second constraint (*individual*) is incompatible with the features PRESIDENT and FOUNDERS from the first constraint.

Exercise 2, page 68

A. We indicate the rule licensing each subtree by putting the number of the rule in parentheses next to the mother node of the subtree it licenses:



B. Replacing each of the feature structure node labels in (45) with the appropriate abbreviation gives the following tree:



Exercise 3, page 69

The claim is true: The only categories of type *phrase* licensed by the grammar are the mother nodes of phrase structure rules. These are all specificed as [COMPS itr]. Note, however, that we can't leave off the [COMPS itr] specification on the left hand side of (47). In this case, there is no other constraint in the grammar which would allow us to infer that specification. If that constraint were removed from the rule, we would end up generating such ungrammatical strings as (i) and (ii), because phrases licensed by such an underconstrained rule could then be the head daughters of the rules in (37).

(i)*Alex saw the movie on Tuesday the film.

(ii)*Alex saw the movie on Tuesday Sandy a book.

Chapter 4

Exercise 1, page 94

Here are some verbs for each class, along with illustrative sentences:

NP-S *convince, persuade, remind, tell*

$$\text{We} \left\{ \begin{array}{l} \text{convinced} \\ \text{persuaded} \\ \text{reminded} \\ \text{told} \end{array} \right\} \text{them (that) we knew the answer.}$$

NP-AP *call, consider, make, find*

$$\text{They} \left\{ \begin{array}{l} \text{called} \\ \text{considered} \\ \text{made} \\ \text{found} \end{array} \right\} \text{the proposal unpopular with voters.}$$

PP-S *admit, confess, remark, report*

$$\text{The witness} \left\{ \begin{array}{l} \text{admitted} \\ \text{confessed} \\ \text{remarked} \\ \text{reported} \end{array} \right\} \text{to the police that the evidence had been planted.}$$

PP-PP *speak, talk, whisper, write*

$$\text{Sandy} \left\{ \begin{array}{l} \text{spoke} \\ \text{talked} \\ \text{whispered} \\ \text{wrote} \end{array} \right\} \text{to Terry about the problem.}$$

Exercise 2, page 99

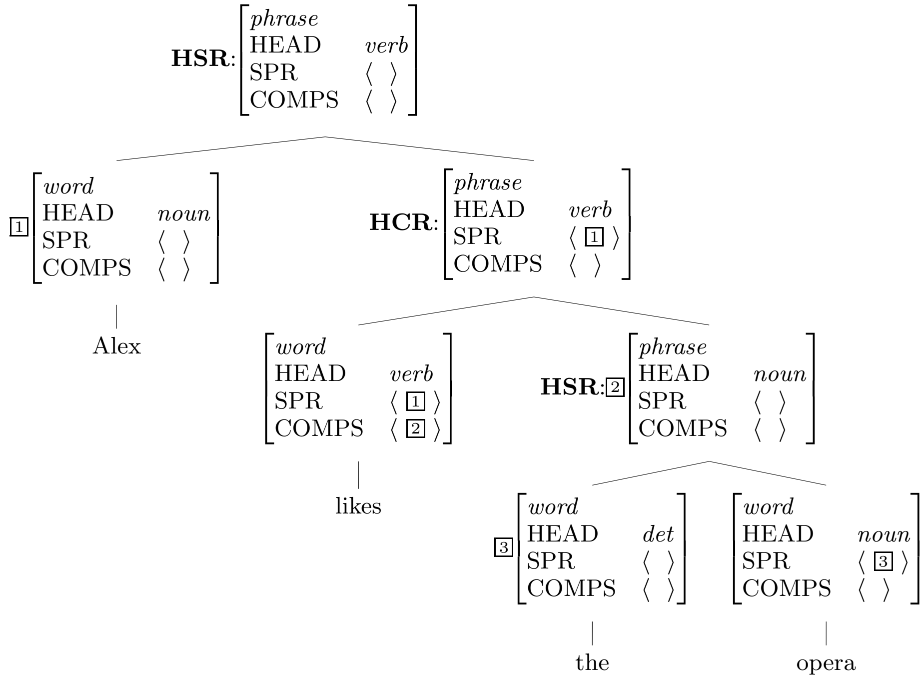
The data given in (10)–(12) support the following COMPS values.

happy	⟨ (PP | S) ⟩
magazine	⟨ (PP) ⟩
Newsweek	⟨ ⟩
report	⟨ (S) ⟩
book	⟨ ⟩
after	⟨ NP | S ⟩
during	⟨ NP ⟩
while	⟨ S ⟩

However, both *report* and *book* can also take PP complements. Therefore, they should have the COMPS values ⟨ (PP | S) ⟩ and ⟨ (PP) ⟩, respectively.

Exercise 3, page 105

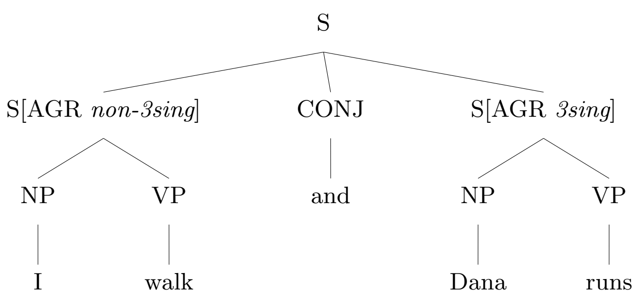
We indicate which rule licenses which subtree by putting the abbreviated name of the rule next to the mother node of the subtree it licenses:



Exercise 4, page 111

am [AGR *1sing*]   *are* [AGR *non-1sing*]

Exercise 5, page 116



The verbs *walk* and *runs* are lexically specified as [AGR *non-3sing*] and [AGR *3sing*], respectively. (By the SHAC, they share these AGR values with their specifiers.) Since the VPs are the heads of their respective Ss, the Head Feature Principle ensures that the S nodes will share those AGR values.

Chapter 5

Exercise 1, page 148

The adverb *today* must combine with a VP, which it modifies. By the Semantic Inheritance Principle, the INDEX of a modified VP like *aches today* is inherited from its head daughter, that is, the VP without the adverb (in this case, *aches*). So the INDEX value of the adverb *today* plays no role in the semantics of the modified VP *aches today*. Hence, we could assign it any value we want without changing the semantics of the modified VP.

Exercise 2, page 148

VP is an abbreviation for a feature structure whose HEAD value is of type *verb*, whose COMPS value is the empty list, and whose SPR list contains one element. V abbreviates a feature structure of type *word* whose HEAD value is of type *verb*. The feature structure of the node above *aches* satisfies both of these descriptions. Hence, it is both a V and a VP, as we have defined these abbreviations.

Chapter 6

Exercise 1, page 179

The difference is that *us*, unlike *letter* has empty SPR and COMPS lists. Recall from the discussion in Section 6.2.1 that it is the RESTR values of specifiers and complements that are essentially unbounded: a noun selecting for a PP complement might get a PP of any length. With empty SPR and COMPS lists, there isn't the same room for variation.

Chapter 7

Exercise 1, page 206

devour	$\langle$ NP , NP $\rangle$
elapse	$\langle$ NP $\rangle$
put	$\langle$ NP , NP , PP $\rangle$
rely	$\langle$ NP , PP $\rangle$
letter	$\langle$ DP (, PP) $\rangle$
of	$\langle$ NP $\rangle$
today	$\langle\ \rangle$
Venezuela	$\langle\ \rangle$

Exercise 2, page 215

The nodes labeled $\boxed{1}\mathrm{NP}_i$, V, D, N_j, P_i, and $\boxed{4}\mathrm{NP}_i$ all bear the feature ARG-ST. No other nodes do. This is because ARG-ST is a feature of *word*s only.

Chapter 8

Exercise 1, page 240

This lexical entry doesn't override any defeasible constraints. The rest of the constraints and where they come from are listed in the table below:

Constraint	type
$\begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{AGR} & \boxed{1}\end{bmatrix}\\ \text{SPR} & \langle \begin{bmatrix}\text{AGR} & \boxed{1}\end{bmatrix}\rangle\end{bmatrix}$	*infl-lxm*
HEAD *noun*	*cn-lxm*
AGR [PER 3rd]	*cn-lxm*
SPR ⟨ [HEAD *det*] ⟩	*cn-lxm*
COMPS ⟨ ⟩	*cn-lxm*
MODE ref	*cn-lxm*
SPR ⟨ [COUNT +] ⟩	*cntn-lxm*
$\begin{bmatrix}\text{INDEX} & i\\ \text{RESTR} & \left\langle\begin{bmatrix}\text{RELN} & \textbf{dog}\\ \text{INST} & i\end{bmatrix}\right\rangle\end{bmatrix}$	lexical entry

Exercise 2, page 246

The Case Constraint does account for accusative case on the objects of both argument-marking and predicational prepositions. As shown in (41b), argument-marking prepositions only select for one argument, but that argument will be outranked by the other arguments of the verb that selects for the PP (see Chapter 7 for details). Predicational prepositions select for two arguments, as shown in (41a), and the second one of these will be the object. Since it is outranked by the first argument of the preposition, it will be required to be [CASE acc] by the Case Constraint.

Chapter 10

Exercise 1, page 321

In order for the lexical entry in (23) to license sentences like (i), it would have an ARG-ST of the form in (ii) or (iii):

(ii) ⟨ NP , NP , VP ⟩
(iii) ⟨ NP , S ⟩

But its ARG-ST isn't like either of these: It is only two elements long, and the second element is a VP, not an S.

Chapter 12

Exercise 1, page 363

Examples like (i) and (ii) are now incorrectly allowed because the Imperative Rule is underconstrained. The daughter in that rule is required to be [FORM base], which is now also consistent with an ([INF +]) infinitival *to*-phrase. To rule these out, the Imperative Rule must be modified so as to require that the daughter also be [INF −].

Chapter 13

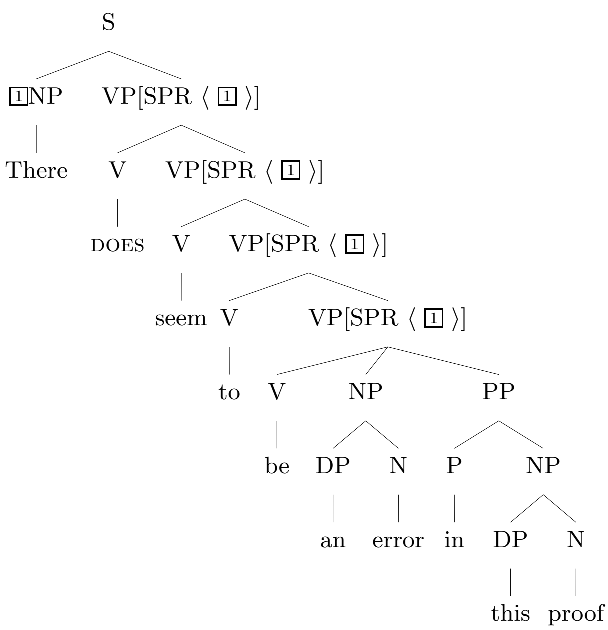
Exercise 1, page 399

In examples like the following, the boldfaced modals permit deontic interpretations, despite the fact that their subjects are nonreferential (dummies or idiom chunks):

(i) The mayor has decided that there **may** be a demonstration in front of City Hall.
(ii) There **must** be a signature at the bottom, or the application will not be considered.
(iii) If the play is to be a success, it **should** upset the audience that the hero dies.
(iv) Under the Patriot Act, tabs **can** be kept on anyone working with certain substances.
(v) The court ruled that advantage **could** be taken of such opportunities.

Thus, the syntactic tests for the raising/control distinction indicate that even when they are interpreted deontically, modals are raising verbs.

Exercise 2, page 403



The lexical entry for *be* requires a [FORM there] NP as its first argument. By the ARP, this same NP is on the SPR list of *be*. By the Valence Principle, the SPR value of the VP *be an error in this proof* is the same as the SPR value of *be*. The verbs *to*, *seem*, and *does* are all raising verbs, and thus subject to the constraint that their first argument be identified with their second argument's specifier. By the ARP, this first argument appears as the SPR value of each of these verbs. By the Valence Principle, it is also shared with the VPs they each head. Finally, the Head-Specifier Rule identifies the specifier requirement of the VP *does seem to be an error in this proof* with the NP *there*.

Exercise 3, page 413

The following is a sketch of an instantiation of the 3rd-Singular Verb Lexical Rule, with the lexeme *be* as input:

$$
\left[\begin{array}{ll}
\textit{i-rule} & \\
\text{INPUT} & \left\langle \text{be}\ ,\ \left[\begin{array}{ll} \textit{auxv-lxm} & \\ \text{SYN} & \boxed{4}\left[\begin{array}{ll}\text{HEAD} & [\text{AGR}\ \boxed{1}] \\ \text{VAL} & \left[\text{SPR}\ \ \langle\ [\text{AGR}\ \boxed{1}]\ \rangle\right]\end{array}\right] \\ \text{ARG-ST} & \boxed{B}\langle\ \text{X}\ ,\ \left[\text{PRED}\ \ +\right]\rangle \end{array}\right] \right\rangle \\
\text{OUTPUT} & \left\langle \text{is}\ ,\ \left[\begin{array}{ll} \textit{word} & \\ \text{SYN} & \boxed{4}\left[\begin{array}{ll}\text{HEAD} & \left[\begin{array}{ll}\text{FORM} & \text{fin} \\ \text{AGR} & \boxed{1}\textit{3sing}\end{array}\right] \\ \text{VAL} & \left[\begin{array}{ll}\text{SPR} & \langle\ \boxed{2}[\text{AGR}\ \boxed{1}]\ \rangle \\ \text{COMPS} & \langle\ \boxed{3}\ \rangle\end{array}\right]\end{array}\right] \\ \text{ARG-ST} & \boxed{B}\langle\ \boxed{2}[\text{CASE}\ \ \text{nom}]\ ,\ \boxed{3}[\text{PRED}\ \ +]\ \rangle \end{array}\right] \right\rangle
\end{array}\right]
$$

This sketch does not show all of the constraints contributed by either the lexical rule or the lexeme. It is simplified in order to highlight the constraints of relevance to the matter at hand. The SHAC gives us the identity between the AGR value of the HEAD and the SPR on the INPUT ($\boxed{1}$). This is carried over to the OUTPUT because *i-rule*s identify the entire SYN values of their INPUT and OUTPUT ($\boxed{4}$). The OUTPUT of this lexical rule (and therefore also the INPUT) is constrained to be [AGR *3sing*]. Therefore, the SPR of the OUTPUT must also be [AGR *3sing*]. Furthermore, the OUTPUT is subject to the ARP, which identifies the SPR of the OUTPUT with the first ARG-ST element. This means that the first element of the ARG-ST must be [AGR *3sing*].

The OUTPUT value above meets all of the constraints of the INPUT value of the Inversion Lexical Rule. Therefore, that INPUT value can be resolved to any such 3rd-Singular Verb Lexical Rule OUTPUT. The following is a sketch of an instantiation of the Inversion Lexical Rule with such an INPUT:

$$\begin{bmatrix} \textit{pi-rule} \\ \text{INPUT} & \left\langle \text{is}\ , \begin{bmatrix} \textit{word} \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \boxed{5}\begin{bmatrix} \text{FORM} & \text{fin} \\ \text{AUX} & + \\ \text{AGR} & \boxed{1}\textit{3sing} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \boxed{2}\begin{bmatrix}\text{AGR}\ \boxed{1}\end{bmatrix}\ \rangle \\ \text{COMPS} & \langle\ \boxed{3}\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \boxed{A}\langle\ \boxed{2}\begin{bmatrix}\text{CASE} & \text{nom}\end{bmatrix}, \boxed{3}\begin{bmatrix}\text{PRED} & +\end{bmatrix}\ \rangle \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \text{is}\ , \begin{bmatrix} \textit{word} \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \boxed{5}\begin{bmatrix}\text{INV} & +\end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \boxed{A} \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \boxed{A}\langle\ \boxed{2}\begin{bmatrix}\text{CASE} & \text{nom}\end{bmatrix}, \boxed{3}\begin{bmatrix}\text{PRED} & +\end{bmatrix}\ \rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

Once again, this sketch does not show all of the constraints contributed by the lexical rule or the lexical sequence used as the input. The crucial thing to notice here is that, although the SPR values are different between the INPUT and OUTPUT, the ARG-STs are identified. Therefore, through the chain of identities expressed by $\boxed{A}$, $\boxed{2}$, and $\boxed{1}$, the first ARG-ST element of the OUTPUT must be [AGR *3sing*]. By the ARP and the constraint on the SPR value of the OUTPUT of the Inversion Lexical Rule, the COMPS list of the OUTPUT is identical to its SPR list. Therefore, the first complement of inverted *is* is correctly predicted to be constrained to be 3rd-person singular.

Exercise 4, page 417

The outputs of the Ellipsis Lexical Rule can't be of type *auxv-lxm* because their ARG-ST specification (an ARG-ST of length one) is incompatible with the ARG-ST constraint on *auxv-lxm* (which is inherited from *srv-lxm*). They must be lexemes (so they can undergo the various inflectional rules) and they must have some fully specified type. You might think *siv-lxm* would work, but it inherits the constraint [AUX −] from *verb-lxm*. Although this is a defeasible constraint on *verb-lxm*, it cannot be overridden by the Ellipsis LR.

Chapter 14

Exercise 1, page 430

(12b) The selecting element is *rely*. It requires a PP as its complement. The filler *this* is an NP, and therefore has the wrong HEAD value.

(12c) The selecting element is *on*. It requires an NP as its complement. The filler *on this* is a PP, and therefore has the wrong HEAD value.

(12e) The selecting element is *trust*. It requires an NP complement. The filler *on this* is a PP, and therefore has the wrong HEAD value.

(13b) The selecting element is *on*. It requires an accusative NP complement. The filler *he* is nominative, and therefore has the wrong CASE value.

(14b) The selecting element is *between*. It requires a plural NP complement. The filler *couple* is singular, and therefore has the wrong NUM value.

Glossary

This glossary contains linguistic terms that either play a direct role in or are presupposed by this book. For further information, there are a number of dictionaries of linguistics, including Crystal 1985 and Trask 1993. (In the following definitions, 'q.v.' stands for Latin *quod vide* 'which see'.)

AAVE In informal speech, many African Americans use a distinctive variety of English known as African American Vernacular English, or AAVE for short. Also known as Black English, African American English, and Ebonics, both the phonology and aspects of the syntax of AAVE have been extensively studied.

absolutive *See* case.

accusative *See* case.

active A verb form or clause that is not in the passive is referred to as active. *See also* passive; voice.

affix An affix is a morphological element added to a stem to form another stem or a word. Two common types of affix are prefixes (e.g. *re-*, as in *reread*; *out-*, as in *outperform*) and suffixes (*-ed*, as in *visited*; *-s*, as in *visits*). Many languages exhibit other types of affix as well, including infixes (an element inserted into a stem) and circumfixes (e.g. a pair of elements wrapped around a stem).

agreement In many languages, the forms of certain elements can vary to indicate such properties as person [q.v.], number [q.v.], gender [q.v.], etc. Often, these variations are marked with affixes. Some grammatical relationships between pairs of linguistic elements require that they agree on these properties. In English, for example, present tense verbs are marked to indicate whether their subjects are third-person singular (with the suffix *-s*), and nouns indicate plurality (also with a suffix *-s*). The systematic covariation of the forms of the subject and verb is called 'subject-verb agreement'. Similarly, pronouns must agree with their antecedents in person, number, and (if third-person) gender. *See also* inflection.

anaphor *See* anaphora.

anaphora Certain expressions depend for their interpretation on their association with some other element in the discourse (usually earlier). Paradigm examples are pronouns like *he, her*, and *itself*; other examples include *do so* and verb phrase ellipsis. 'Anaphora' is the term for the relationship between such elements and their antecedents. The term 'anaphor' is sometimes used for all anaphoric elements and is sometimes reserved for only certain kinds (primarily reflexives [q.v.] and reciprocals [q.v.]). *See also* antecedent; binding; ellipsis.

antecedent This term is used for a linguistic element that is the basis for interpreting some anaphoric element occurring (typically later) in the sentence or discourse. In particular, pronouns are often described as referring back to their antecedents (or to the referents of their antecedents). *See also* anaphora.

argument (or **syntactic argument)** This is a general term for any phrase that is selected for by a lexical head, such as a complement or specifier. This usage derives from the semantic term 'argument', which refers to a component of a proposition that must occur with a given predicate or relation. For example, the meaning of the verb *wash* has two semantic arguments (the washer and the washed) that correspond to the two syntactic arguments of the verb *wash* (the subject and the object) – as in *Alex washed the car.* The simplest examples of (syntactic) arguments are noun phrases, but prepositional phrases and subordinate clauses can also function as arguments. *See also* complement; specifier; subcategorization; valence; argument structure.

argument-marking preposition English prepositions serve two distinct functions. In some cases, a preposition is used to indicate the role its object NP plays within the semantic structure of some predicate. In other cases, the preposition itself functions as a predicate, and its object is one of its arguments. In this text, the first kind of use is called an 'argument-marking preposition'. An example is the preposition *on* in *They rely on us.* We call the second kind of preposition 'predicational', illustrated by the use of *on* in *They are on the porch.*

argument structure In the theory developed in this text, the phrases that serve as arguments of a given head are listed in the value of a feature called ARGUMENT-STRUCTURE (ARG-ST). This term is also sometimes used in a less technical sense to designate the semantic relations between a head [q.v.] and its arguments. *See also* argument.

aspect Many languages have special grammatical elements for locating in time the situations referred to. Among the temporal notions often expressed are whether situations are in process or completed and whether they occur repeatedly. These notions are often called 'aspect', and words or affixes whose function is to express aspect are called 'aspectual markers'. *See also* perfective, progressive.

aspectual marker *See* aspect.

auxiliary This term refers to elements found in many languages that share the following semantic and syntactic characteristics: (i) they express such notions as time (past, present, future; continuation, completion), necessity, possibility, obligation, permission, negation, or questioning; and (ii) they occur in fixed positions in sentences, usually at or near the beginning or end. English auxiliaries are a special kind of verb. It is the auxiliary verb that is inverted with the subject in yes/no questions (e.g. *Did she fall?*) and that carries the negative suffix in contractions [q.v.] (e.g. *can't, won't*).

base form Almost all English verbs have an uninflected form that occurs after *to* in infinitives [q.v.] and after modals [q.v.], which we refer to as the 'base form'.

binding Pronouns are often said to be 'bound' by their antecedents [q.v.], and the term 'binding' is used to refer to the relationship between pronouns and their antecedents. The study of the principles governing pronominal anaphora [q.v.] is called 'binding theory'. *See also* reciprocal; reflexive.

case Certain words – particularly nouns and pronouns – can appear in different forms depending on their relationship to other elements in the sentence in which they appear. In English, for example, personal pronouns exhibit nominative case (e.g. *I, they*) or accusative case (e.g. *me, them*), depending on whether they are subjects or objects. In many languages, case is the primary way of indicating the roles of the noun phrases in the clause – that is, who did what to whom. Among the names of cases commonly used across languages are 'nominative', 'accusative', 'dative', 'genitive', 'ergative', and 'absolutive'. *See also* inflection.

clause A clause is a phrase that includes a predicate and all of its arguments and modifiers. The term is sometimes limited to phrases headed by a verb.

common noun Nouns are often divided into two kinds: proper and common. Proper nouns are names, denoting individual things by themselves, and do not normally take determiners, complements, or modifiers. Common nouns stand for kinds of things and take determiners [q.v.], modifiers [q.v.], and (sometimes) complements [q.v.]. In English orthography, proper nouns are conventionally capitalized, but common nouns are not.

competence In a number of works, Chomsky has distinguished between the (largely unconscious) knowledge of language that people have and how they put that knowledge to use in speech and writing. The former is called 'competence'; the latter 'performance'. The study of linguistic competence abstracts away from such factors as memory limitations, disfluencies, and speech errors. Work in generative grammar [q.v.] has concentrated largely on developing models of competence, though there has been much discussion of how such models relate to what is known about performance.

complement The lexical head [q.v.] of a phrase characteristically selects which arguments co-occur with it, and some of these are referred to as 'complements'. When the phrase's head is a verb, the complements include what are traditionally called direct and indirect objects, as well as some prepositional phrases and subordinate clauses [q.v.]. Subjects – and determiners of NPs – are arguments that are not complements, but specifiers. Complements occur as sisters to the lexical head in syntactic structure and, in English, follow the head. For example, a verb such as *hit* takes one complement, namely, an NP (e.g. *hit the ball*); *rely* takes a PP complement (e.g. *rely on Sandy*). A preposition such as *in* also takes a single NP complement (e.g. *in the box*). Some nouns can also take complements, such as *picture* which takes an optional PP complement (e.g. *picture of Kim*). *See also* argument; specifier.

complementizer The term 'complementizer' is close in meaning to the traditional term 'subordinate conjunction'. It is normally reserved for elements introducing clausal complements headed by a verb. In English, the use of *that* to introduce subordinate clauses [q.v.] (as in *It is remarkable that ice floats*) is the clearest example of a complementizer. *See also* complement.

conjunction (conjunct) Traditional grammarians use the term 'conjunction' to refer to words that connect two linguistic units in some way. In this text, we use it only for what are traditionally called 'coordinate conjunctions', that is, words connecting elements that are, intuitively, of equal status. In English, the paradigmatic coordinate conjunctions are *and* and *or*, though *but* and *nor* can also function in this way. The individual words or phrases that have been conjoined can be referred to as conjuncts. *See also* coordination.

constituent The term 'constituent' is used by linguists as a near synonym for 'phrase', meaning a part of a sentence that functions syntactically as a single unit. The difference is that 'constituent' is usually limited to phrases that are proper parts of larger expressions.

construct We refer to the MOTHER value of a construction instantiation [q.v] as a 'construct'.

construction This term has a traditional informal use, designating any sequence of words or types of words that pattern alike in some way. Thus, grammarians may talk of an 'imperative construction', a 'passive construction', or a 'filler-gap construction'. In Chapter 16 of this book, we introduce a more precise meaning: constructions are a type of feature structure containing the features MOTHER and DAUGHTERS. The grammar rules and lexical rules of the theory developed in Chapters 1-15 are seen as two kinds of constructions in this sense.

construction instantiation Constructions [q.v] are descriptions, specifying the values of some features and leaving others unspecified. A construction instantiation is a fully resolved feature structure that is consistent with the specifications of some construction.

context-free grammar (CFG) A context-free grammar is a particular type of formal system that has proved very useful in the precise characterization of computer languages and also serves as the starting point for much work in syntactic theory. CFGs consist of an initial symbol [q.v.], a finite lexicon with words classified into grammatical categories [q.v.], and a finite collection of rules of the form A $\rightarrow \omega$, where A is a single symbol (representing a type of phrase), and ω is a finite string of lexical and/or phrasal categories.

contraction Reduced forms of words are sometimes combined with other words (that would typically occur adjacent to the reduced words) to form a new word; these are referred to as 'contractions'. English examples include combinations of a finite auxiliary [q.v.] verb with a reduced form of *not* to produce such words as *isn't* and *can't*, as well as simple contraction of finite auxiliaries, e.g. *They're arriving tomorrow* and *Kim's here.*

control Some complements have no overt specifier, but are interpreted as if they had subjects with the same reference as (i.e. coindexed with) another complement to the same predicate. For example, in both *Pat tries to be on time* and *We urged Pat to be on time* the individual Pat is understood as the person who is meant to be on time. This relationship (between two noun phrases, the second typically an unexpressed subject) is referred to as 'control'; in this case with the NP *Pat* being the 'controller' of the unexpressed subject of the infinitival phrase. Control predicates are not to be confused with raising [q.v.] predicates (like *continue* and *expect*), one of whose arguments actually is simultaneously an argument of another complement. A number of syntactic and semantic diagnostics distinguish these two types of predicate. *See also* raising.

coordination This term refers to the linking of two words or phrases of equivalent syntactic status (i.e. neither coordinate (or conjoined) element is subordinate to the other). An example of a coordinated clause is *Kim cooked the food and Sandy washed the dishes. See also* conjunction.

copula This term is used by traditional grammarians to refer to verbs with little semantic content, which serve to link a subject with a predicate. In English, the copula is *be.*

coreference/coreferential Two referring expressions that refer to the same entity are called 'coreferential', and the relationship between them is called 'coreference'. *See also* anaphora.

count noun Common nouns are divided into two subclasses known as 'count' and 'mass' according to which determiners they can combine with. Count nouns co-occur with *a(n), several, few*, etc.; mass nouns co-occur with *much* and can occur in the singular with no determiner. This distinction is correlated with a semantic distinction: mass nouns usually are used to refer to substances and count nouns to (sets of) entities. A portion of a substance (e.g. helium) is still the same substance, whereas a portion of an entity (e.g. a bicycle) is not usually an entity of the same kind. This correlation is not perfect, however, as evidenced by the mass noun *furniture* and by minimal pairs like *cabbage* (which can be either count or mass) vs. *lettuce* (which, for many speakers, must be mass).

dative Many languages employ a case called dative to grammatically encode the participation of some argument (q.v.) in a given situation as recipient, goal, or beneficiary. See *case.*

declarative/interrogative/imperative These are terms used in the classification of sentence types. Declarative sentences are used to make a statement (or – equivalently for our purposes – to assert the truth of a proposition [q.v.]), as in *The mayor is reading a book.* Interrogative sentences are used to ask questions [q.v.], as in *What are they doing?* Imperative sentences are used to give orders (or to issue 'directives' [q.v.]), as in *Read a book!*

defeasible A constraint is said to be 'defeasible' if it can be overridden – that is, if it allows for the existence of exceptions. *See also* inviolable.

demonstrative Expressions used for referring through direct indication (often accompanied by pointing) are called 'demonstratives'. The best examples in English are *this, that, these*, and *those.*

description We model elements of language using feature structures, which are either atoms or else functions (in the mathematical sense). Such functions map features into feature structures (atoms or functions). They are moreover total functions, in the sense that they map every feature in a relevant domain into a value. Often, rather than specifying a full feature structure, it is convenient to describe a class of feature structures by specifying only the values of some features. Many of the constructs of our grammar – notably lexical types, lexical entries, and lexical rules – are descriptions in this sense.

descriptive grammar *See* prescriptive grammar.

determiner The sorts of specifiers [q.v.] that nouns take are called 'determiners'. These include articles (*a, the*, etc.), quantifiers [q.v.] (*some, every, many, two*, etc.), and possessives [q.v.] (*my, Kim's*, etc.). *See also* specifier.

directive A directive is a particular kind of semantic object, characteristically associated with imperative [q.v.] sentences. It is the kind of object that can be issued by uttering such a sentence, and fullfilled by causing the conditions associated with the sentence to be met. The grammar in this text puts [MODE dir] into the semantics of imperative sentences.

discourse This term refers to units of language longer than a sentence – for example, dialogues or paragraphs.

distribution Linguists use this term to refer to the set of total environments – or contexts – in which some linguistic unit can occur.

ditransitive verb In this book, verbs that take two NP objects are called 'ditransitive'. The standard example is *give*, in examples like *The teacher gave the students an exam*. *See also* intransitive verb; transitive verb; valence.

dummy Words that evidently have no meaning and serve only to fill some grammatical function are sometimes called 'dummies'. The paradigm examples in English are the *there* that occurs in existential sentences (e.g. *There is a seat available* and the *it* of extraposition [q.v.] (e.g. *It is fortunate that you have a seat*). Other terms used for these are 'expletives' and 'pleonastic' elements.

ellipsis Ellipsis means 'leaving out' or 'omitting': in certain contexts, parts of a sentence can be omitted if their interpretation is reconstructable. An example is the following case of verb phrase ellipsis, where the bracketed material may be left out:

(i) Pat won't taste the soup, but Chris will [taste the soup].

See also anaphora.

ergative *See* case.

existential *be*/existential *there* English has a special construction for expressing existence, involving the dummy *there* as subject and forms of the verb *be*. These are called 'existential'. *See also* dummy.

expletive *See* dummy.

extraction Some grammatical theories deal with long-distance dependencies [q.v.] by means of rules that move the filler from the gap position to where it actually appears. Since the position of the filler is always less deeply embedded in the tree than the position of the gap, this is sometimes referred to as 'extraction' of the filler. This terminology is carried over into the present text in the Subject Extraction Lexical Rule.

extraposition Predicates that can take complementizer [q.v.] phrases (i.e. *that*-clauses) as subjects can also occur with a dummy *it* as subject and the CP as the last complement. The latter construction is called 'extraposition', and is exemplified by the following:

(i) It bothers Alex that Dana left.

The term is also sometimes used for expressions in which a complement or modifier is separated from its head by intervening material, as in *A review appeared of Lee's latest book*. *See also* dummy.

feature structure A standard way of representing linguistic information is in terms of complexes of features and values. A feature can be thought of as a dimension along which different linguistic entities (such as words, phrases, or sentences) may differ, and values identify locations on those dimensions. A feature-value pair models a property of a linguistic entity that distinguishes it in a linguistically interesting way from some other entities. For example, the feature PERSON (PER) in English has three possible values, namely '1st', '2nd', and '3rd'. It is a property of the word *you* that it is second-person, and we represent that with the feature-value pair [PER 2nd]. A feature structure can thus be treated as set of feature-value pairs in which no feature is paired with more than one value. (Feature structures are functions and **types** are used to organize the domains and ranges of such functions.) Values of features in our theory may themselves be feature structures, or even lists of feature structures. Feature structure descriptions are standardly given in terms of matrices, listing feature names paired with their values, also known as 'feature specifications'. *See also* inheritance hierarchy; type.

filler *See* long-distance dependency.

finite-state grammar Finite-state grammars are a type of formal system sometimes used to describe certain rather simple artificial languages. They are mathematically equivalent to regular expressions. *See also* context-free grammar; regular expression.

finite verb A finite verb is one that is marked for tense [q.v.] (present or past, in English).

gap *See* long-distance dependency.

gender The nouns in many languages divide into classes, differing in their patterns of inflection and agreement. In a number of languages (e.g. French and German), these noun classes are referred to as 'genders', because nouns used to refer to males or females (of any species) are generally (though not invariably) grammatically masculine or feminine, respectively. In English, gender is marked grammatically only on third-person singular pronouns (*he, she*, and *it*) and is virtually always predictable from the sex of the referent.

generative grammar Chomsky introduced this term based on the idea that a grammar is a formal system for generating the sentences of a language. The term is now used in at least three distinct senses, to denote: (i) work in the Chomskyan tradition (fairly broadly conceived); (ii) an explicit system of rules, principles, and/or constraints that characterizes all and only the well-formed sentences of a language; or (iii) the system in the mind or brain of a speaker that makes language use possible.

genitive *See* case.

grammatical category Words and phrases can be classified in various ways, any of which can be called a 'grammatical category'. The term is usually used to refer to parts of speech [q.v.], such as noun, verb, etc., as well as types of phrases, such as noun phrase, verb phrase, and sentence.

head The constituent [q.v.] of a phrase that is grammatically the most important constituent of that phrase is called the 'head' of the phrase. The head usually determines the category of the phrase, as well as many of its other properties. Thus noun phrases have nouns as heads, verb phrases have verbs as heads, etc. The term is used ambiguously to refer to the word that functions as head of the phrase and any subphrase containing that word. For example, in *the destruction of the city*, both *destruction* and *destruction of the city* can be called heads of the phrase.

idiom Some combinations of words have interpretations that are not fully predictable from the meanings that those same words have in other contexts. These are known as 'idioms'. English examples include *take advantage* to mean (roughly) 'exploit', *keep tabs on* for 'monitor', and *kick the bucket* for 'die'. Parts of an idiom are sometimes called 'idiom chunks', e.g. *advantage* in *take advantage*. Idiom chunks play a central role in one of the diagnostics for distinguishing raising [q.v.] predicates from control [q.v.] predicates.

imperative *See* declarative.

infinitive Certain kinds of nonfinite verbs are referred to as 'infinitives'. English infinitives are preceded by *to*, which we analyze as a verb with the feature [INF +], taking a [FORM base] verb phrase as its complement. In many other languages, the infinitive verb form is marked with special affixes.

inflection Languages often add affixes to words to mark the syntactic function or relationships of the word in the sentence. For example, present tense verbs in English are usually inflected with the suffix *-s* when the subject is third-person singular, and past tense verbs are inflected with *-ed*. The term may also be used to refer to the affix itself. Among the common uses of inflectional affixes are to indicate tense [q.v.], agreement [q.v.], number [q.v.] (singular or plural), and case [q.v.]. The theory in this book employs lexical rules [q.v.] to account for the relationships among different forms of a lexeme. Note, however, that we restrict the term 'inflectional lexical rule' to those that map lexemes to words; other instances of what some might call 'inflection' (e.g. the participial forms of verbs) are handled by means of other types of lexical rule.

inheritance hierarchy The elements of some domains of study can naturally be organized into classes, based on shared properties. Some classes can be further subdivided into subclasses, with additional shared properties. The organization of such domains can be thought of as a hierarchy, with the most inclusive class (encompassing the entire domain) at the top, and the most restricted classes at the bottom. In between are various classes of interest.The properties associated with particular classes are inherited by their subclasses, and ultimately by their individual members. Domains organized in this way are referred to as 'inheritance hierarchies'. In linguistics, inheritance hierarchies have been used to organize lexical information, among other things. *See also* type.

initial symbol Grammars characterize languages. But languages can be conceived of in a variety of ways: as consisting of sentences, of phrases, of any expressions that can serve as stand-alone utterances, etc. A formal theory of grammar must include a specification of which of the expressions it characterizes are to be regarded as those that constitute the language. The initial symbols of a formal theory state what is to count as an element of the language. In this book, the initial symbol definition specifies conditions that phrases must satisfy if they can stand alone, i.e. be used in isolation to communicate a message.

interrogative *See* declarative.

intonation This term is used to refer to the patterns of pitch in speech.

intransitive verb A verb that does not take any NP objects is referred to as 'intransitive'. A standard example in English is *die*. *See also* ditransitive verb; transitive verb; valence.

inversion Grammarians use this term fairly generally to refer to any construction in which two elements appear with their typical ordering reversed. In this text, it is used in particular for sentences (mostly questions) in which a finite auxiliary [q.v.] verb precedes the subject, as in *Are you sleeping?*

inviolable A constraint is said to be 'inviolable' if the grammar never allows it to be violated. Constraints that are not inviolable are said to be 'defeasible' [q.v.].

island constraint While long-distance dependencies can, in principle, stretch over arbitrary distances, there are some pairings of filler and gap positions that are not possible. For example, a gap inside a CP subject cannot, in general, be paired with a filler outside that CP, as in **Which candidate did [that I voted for __] bother you.* The constraints on possible filler-gap pairings are known as 'island constraints'. *See also* long-distance dependency.

Kleene star It is useful in the formal representation of languages (both natural and artificial) to allow certain patterns to be repeated any finite number of times (including zero). The standard notation for this is a superscripted asterisk, known as the 'Kleene star' (after the mathematician Stephen Kleene). For example, ab^*c is shorthand for the infinite set of strings: *ac, abc, abbc, abbbc, ...* 'Kleene plus', denoted by a superscripted plus sign, means any nonzero number of repetitions. *See also* regular expression.

lexeme The term 'word' is used ambiguously to mean either a particular form, such as *sees*, or a set of related forms such as *see, sees, saw, seen*, and *seeing*. To avoid this ambiguity, linguists sometimes posit an abstract entity called a 'lexeme' that gives rise to a family of related words. *See also* word.

lexical entry Information about individual words [q.v.] that must be stipulated is put into the lexicon [q.v.] in the form of descriptions that we call lexical entries. They are ordered pairs, consisting of a phonological form (description) and a partial feature structure description. Fully resolved lexical sequences [q.v.] consistent with lexical entries can serve as the INPUT values of lexical rules [q.v.].

lexical rule Lexical rules are one of the mechanisms (along with the type hierarchy [q.v.]) used to capture generalizations within the lexicon. Families of related words – such as the different inflectional forms of a verb – can be derived from a single lexical entry [q.v.] by means of lexical rules. We formalize lexical rules as a type of feature structure with features INPUT and OUTPUT. There are three subtypes of lexical rules: derivational (relating lexemes [q.v.] to lexemes), inflectional (relating lexemes to words [q.v.]), and post-inflectional (relating words to words).

lexical rule instantiation Our lexical rules [q.v] are descriptions, specifying the values of some features and leaving others unspecified. A lexical rule instantiation is a fully resolved feature structure that is consistent with the specifications of some lexical rule.

lexical sequence Ordered pairs that can serve as the INPUT and OUTPUT values of lexical rules [q.v.] are called lexical sequences. They consist of a phonological form and a fully resolved feature structure.

lexicalism Lexicalism often refers to the doctrine that (1) the internal structure of words is independent of how words are put together to make sentences, and (2) words are the atoms of syntactic combination. For example, in a lexicalist theory, bound morphemes (inflectional affixes that must be attached to a word) are not treated as independent syntactic elements, as they are in most (nonlexicalist) versions of Transformational Grammar (see Appendix B). Theories of grammar also differ in their organization and in where they locate syntactic information. Some theories (e.g. Generalized Phrase Structure Grammar) have rich systems of rules and relatively impoverished lexical entries. Others (e.g. Categorial Grammar or Lexical Functional Grammar) have highly structured lexical entries and a small number of very general rule schemata. 'Lexicalism' is sometimes also used to distinguish the latter sort of theory.

lexicon The list of all words [q.v.] (or lexemes [q.v.]) of a language is called its 'lexicon'. The lexicon is the repository of all idiosyncratic information about particular words, including syntactic, semantic, and phonological information. In some theories of grammar, the lexicon can also contain a great deal more systematic information, organized by a type hierarchy [q.v.] and/or lexical rules.

long-distance dependency Certain constructions, including *wh*-questions, topicalization, and relative clauses, permit an element in one position to fill the grammatical role associated with another position. The two positions can be arbitrarily far apart. For example, in *Which student did the principal say that the teacher thought was responsible?* the NP *which student* functions as the subject of *was responsible*, although they are separated by most of the sentence. Such constructions are called 'long-distance dependencies' (LDDs). Elements like *which student* in the above example are called 'fillers', and the position normally associated with the filler's role (in this case, immediately preceding *was responsible*) is called the 'gap'. *See also* island constraints.

main clause *See* root sentence.

modal The English verbs *can, could, may, might, must, shall, should, will*, and *would*, along with their negated forms (*can't*, etc.) are referred to as 'modals' or 'modal verbs'. They share the following properties: they function only as finite verbs [q.v.]; they exhibit auxiliary behavior (negation, inversion, contraction, and ellipsis); they take base form VP complements; and they show no agreement [q.v.] (i.e. no third-person singular *-s* suffix). Some other languages have similar syntactically distinctive classes of words expressing necessity, possibility, obligation, and permission; these are also known as modals. *See also* auxiliary.

mass noun *See* count noun.

model Understanding real world phenomena may be enhanced by investigating mathematical entities that share certain properties with the phenomena in question. Such mathematical entities are called 'models' of the real-world phenomena. Models are useful because they make it possible to abstract away from incidental properties and focus on those that are of theoretical interest. With respect to the grammar we develop in this text, we use the word 'model' in two different senses. On the one hand, the grammar as a whole is a model of (a fragment of) the English language, or of speakers' knowledge of English. On the other hand, fully resolved feature structures are models of linguistic entities. In this sense, 'model' contrasts with 'description' [q.v.].

modifier Most phrases consist of a head [q.v.], together with that head's arguments [q.v.]. Semantically, the head typically denotes either a situation or an individual, and the arguments denote essential associated entities. In addition, phrases may contain modifiers, which serve to place further restrictions on the situation or individual picked out by the phrase as a whole. Modifiers can take a wide variety of syntactic forms, including adjectives and adjective phrases, adverbs and adverbial phrases, prepositional phrases, and modifying clauses (such as relative clauses). *See also* argument structure.

morphology This term refers ambiguously to the study of word structure – how words are put together out of stems and affixes – or to word structure itself.

negation Languages include devices for reversing or contradicting the meaning or truth conditions of expressions, a semantic effect known as 'negation'. In English, the most common element expressing negation is the word *not*.

nominalization Nominalizations are nouns constructed out of words of other categories, usually through affixation. An example is *destruction*, derived from the verb *destroy* through the affixation of *-tion* (together with some other modifications). The term 'nominalization' is also used to refer to a process of turning verbs and adjectives into nouns.

nominative *See* case.

number Most English nouns take different forms depending on whether they can head NPs that refer to single entities or multiple entities, e.g. *some dog/dogs, some man/men*. Similarly, present tense [q.v.] verbs with third-person subjects have different forms depending on whether the subjects are singular or plural. The term 'number' is used for such distinctions. Some languages also mark number on other types of words, e.g. adjectives may be marked for the number of the noun they modify. There are also languages that make finer number distinctions than just singular vs. plural, notably languages that have special 'dual' forms for expressions referring to sets with exactly two members.

orthography This term refers to written representations of language. For example, the plural of the noun *doe* and the present tense form of the verb *do* that goes with a third-person singular subject share the same orthography (namely, 'does'), although their pronunciations (and almost everything else about them) are different.

overgenerate A grammar that licenses sentences that are not part of the language the grammar writer is trying to analyze is said to 'overgenerate'. This term is usually used when a proposal for part of a grammar of a natural language licenses strings that native speakers of that language say are not well-formed sentences of the language. *See also* undergenerate.

paradigm Certain words have multiple inflectional forms. For example, verbs in English typically change their form depending on whether they are past or present tense, and their present-tense forms depend on the person and number of the subject. They also have a variety of nonfinite forms. The full array of inflectional forms of a word is known as its 'paradigm'. *See also* inflection.

paraphrase Two sentences are said to be paraphrases of one another if they differ in form but convey the same meaning.

parsing This term refers to the process of assigning a tree structure to a sentence. Many computer systems designed to process natural language include components for parsing, and much psycholinguistic research is concerned with discovering what parsing mechanisms humans use in language comprehension.

part of speech This is the traditional term for lexical categories (i.e. categories of words), based on a combination of semantic and distributional criteria. Among the standard parts of speech are noun, verb, adjective, preposition, adverb, and conjunction. *See also* grammatical category.

participle Certain nonfinite verbs – usually ones that share some properties with adjectives – are referred to as 'participles'. English has three types of participles: present participles, which end in *-ing* and usually follow some form of *be*; past participles, which usually end in *-ed* or *-en* and follow some form of *have*; and passive participles, which look exactly like past participles but indicate the passive voice [q.v.]. The three participles of *eat* are illustrated in the following sentences:

(i) Termites are eating the house.
(ii) Termites have eaten the house.
(iii) The house was eaten by termites.

passive Many languages have a construction in which the grammatical subject of a verb plays the same semantic role that the object plays when the verb in question appears elsewhere (in active [q.v.] forms). The term 'passive' is used to refer both to this construction, and to the verb whose arguments' roles are at issue. In English, the passive form of the verb looks exactly like the past participle and is usually preceded by a form of *be*; a prepositional phrase headed by *by* is also common, and is used for marking what would be the subject if the verb were not passive. An example is *The dog was attacked (by wombats)*. *See also* participle; voice.

perfective Many languages have special verb forms or constructions used to indicate that the event denoted by the verb is completed. These are referred to as 'perfective' (or just 'perfect') in aspect. The English perfective involves the combination of *have* with a past participle [q.v.], as in *The dog has eaten the cake. See also* aspect.

performance *See* competence.

person Many languages distinguish grammatically among expressions referring to the speaker, to the hearer, and to third parties. This is called the expression of 'person'. Reference to the speaker or a set including the speaker is called 'first person'; reference to (sets including) the adressee(s) is called 'second person'; and everything else is called 'third person'. Person distinctions are clearest with pronouns, since these are the most common forms used to refer to the speaker and hearer. But in some languages nouns also show person marking, and verbs and adjectives may agree with nouns and pronouns in person.

phonetics Phonetics is the study of the acoustic or articulatory properties of speech sounds.

phonology Phonology is the study of the sound systems of languages, i.e. the systematic grammatical patterns in the distribution [q.v.] of speech sounds.

possessive Many languages have grammatical mechanisms for indicating a relation of possession between the referents of two NPs. When one noun or NP is marked as the possessor of another, this marking is referred to as the 'possessive'. In English, the possessive is marked by *'s* attached at the end of the noun phrase functioning as the 'possessor'. There is also a set of determiners that express possession (*my*, *our*, *your*, etc.). These are called 'possessive pronouns'.

pragmatics The information conveyed by a linguistic expression in a particular instance of use is typically much more than just its literal (or 'linguistic') meaning. The study of how linguistic meaning contributes more generally to communication is called (linguistic) 'pragmatics'. *See also* semantics.

predicational preposition *See* argument-marking preposition.

prefix *See* affix.

prescriptive grammar Much of traditional work in grammar is concerned with setting norms – that is, dictating that some usages are 'incorrect' and hence to be avoided. Modern linguists refer to this as 'prescriptive grammar' (or just 'prescriptivism'). Most scientific work on grammar purports instead to be 'descriptive', seeking systematic explanations for the way the language is actually used.

productive A relationship between two linguistic forms is said to be 'productive' if it generalizes to novel forms. For example, the use of the suffix *-ing* to mark the present participle form of a verb is productive, since it gets applied to new coinages (as in *faxing*). Productivity is usually thought of as a matter of degree, with exceptionless relationships counting as more productive than those with exceptions.

progressive Special verb forms or constructions used to indicate that the event denoted by the verb is in progress are referred to as 'progressive' aspect. The English progressive involves combination of *be* with a present participle [q.v.], as in *The dog is eating the cake. See also* aspect.

proper noun *See* common noun.

proposition A proposition is a particular kind of abstract object, specifically, the sort of thing that can be true or false. Propositions are also what one asserts, believes, denies, etc. Declarative sentences characteristically express propositions, a fact represented in this text by putting [MODE prop] in the semantics of such sentences.

quantifier Words or phrases used to restrict the number or amount of some referent are called 'quantifiers'. In English, these include such expressions as *all, each, some, many, few, two, more than half*, etc.

question A question is a particular kind of semantic object, specifically, the sort of thing that can be asked and answered. Interrogative sentences are characteristically associated with questions, a fact represented in this text by the presence of [MODE ques] in the semantics of such sentences.

raising Some predicates take one more syntactic argument than semantic argument. In these cases, the extra syntactic argument functions as the subject of another complement and must obey any special co-occurrence restrictions imposed by that complement. These predicates are called 'raising' predicates. Raising is exemplified by the sentences *Pat continues to be on time* and *We expected Pat to be on time.* In these examples, *Pat*, though a syntactic argument of *seem* and *expect*, is semantically an argument only of *be on time.* A semantically empty dummy [q.v.] is possible with raising predicates, where it would not be possible in the corresponding positions with control predicates: *There continued/*tried to be demonstrations on campus. See also* control.

reciprocal A reciprocal pronoun is one that expresses a mutual relationship, such as the English pronoun *each other. See also* anaphora.

referent This term is used for the entity (e.g. a person, object, notion, or situation) that is denoted by (a use of) a linguistic expression.

reflexive Many languages use special forms of pronouns when the subject and object refer to the same individual or individuals, e.g. the English forms ending in *-self* or *-selves.* These are called 'reflexives' or 'reflexive pronouns'. It is common for these pronouns also to be acceptable in some other environments, but those environments differ from language to language. *See also* anaphora; binding.

regular expression It is possible to characterize the well-formed expressions of some simple formal languages by means of a few abbreviatory devices. One system that has proved very useful in some contexts involves templates, made up of words and/or categories of words, together with parentheses (to indicate optionality), a disjunction symbol (to indicate alternatives), and Kleene star [q.v.] (and/or Kleene plus), to indicate arbitrary numbers of repetitions of a sequence. Such templates are called 'regular expressions'. *See also* finite-state grammar.

relative clause These are clauses that are used to modify nouns or noun phrases. A relative clause characteristically contains either a gap or a pronoun understood to be coreferential with the noun it modifies.

root sentence The traditional distinction between main clause and subordinate clause is motivated in part by the fact that certain phenomena seem to be restricted to main clauses, e.g. the inversion of finite auxiliaries [q.v.] in English interrogatives (compare: *Can I do it?* vs. *I wonder whether I can do it*). Consequently, some version of this distinction has been maintained in most formal theories of grammar. The term 'root sentence' is sometimes used for main clauses, or, more technically, a phrase of category S that is not dominated by anything else. *See also* subordinate clause.

saturated In the system of grammar developed in this book, a saturated phrase is one that is specified as [SPR $\langle\ \rangle$] and [COMPS $\langle\ \rangle$]. The intuition behind this is that headed phrases can be thought of as being generated bottom-up, starting from the lexical head, via a regime of cancelling elements from the head's valence specifications. For example, a verb combines first with however many complements are on its COMPS list to build a VP (a verbal phrase that is [COMPS $\langle\ \rangle$] but [SPR $\langle$ NP $\rangle$]); the resulting (SPR-)unsaturated phrase then combines with the subject NP to build a saturated phrase, i.e. an S.

semantics Semantics is the branch of linguistics concerned with the study of linguistic meaning. Linguists also use the locution 'the semantics of' some expression as a way of talking about the literal interpretation of that expression. Not all information that is conveyed by the utterance of an expression is part of its semantics, but the line between literal meaning and what is conveyed can be hard to draw. At a minimum, the semantics of a (declarative) sentence is usually taken to include a specification of the conditions under which it would be true. *See also* pragmatics.

situation Situations are what natural language sentences are about: events or states (real or imaginary), involving entities, their properties, and their relations to one another.

specifier We use the term 'specifier' to cover subjects of clauses, determiners of noun phrases, and certain other constituents that are neither heads of the phrases they appear in nor complements to the heads. In English, the specifier of a phrase precedes its head [q.v.] and complements [q.v.]. *See also* determiner; complement.

subcategorization Lexical heads differ according to how many and what types of things they must combine with in order to make complete phrases. Each grammatical category [q.v.] (that is, part of speech [q.v.]) can be divided into subcategories, based on the valence, or combinatoric potential, of the particular words. When we talk of the subcategorization of a verb (or other type of head), we mean the restrictions on which sorts of phrases it can combine with. Another common locution is to say that a given verb 'subcategorizes for' a certain phrase, meaning that it combines with such a phrase. *See also* valence.

subordinate clause A subordinate clause is one that is dependent on, and usually a constituent [q.v.] of, another clause [q.v.]. An example of a subordinate clause is *when Kim went* in *Sandy came when Kim went*. *See also* root sentence.

suffix *See* affix.

tense Finite verbs come in different forms depending on the time they denote; these forms are called 'tenses'. English has present and past tense, exemplified by the present tense forms *walk* and *walks*, and by the past tense form *walked*. Some languages also have future tenses, but English uses other means (e.g. the modal [q.v.] *will*) to express future time. *See also* aspect; finite verb.

transitive verb Verbs that take an NP object are called 'transitive'. The term can also be used for other parts of speech that can take objects, e.g. prepositions. It is sometimes taken to encompass both simple transitive verbs (that is, those taking a single object) and ditransitive verbs. A standard example of a transitive verb is *hit*. *See also* ditransitive verb; intransitive verb; valence.

type Elements of any collection can be sorted into types, based on properties they have in common. In the theory presented in this text, linguistic entities (e.g. words and phrases) are described by means of feature structures [q.v.]. Particular features are appropriate only for certain types of entity, and constraints on possible feature-value pairings are also associated with particular types. *See also* feature structure; inheritance hierarchy; lexicon.

type hierarchy Types (q.v.) are organized into a hierarchy that determines the properties of linguistic entities through the mechanism of constraint inheritance. The type hierarchy is especially important for capturing regularities in the lexicon. *See* inheritance hierarchy.

undergenerate A grammar that fails to license sentences that the grammar writer wants it to generate is said to 'undergenerate'. *See also* overgenerate.

unification The operation of unification merges two feature structure descriptions into a description that contains all the information in both. Two feature structure descriptions can unify so long as the information in them is consistent – that is, so long as there is no feature for which they have conflicting values. The unification simply consists of all of the features and values specified in the two feature structure descriptions. Unification is an operation for merging descriptions of a certain form whose effect is equivalent to conjunction of constraints.

universal grammar Many linguists claim that there is a great deal in common among the grammars of the world's languages. Most advocates of this position believe that the commonalities exist because linguistic structure is largely determined by human biology. The term 'universal grammar' is used to mean three subtly different things: (i) what is common to the world's languages; (ii) linguists' representations of these commonalities; and (iii) the biological structures that are claimed to underlie the common features.

unsaturated *See* saturated.

valence This term is used (by analogy with the chemical term) to refer to the combinatoric potential of words and phrases. In this text, the VALENCE (VAL) features are those that specify this kind of information. Specifically, the VAL features SPR and COMPS for the verb *put* specify that it requires a subject NP, an object NP, and a PP in order to form a clause. *See also* argument; argument structure; ditransitive verb; intransitive verb; transitive verb.

voice This term refers to the way the semantic arguments of the verb are expressed grammatically. The term is used in English primarily to distinguish active voice and passive voice, but some other languages have far richer systems of voice. *See also* active; passive.

word This term is used in many different ways. In this text, a word is a particular form derived from a lexeme by some inflectional rule. *See also* lexeme, lexicon.

References

Abeillé, Anne, and Owen Rambow (Eds.). 2000. *Tree-Adjoining Grammars: Formalisms, Linguistic Analysis and Processing.* Stanford, CA: CSLI Publications.

Akmajian, Adrian, Susan Steele, and Thomas Wasow. 1979. The category AUX in Universal Grammar. *Linguistic Inquiry* 10:1–64.

Akmajian, Adrian, and Thomas Wasow. 1974. The constituent structure of VP and AUX and the position of the verb BE. *Linguistic Analysis* 1:205–245.

Andrews, Avery. 1982. The representation of case in modern Icelandic. In *The Mental Representation of Grammatical Relations* (Bresnan 1982b).

Archangeli, Diana, and D. Terence Langendoen (Eds.). 1997. *Optimality Theory: An overview.* Oxford: Blackwell Publishing.

Arnold, Jennifer, Maria Fagnano, and Michael K. Tanenhaus. 2002. Disfluencies signal, theee, um, new information. *Journal of Psycholinguistic Research.*

Asudeh, Ash. in press. A licensing theory for Finnish. In D. C. Nelson and S. Manninen (Eds.), *Generative Approaches to Finnic and Saami Linguistics.* Stanford: CSLI Publications.

Bach, Emmon. 1979. Control in Montague Grammar. *Linguistic Inquiry* 10:515–31.

Bach, Emmon. 1989. *Informal Lectures on Formal Semantics.* Albany: SUNY Press.

Baker, Mark. 2001. *The Atoms of Language: The Mind's Hidden Rules of Grammar.* New York: Basic Books.

Bar-Hillel, Yehoshua, and Eliyahu Shamir. 1960. Finite-state languages: Formal representations and adequacy problems. *The Bulletin of the Research Council of Israel* 8F:155–166. Reprinted in Y. Bar-Hillel (1964) Language and Information: Selected Essays on Their Theory and Application Reading, MA: Addison-Wesley.

Barbosa, Pilar, Danny Fox, Paul Hagstrom, Martha McGinnis, and David Pesetsky (Eds.). 1998. *Is the Best Good Enough? Optimality and Competition in Syntax.* Cambridge, MA: MIT Press.

Barlow, Michael, and Charles Ferguson (Eds.). 1988. *Agreement in Natural Language: Approaches, Theories, Descriptions.* Stanford.

Barwise, Jon, and John Etchemendy. 1989. Semantics. In *Foundations of Cognitive Science* (Posner 1989).

Bates, Elizabeth, and Brian MacWhinney. 1989. Functionalism and the competition model. In B. MacWhinney and E. Bates (Eds.), *The Cross-linguistic Study of Sentence Processing*. Cambridge: Cambridge University Press.

Baugh, John. 1983. *Black Street Speech: Its History, Structure, and Survival*. Austin: University of Texas Press.

Bear, John. 1981. Gaps as syntactic features. Technical report, Center for Cognitive Science, University of Texas at Austin.

Bender, Emily, and Dan Flickinger. 1999. Peripheral constructions and core phenomena: Agreement in tag questions. In G. Webelhuth, J.-P. Koenig, and A. Kathol (Eds.), *Lexical and Constructional Aspects of Linguistic Explanation*, 199–214. Stanford, CA: CSLI.

Bender, Emily M. 2001. *Syntactic Variation and Linguistic Competence: The Case of AAVE Copula Absence*. PhD thesis, Stanford University.

Bender, Emily M. 2002. Review of R. Martin, D. Michaels and J. Uriagereka (eds.), *Step by Step: Essays on Minimalist Syntax in Honor of Howard Lasnik*. *Journal of Linguistics* 38:432–439.

Bender, Emily M., Dan Flickinger, and Stephan Oepen. 2002. The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In J. Carroll, N. Oostdijk, and R. Sutcliffe (Eds.), *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, 8–14.

Bever, Thomas. 1970. The cognitive basis for linguistic structure. In J. R. Hayes (Ed.), *Cognition and the Development of Language*. New York: Wiley.

Bloomfield, Leonard. 1933. *Language*. New York: H. Holt and Company.

Bouma, Gosse, Rob Malouf, and Ivan A. Sag. 2001. Satisfying constraints on extraction and adjunction. *Natural Language and Linguistic Theory* 19:1–65.

Brame, Michael K. 1979. *Essays Toward Realistic Syntax*. Seattle: Noit Amrofer.

Bresnan, Joan. 1978. A realistic transformational grammar. In M. Halle, J. Bresnan, and G. A. Miller (Eds.), *Linguistic Theory and Psychological Reality*. Cambridge, MA: MIT Press.

Bresnan, Joan. 1982a. Control and complementation. In *The Mental Representation of Grammatical Relations* (Bresnan 1982b).

Bresnan, Joan (Ed.). 1982b. *The Mental Representation of Grammatical Relations*. Cambridge, MA: MIT Press.

Bresnan, Joan. 1982c. The passive in lexical theory. In *The Mental Representation of Grammatical Relations* (Bresnan 1982b).

Bresnan, Joan. 1995. Linear order, syntactic rank, and empty categories: On weak crossover. In *Formal Issues in Lexical-Functional Grammar* (Dalrymple et al. 1995).

Bresnan, Joan. 2000. Optimal syntax. In J. Dekkers, F. van der Leeuw, and J. van de Weijer (Eds.), *Optimality Theory: Phonology, Syntax and Acquisition*, 334–385. Oxford: Oxford University Press.

Bresnan, Joan. 2001. *Lexical-Functional Syntax*. Oxford and Cambridge, MA: Blackwell.

Briscoe, Edward, and Ann Copestake. 1999. Lexical rules in constraint-based grammar. *Computational Linguistics* 25(4):487–526.

Briscoe, Edward, Ann Copestake, and Valeria de Paiva (Eds.). 1993. *Inheritance, Defaults, and the Lexicon.* Cambridge: Cambridge University Press.

Brody, Michael. 1995. *Lexico-Logical Form: A Radically Minimalist Theory.* Cambridge, MA: MIT Press.

Burzio, Luigi. 1986. *Italian Syntax.* Dordrecht: Reidel.

Cacciari, Cristina, and Patrizia Tabossi (Eds.). 1993. *Idioms: Processing, Structure, and Interpretation.* Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Cameron, Deborah. 1995. *Verbal Hygiene.* London and New York: Routledge.

Campbell, Lyle. 1985. *The Pipil Language of El Salvador.* New York: Mouton Publishers.

Carpenter, Bob. 1992. *The Logic of Typed Feature Structures: with Applications to Unification Grammars, Logic Programs, and Constraint Resolution.* Cambridge: Cambridge University Press.

Carpenter, Bob. 1997. *Type-Logical Semantics.* Cambridge, MA: MIT Press.

Chierchia, Gennaro, and Sally McConnell-Ginet. 1990. *Meaning and Grammar: An Introduction to Semantics.* Cambridge, MA: MIT Press.

Chomsky, Noam. 1957. *Syntactic Structures.* The Hague: Mouton.

Chomsky, Noam. 1959. Review of B. F. Skinner's *Verbal Behavior. Language* 35:26–58. Reprinted in J. A. Fodor and J. J. Katz (Eds.) (1964), *The Structure of Language: Readings in the Philosophy of Language*, Englewood Cliffs, N.J.: Prentice-Hall.

Chomsky, Noam. 1963. Formal properties of grammars. In R. D. Luce, R. Bush, and E. Galanter (Eds.), *Handbook of Mathematical Psychology*, Vol. Volume II. New York: Wiley.

Chomsky, Noam. 1965. *Aspects of the Theory of Syntax.* Cambridge, MA: MIT Press.

Chomsky, Noam. 1966. *Topics in the Theory of Generative Grammar.* The Hague: Mouton.

Chomsky, Noam. 1970. Remarks on nominalization. In R. A. Jacobs and P. S. Rosenbaum (Eds.), *Readings in English Transformational Grammar.* Waltham, MA: Ginn-Blaisdell.

Chomsky, Noam. 1972. *Language and Mind, enlarged edition.* New York: Harcourt, Brace, Jovanovich.

Chomsky, Noam. 1973. Conditions on transformations. In S. Anderson and P. Kiparsky (Eds.), *A Festschrift for Morris Halle.* New York: Holt, Rinehart and Winston.

Chomsky, Noam. 1975. *The Logical Structure of Linguistic Theory.* Chicago: University of Chicago Press. Written in 1955 and widely circulated in mimeograph form.

Chomsky, Noam. 1977. On wh-movement. In *Formal Syntax* (Culicover et al. 1977).

Chomsky, Noam. 1980. *Rules and Representations.* New York: Columbia University Press.

Chomsky, Noam. 1981. *Lectures on Government and Binding.* Dordrecht: Foris.

Chomsky, Noam. 1986a. *Barriers.* Cambridge, MA: MIT Press.

Chomsky, Noam. 1986b. *Knowledge of Language: Its Nature, Origin, and Use.* New York: Praeger.

Chomsky, Noam. 1995. *The Minimalist Program.* Cambridge, MA: MIT Press.

Chomsky, Noam. 2002. Beyond explanatory adequacy. *MIT Occasional Papers in Linguistics* 20.

Clark, Eve V., and Herbert H. Clark. 1979. When nouns surface as verbs. *Language* 55:767–811.

Clark, Herbert H., and Jean E. Fox Tree. 2002. Using uh and um in spontaneous speaking. *Cognition.*

Clark, Herbert H., and Thomas Wasow. 1998. Repeating words in spontaneous speech. *Cognitive Psychology* 37:201–242.

Copestake, Ann. 1992. *The Representation of Lexical Semantic Information.* PhD thesis, University of Sussex. Published as Cognitive Science Research Paper CSRP 280, 1993.

Copestake, Ann. 2002. *Implementing Typed Feature Structure Grammars.* Stanford: CSLI Publications.

Copestake, Ann, Dan Flickinger, Robert Malouf, Susanne Riehemann, and Ivan A. Sag. 1995. Translation using Minimal Recursion Semantics. In *Proceedings of The Sixth International Conference on Theoretical and Methodological Issues in Machine Translation*, 15–32, Leuven.

Copestake, Ann, Daniel Flickinger, Ivan A. Sag, and Carl Pollard. 1999. Minimal Recursion Semantics: an introduction. Unpublished ms., Stanford University. Available online at: http://www-csli.stanford.edu/~aac/papers/newmrs.ps.

Copestake, Ann, Alex Lascarides, and Daniel Flickinger. 2001. An algebra for semantic construction in constraint-based grammars. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, Toulouse, France. Association for Computational Linguistics.

Crain, Steven, and Mark Steedman. 1985. On not being led up the garden path: The use of context by the psychological syntax processor. In D. R. Dowty, L. Karttunen, and A. M. Zwicky (Eds.), *Natural Language Processing.* Cambridge: Cambridge University Press.

Crystal, David. 1985. *A Dictionary of Linguistics and Phonetics.* London: B. Blackwell in association with A. Deutsch.

Culicover, Peter, Adrian Akmajian, and Thomas Wasow (Eds.). 1977. *Formal Syntax.* New York: Academic Press.

Dalrymple, Mary. 1993. *The Syntax of Anaphoric Binding.* Stanford: CSLI Publications.

Dalrymple, Mary. 2001. *Lexical Functional Grammar.* New York: Academic Press. Syntax and Semantics, Volume 34.

Dalrymple, Mary, and Ronald M. Kaplan. 2000. Feature indeterminacy and feature resolution. *Language* 76(4):759–798.

Dalrymple, Mary, Annie Zaenen, John Maxwell III, and Ronald M. Kaplan (Eds.). 1995. *Formal Issues in Lexical-Functional Grammar.* Stanford: CSLI Publications.

Davidson, Donald. 1980. *Essays on Actions and Events.* Oxford: Clarendon Press; New York: Oxford University Press.

Davis, Anthony. 2001. *Linking by Types in the Hierarchical Lexicon.* Stanford: CSLI Publications.

de Swart, Henriëtte. 1998. *Introduction to Natural Language Semantics.* Stanford: CSLI Publications.

Dowty, David, Robert Wall, and Stanley Peters. 1981. *Introduction to Montague Semantics.* Dordrecht: D. Reidel.

Elman, Jeffrey, Elizabeth Bates, Mark Johnson, Annette Karmiloff-Smith, Domenico Parisi, and Kim Plunkett. 1996. *Rethinking Innateness – A Connectionist Perspective on Development.* MIT Press. A Bradford Book.

Emonds, Joseph. 1975. *A Transformational Approach to Syntax.* New York: Academic Press.

Epstein, Samuel, and Daniel Seely (Eds.). 2002. *Derivation and Explanation in the Minimalist Program.* Oxford: Blackwell's.

Everaert, Martin, Erik-Jan Van der Linden, Andre Schenk, and Rob Schreuder (Eds.). 1995. *Idioms: Structural and Psychological Perspectives.* Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Ferguson, Charles. 1971. Absence of copula and the notion of simplicity: A study of normal speech, baby talk, foreigner talk, and pidgins. In D. Hymes (Ed.), *Pidginization and Creolization of Languages.* New York: Cambridge University Press.

Fillmore, Charles J., Paul Kay, Laura Michaelis, and Ivan A. Sag. forthcoming. *Construction Grammar.* Stanford: CSLI Publications.

Fillmore, Charles J., Paul Kay, and Mary Catherine O'Connor. 1988. Regularity and idiomaticity in grammatical constructions: The case of *let alone*. *Language* 64(3):501–538.

Flickinger, Daniel, Carl Pollard, and Thomas Wasow. 1985. Structure sharing in lexical representation. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, Morristown, N.J. Association for Computational Linguistics.

Fodor, Janet D. 1995. Comprehending sentence structure. In *Language* (Gleitman and Liberman 1995).

Fodor, Jerry A. 1983. *The Modularity of Mind.* Cambridge, MA: MIT Press.

Fodor, Jerry A., Thomas Bever, and Merrill Garrett. 1974. *The Psychology of Language.* New York: McGraw-Hill.

Fraser, Bruce. 1970. Idioms within a transformational grammar. *Foundations of Language* 6:22–42.

Frege, Gottlob. 1892. On sense and reference. *Zeitschrift für Philosophie und philosophische Kritik* 100:25–50. Translation (under the title 'On Sense and Meaning') appears in Geach and Black (1980).

Gamut, L. T. F. 1991. *Logic, Language, and Meaning.* Chicago: University of Chicago Press.

Garrett, Merrill. 1990. Sentence processing. In D. Osherson and H. Lasnik (Eds.), *Language: An Invitation to Cognitive Science*, Vol. Volume 1, first edition. Cambridge, MA: MIT Press.

Gazdar, Gerald. 1981. Unbounded dependencies and coordinate structure. *Linguistic Inquiry* 12:155–184.

Gazdar, Gerald. 1982. Phrase structure grammar. In P. Jacobson and G. K. Pullum (Eds.), *The Nature of Syntactic Representation*. Dordrecht: Reidel.

Gazdar, Gerald, Ewan Klein, Geoffrey K. Pullum, and Ivan A. Sag. 1985. *Generalized Phrase Structure Grammar*. Cambridge, MA: Harvard University Press and Oxford: Basil Blackwell.

Gazdar, Gerald, and Geoffrey K. Pullum. 1981. Subcategorization, constituent order, and the notion 'head'. In M. Moortgat, H. van der Hulst, and T. Hoekstra (Eds.), *The Scope of Lexical Rules*. Dordrecht: Foris.

Gazdar, Gerald, Geoffrey K. Pullum, and Ivan A. Sag. 1982. Auxiliaries and related phenomena in a restrictive theory of grammar. *Language* 58:591–638.

Geach, Peter, and Max Black (Eds.). 1980. *Translations from the Philosophical Writings of Gottlob Frege, third edition*. Oxford: Basil Blackwell.

Ginzburg, Jonathan, and Ivan A. Sag. 2000. *Interrogative Investigations: The Form, Meaning and Use of English Interrogatives*. Stanford: CSLI Publications.

Gleitman, Lila R., and Mark Liberman (Eds.). 1995. *Language*. Vol. 1 of *An Invitation to Cognitive Science*. Cambridge, Mass: MIT Press. Second edition.

Goldberg, Adele E. 1995. *A Construction Grammar Approach to Argument Structure*. Chicago: University of Chicago Press.

Green, Lisa. 1993. *Topics in African American English: The Verb System Analysis*. PhD thesis, University of Massachusetts at Amherst.

Green, Lisa J. 2002. *African American English: A Linguistic Introduction*. Cambridge: Cambridge University Press.

Greenbaum, Sidney. 1996. *The Oxford English Grammar*. Oxford: Oxford University Press.

Grice, H. Paul. 1989. *Studies in the Way of Words*. Cambridge, MA: Harvard University Press.

Haegeman, Liliane. 1994. *Introduction to Government and Binding Theory, second edition*. Oxford and Cambridge, MA: Basil Blackwell.

Harman, Gilbert. 1963. Generative grammar without transformation rules: A defense of phrase structure. *Language* 39:597–616.

Harris, Randy Allen. 1993. *The Linguistic Wars*. Oxford: Oxford University Press.

Harris, Zellig S. 1970. *Papers in Structural and Transformational Linguistics*. Dordrecht: Reidel.

Hauser, Marc D., Noam Chomsky, and W. Tecumseh Fitch. 2002. The faculty of language: What is it, who has it, and how did it evolve? *Science* Nov 22 2002:1569–1579.

Hockett, Charles Francis. 1958. *A Course In Modern Linguistics*. New York: Macmillan.

Hopcroft, John E., Rajeev Motwani, and Jeffrey D. Ullman. 2001. *Introduction to Automata Theory, Languages, and Computation.* Reading, MA: Addison-Wesley.

Huck, Geoffrey J., and John A. Goldsmith. 1995. *Ideology and Linguistic Theory.* London and New York: Routledge.

Huddleston, Rodney, and Geoffrey K. Pullum. 2002. *The Cambridge Grammar of the English Language.* Cambridge: Cambridge University Press.

Hudson, Richard. 1984. *Word Grammar.* Oxford: Blackwell.

Hudson, Richard. 1990. *English Word Grammar.* Oxford: Blackwell.

Hudson, Richard. 1998. Word grammar. In V. Agel, et al (Ed.), *Dependency and Valency: An International Handbook of Contemporary Research.* Berlin: Walter de Gruyter.

Jackendoff, Ray. 1972. *Semantic Interpretation in Generative Grammar.* Cambridge, MA: MIT Press.

Jackendoff, Ray. 1975. Morphological and semantic regularities in the lexicon. *Language* 51:639–671.

Jackendoff, Ray. 1994. *Patterns in the Mind.* New York: Basic Books.

Jackendoff, Ray. 2002. *Foundations of Language: Brain, Meaning, Grammar, Evolution.* Oxford: Oxford University Press.

Johnson, David. 1977. On relational constraints on grammars. In P. Cole and J. Sadock (Eds.), *Syntax and Semantics, Volume 8: Grammatical Relations.* New York: Academic Press.

Johnson, David, and Shalom Lappin. 1999. *Local Constraints vs. Economy.* Stanford: CSLI Publications.

Johnson, David, and Paul Postal. 1980. *Arc Pair Grammar.* Princeton: Princeton University Press.

Joshi, Aravind K. 1990. Processing crossed and nested dependencies: An automaton perspective on the psycholinguistic results. *Language and Cognitive Processes* 5:1–27.

Joshi, Aravind K. 2003. Tree-adjoining grammars. In R. Mitkov (Ed.), *Handbook of Computational Linguistics.* Oxford University Press.

Joshi, Aravind K., Tilman Becker, and Owen Rambow. 2000. Complexity of scrambling: A new twist to the competence-performance distinction. In A. Abeillé and O. Rambow (Eds.), *Tree-Adjoining Grammars: Formalisms, Linguistic Analysis and Processing.* Stanford: CSLI Publications.

Joshi, Aravind K., Leon S. Levy, and Masako Takahashi. 1975. Tree adjunct grammars. *Journal of Computer and System Sciences* 10(1).

Jurafsky, Daniel, and James H. Martin. 2000. *Speech and Language Processing – An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Upper Saddle River, New Jersey: Prentice Hall.

Kager, René. 1999. *Optimality Theory.* Cambridge: Cambridge University Press.

Kaplan, Ronald M. 1975. *Transient Processing Load in Relative Clauses.* PhD thesis, Harvard University.

Kaplan, Ronald M., and Annie Zaenen. 1989. Long-distance dependencies, constituent structure and functional uncertainty. In M. R. Baltin and A. S. Kroch (Eds.), *Alternative Conceptions of Phrase Structure*, 17–42. University of Chicago Press. Reprinted in Mary Dalrymple, Ronald Kaplan, John Maxwell, and Annie Zaenen (Eds.) (1995), *Formal Issues in Lexical-Functional Grammar*, Stanford: CSLI Publications, Pages 137–165.

Kasper, Robert, Bernard Kiefer, Klaus Netter, and K. Vijay-Shanker. 1995. Compilation of HPSG to TAG. In *Proceedings of the Association for Computational Linguistics (ACL '95)*, 92–99.

Katz, Jerrold J., and Paul M. Postal. 1964. *An Integrated Theory of Linguistic Descriptions*. Cambridge, MA: MIT Press.

Katz, Jerrold J., and Paul M. Postal. 1991. Realism versus conceptualism in linguistics. *Linguistics and Philosophy* 14:515–554.

Kay, Martin. 1979. Functional grammar. In C. Chiarello (Ed.), *Proceedings of the Fifth Annual Meeting of the Berkeley Linguistic Society*.

Kay, Paul. 1995. Construction grammar. In J. Verschueren, J.-O. Ostman, and J. Blommaert (Eds.), *Handbook of Pragmatics*. Amsterdam and Philadelphia: John Benjamins.

Kay, Paul. 2002. An informal sketch of a formal architecture for construction grammar. *Grammars* 5:1–19.

Kay, Paul, and Charles J. Fillmore. 1999. Grammatical constructions and linguistic generalizations: The what?s x doing y? construction. *Language* 75.1:1–33.

Keenan, Edward, and Bernard Comrie. 1977. Noun phrase accessibility and universal grammar. *Linguistic Inquiry* 8:63–99.

Keenan, Edward, and Dag Westerståhl. 1997. Generalized quantifiers in linguistics and logic. In *Handbook of Logic and Language*, 837–893, Amsterdam and Cambridge, MA. North-Holland and MIT Press.

Kim, Jong-Bok. 2000. *The Grammar of Negation: A Constraint-Based Approach*. Stanford: CSLI Publications.

Kim, Jong-Bok, and Ivan A. Sag. 2002. French and English negation without head-movement. *Natural Language and Linguistic Theory* 20(2):339–412.

King, Paul J. 1989. *A Logical Formalism for Head-Driven Phrase Structure Grammar*. PhD thesis, University of Manchester.

Koster, Jan. 1987. *Domains and Dynasties, the Radical Autonomy of Syntax*. Dordrecht: Foris.

Kurtzman, Howard S., and Maryellen C. MacDonald. 1993. Resolution of quantifier scope ambiguities. *Cognition* 48:243–279.

Labov, William. 1969. Contraction, deletion, and inherent variability of the English copula. *Language* 45:715–762.

Labov, William. 1972. *Language in the Inner City: Studies in the Black English Vernacular*. Philadelphia: University of Pennsylvania Press.

Labov, William. 1995. The case of the missing copula: The interpretation of zeroes in African-Amerircan English. In *Language* (Gleitman and Liberman 1995), 25–54.

Labov, William, Paul Cohen Cohen, Clarence Robins, and John Lewis. 1968. A study of the nonstandard English of Negro and Puerto Rican speakers in New York City. Technical Report Final Report, Cooperative Research Project No. 3288, United States Office of Education.

Lakoff, George. 1987. *Women Fire and Dangerous Things.* Chicago and London: University of Chicago Press.

Langacker, Ronald. 1987. *Foundations of Cognitive Grammar (vol 1).* Stanford, CA: Stanford University Press.

Lappin, Shalom, Robert Levine, and David Johnson. 2000a. The revolution confused: A reply to our critics. *Natural Language and Linguistic Theory* 18:873–890.

Lappin, Shalom, Robert Levine, and David Johnson. 2000b. The structure of unscientific revolutions. *Natural Language and Linguistic Theory* 18:665–671.

Lappin, Shalom, Robert Levine, and David Johnson. 2001. The revolution maximally confused. *Natural Language and Linguistic Theory* 19:901–919.

Larson, Richard. 1995. Semantics. In *Language* (Gleitman and Liberman 1995), 361–380.

Lascarides, Alex, Edward Briscoe, Nicholas Asher, and Ann Copestake. 1996. Order independent and persistent typed default unification. *Linguistics and Philosophy* 19(1):1–89.

Lascarides, Alex, and Ann Copestake. 1999. Default representation in constraint-based frameworks. *Computational Linguistics* 25(1):55–105.

Lasnik, Howard. 1976. Remarks on coreference. *Linguistic Analysis* 2:1–22. Reprinted in Lasnik (1989), *Essays on Anaphora*, Dordrecht: Kluwer.

Lasnik, Howard. 1995. The forms of sentences. In *Language* (Gleitman and Liberman 1995).

Lasnik, Howard, Marcela Depiante, and Arthur Stepanov. 2000. *Syntactic Structures Revisited: Contemporary Lectures on Classic Transformational Theory.* Cambridge, MA: MIT Press.

Legendre, Géraldine, Jane Grimshaw, and Sten Vikner (Eds.). 2001. *Optimality-Theoretic Syntax.* Cambridge: MIT Press.

Levin, Beth. 1993. *English Verb Classes and Alternations: A Preliminary Investigation.* Chicago: University of Chicago Press.

Levine, Robert D., and Ivan A. Sag. 2003. *WH*-nonmovement. *Gengo Kenky: Journal of the Linguistic Society of Japan* 123.

MacDonald, Maryellen C., Neal J. Pearlmutter, and Mark S. Seidenberg. 1994. The lexical nature of syntactic ambiguity resolution. *Psychological Review* 101(4).

Malouf, Robert. in press. Cooperating constructions. In E. Francis and L. Michaelis (Eds.), *Linguistic Mismatch: Scope and Theory.* Stanford: CSLI Publications.

Marcus, Gary F. 2001. *The Algebraic Mind: Integrating Connectionism and Cognitive Science.* MIT Press.

Marcus, Gary F. 2004. *The Birth of the Mind.* Basic Books.

Marslen-Wilson, William D., and Lorraine K. Tyler. 1987. Against modularity. In J. Garfield (Ed.), *Modularity in Knowledge Representation and Natural Language Understanding.* MIT Press.

Matthews, Peter. 1993. *Grammatical Theory in the United States: from Bloomfield to Chomsky.* Cambridge: Cambridge University Press.

McCawley, James D. 1971. Tense and time reference in English. In C. J. Fillmore and D. T. Langendoen (Eds.), *Studies in Linguistic Semantics.* New York: Holt, Rinehart, and Winston.

McCloskey, James. 1979. *Transformational Syntax and Model-Theoretic Semantics.* Dordrecht: Reidel.

Meurers, Walt Detmar. 1999. *Lexical Generalizations in the Syntax of German Non-Finite Constructions.* PhD thesis, Seminar für Sprachwissenschaft, Universität Tübingen, Tübingen, Germany. published 2000 as Volume 145 in Arbeitspapiere des SFB 340, ISSN 0947-6954/00.

Meurers, Walt Detmar. 2001. On expressing lexical generalizations in HPSG. *Nordic Journal of Linguistics* 24(2):161–217. Special issue on 'The Lexicon in Linguistic Theory'.

Michaelis, Laura, and Knud Lambrecht. 1996. Toward a construction-based theory of language function: The case of nominal extraposition. *Language* 72:215–248.

Milsark, Gary. 1977. Towards an explanation of certain peculiarities of the existential construction in English. *Linguistic Analysis* 3:1–31.

Montague, Richard. 1970. Universal grammar. *Theoria* 36:373–398. Reprinted in Richmond Thomason, (Ed.) (1974), *Formal Philosophy*, New Haven: Yale University Press.

Moortgat, Michael. 1997. Categorial type logics. In J. van Benthem and A. ter Meulen (Eds.), *Handbook of Logic and Language*, 93–177. Amsterdam: Elsevier and Cambridge, MA: MIT Press.

Morrill, Glynn. 1994. *Type Logical Grammar.* Dordrecht: Kluwer.

Mufwene, Salikoko S., Guy Bailey, John R. Rickford, and John Baugh (Eds.). 1998. *African-American English: Structure, History, and Use.* London: Routledge.

Nevin, Bruce (Ed.). 2003. *The Legacy of Zellig Harris – Language and Information into the 21st Century, Volume 1.* Amsterdam and Philadelphia: John Benjamins.

Nevin, Bruce, and Stephen Johnson (Eds.). 2003. *The Legacy of Zellig Harris – Language and Information into the 21st Century, Volume 2.* Amsterdam and Philadelphia: John Benjamins.

Newmeyer, Frederick J. 1986. *Linguistic Theory in America, Second Edition.* London: Academic Press.

Nunberg, Geoffrey. 1983. The grammar wars. *The Atlantic Monthly* 256(6):31–58.

Nunberg, Geoffrey, Ivan A. Sag, and Thomas Wasow. 1994. Idioms. *Language* 70:491–538.

Partee, Barbara H. 1995. Lexical semantics and compositionality. In *Language* (Gleitman and Liberman 1995).

Partee, Barbara H., Alice ter Meulen, and Robert Wall. 1990. *Mathematical Methods in Linguistics.* Dordrecht: Kluwer.

Pearlmutter, Neal J., and Maryellen MacDonald. l992. Plausibility and syntactic ambiguity resolution. In *Proceedings of the 14th Annual Conference on Cognitive Science*, 498–503, Hillsdale, N.J. Erlbaum.

Pedersen, Holger. 1959. *The Discovery of Language: Linguistic Science in the Nineteenth Century.* Bloomington: Indiana University Press. Translated by John Webster Spargo.

Penn, Gerald. 2000. *The Algebraic Structure of Attributed Type Signatures.* PhD thesis, Carnegie Mellon University (Computer Science). Document available online at http://www.cs.toronto.edu/~gpenn/publications.html.

Perlmutter, David (Ed.). 1983. *Studies in Relational Grammar 1.* Chicago: University of Chicago Press.

Perlmutter, David, and Paul Postal. 1977. Toward a universal characterization of passivization. In *Proceedings of the 3rd Annual Meeting of the Berkeley Linguistics Society*, Berkeley. University of California, Berkeley. Reprinted in Perlmutter (1983).

Perlmutter, David, and Scott Soames. 1979. *Syntactic Argumentation and the Structure of English.* Berkeley: University of California Press.

Pinker, Steven. 1994. *The Language Instinct.* New York: Morrow.

Pollard, Carl, and Ivan A. Sag. 1987. *Information-Based Syntax and Semantics, Volume 1: Fundamentals.* Stanford: CSLI Publications.

Pollard, Carl, and Ivan A. Sag. 1992. Anaphors in English and the scope of binding theory. *Linguistic Inquiry* 23:261–303.

Pollard, Carl, and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar.* Chicago: University of Chicago Press.

Pollock, Jean-Yves. 1989. Verb movement, Universal Grammar, and the structure of IP. *Linguistic Inquiry* 20:365–424.

Posner, Michael I. (Ed.). 1989. *Foundations of Cognitive Science.* Cambridge, MA: MIT Press.

Postal, Paul. 1964. *Constituent Structure: A Study of Contemporary Models of Syntactic Description.* Bloomington: Research Center for the Language Sciences, Indiana University.

Postal, Paul. 1974. *On Raising.* Cambridge, MA: MIT Press.

Postal, Paul. 1986. *Studies of Passive Clauses.* Albany: SUNY Press.

Postal, Paul, and Brian Joseph (Eds.). 1990. *Studies in Relational Grammar 3.* Chicago: University of Chicago Press.

Postal, Paul, and Geoffrey K. Pullum. 1988. Expletive noun phrases in subcategorized positions. *Linguistic Inquiry* 19:635–670.

Prince, Alan, and Paul Smolensky. 1993. *Optimality Theory: Constraint Interaction in Generative Grammar.* Tech Report RuCCS-TR-2. ROA-537: Rutgers University Center for Cognitive Science.

Quirk, Randoph, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1972. *A Grammar of Contemporary English.* London and New York: Longman.

Quirk, Randoph, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language.* London and New York: Longman.

Radford, Andrew. 1997. *Syntactic Theory and the Structure of English: A Minimalist Approach.* New York and Cambridge: Cambridge University Press.

Richter, Frank. 1999. RSRL for HPSG. In V. Kordoni (Ed.), *Tübingen Studies in Head-Driven Phrase Structure Grammar*, no. 132 in Arbeitsberichte des SFB 340, 74–115. Tübingen: Universität Tübingen.

Richter, Frank. 2000. *A Mathematical Formalism for Linguistic Theories with an Application in Head-Driven Phrase Structure Grammar*. PhD thesis, Universität Tübingen.

Rickford, John R., and Russell J. Rickford. 2000. *Spoken Soul: The Story of Black English*. Hoboken, NJ: John Wiley & Sons.

Riehemann, Susanne. 2001. *A Constructional Approach to Idioms and Word Formation*. PhD thesis, Stanford University.

Robins, R. H. 1967. *A Short History of Linguistics*. Bloomington: University of Indiana Press.

Rosenbaum, Peter. 1967. *The Grammar of English Predicate Complement Constructions*. Cambridge, MA: MIT Press.

Ross, John R. 1967. *Constraints on Variables in Syntax*. PhD thesis, MIT. Published as Infinite Syntax! Norwood, N.J.: Ablex, 1986.

Ross, John R. 1969. Auxiliaries as main verbs. In W. Todd (Ed.), *Studies in Philosophical Linguistics 1*. Evanston, Ill.: Great Expectations Press.

Ruwet, Nicolas. 1991. *Syntax and Human Experience*. Chicago: University of Chicago Press. Edited and translated by J. Goldsmith.

Sag, Ivan A. 1997. English relative clause constructions. *Journal of Linguistics* 33(2):431–484.

Sag, Ivan A. 2003. Coordination and underspecification. In *Proceedings of the 9th International Conference on Head-Driven Phrase Structure Grammar*. Available online at http://cslipublications.stanford.edu/HPSG/3/hpsg02-toc.html.

Sag, Ivan A. to appear. Rules and exceptions in the English auxiliary system. *Journal of Linguistics*.

Sag, Ivan A., Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for nlp. In A. Gelbukh (Ed.), *Proceedings of CICLING-2002*. Springer Verlag. Also appeared as LinGO Working Paper No. 2001-03. See http://lingo.stanford.edu/.

Sag, Ivan A., and Janet D. Fodor. 1994. Extraction without traces. In *Proceedings of the Thirteenth Annual Meeting of the West Coast Conference on Formal Linguistics*, Stanford. CSLI Publications.

Sag, Ivan A., and Carl Pollard. 1991. An integrated theory of complement control. *Language* 67:63–113.

Savitch, Walter J., Emmon Bach, William Marsh, and Gila Safran-Naveh. 1987. *The Formal Complexity of Natural Language*. Dordrecht: D. Reidel.

Schütze, Carson T. 1996. *The Empirical Base of Linguistics*. Chicago: University of Chicago Press.

Sells, Peter. 1985. *Lectures on Contemporary Syntactic Theories*. Stanford: CSLI Publications.

Sells, Peter (Ed.). 2001. *Formal and Empirical Issues in Optimality Theoretic Syntax*. Stanford: CSLI Publications.

Shieber, Stuart. 1986. *An Introduction to Unification-Based Approaches to Grammar*. Stanford: CSLI Publications.

Skinner, B. F. 1957. *Verbal Behavior*. New York: Appleton-Century-Crofts.

Smith, Jeffrey D. 1999. English number names in HPSG. In G. Webelhuth, J.-P. Koenig, and A. Kathol (Eds.), *Lexical and Constructional Aspects of Linguistic Explanation*, 145–160. Stanford: CSLI Publications.

Steedman, Mark. 1996. *Surface Structure and Interpretation*. Cambridge, MA: MIT Press.

Steedman, Mark. 2000. *The Syntactic Process*. Cambridge, MA: MIT Press/Bradford Books.

Steele, Susan. 1981. *An Encyclopedia of AUX*. Cambridge, MA: MIT Press.

Tabossi, Patrizia., Michael J. Spivey-Knowlton, Ken McRae, and Michael K. Tanenhaus. 1994. Semantic effects on syntactic ambiguity resolution: Evidence for a constraint-based resolution process. In C. Umilta and M. Moscovitch (Eds.), *Attention and Performance XV*. Hillsdale, N.J.: Erlbaum.

Tanenhaus, Michael K., Michael J. Spivey-Knowlton, Kathleen M. Eberhard, and July C. Sedivy. 1995. Integration of visual and linguistic information in spoken language comprehension. *Science* 268:1632–1634.

Tanenhaus, Michael K., Michael J. Spivey-Knowlton, Kathleen M. Eberhard, and July C. Sedivy. 1996. Using eye movements to study spoken language comprehension: Evidence for visually mediated incremental interpretation. In T. Inui and J. L. McClelland (Eds.), *Attention and Performance XVI: Information Integration in Perception and Communication*, 457–478. Cambridge, MA: MIT Press.

Tanenhaus, Michael K., and John C. Trueswell. 1995. Sentence comprehension. In J. Miller and P. Eimas (Eds.), *Handbook of Perception and Cognition*, Vol. 11. London: Academic Press.

Tesnière, Lucien. 1959. *Elements de Syntaxe Structurale*. Paris: C. Klincksieck.

Thomason, Richmond (Ed.). 1974. *Formal Philosophy: Selected Papers of Richard Montague*. New Haven: Yale University Press.

Tomasello, Michael. 1992. The social bases of language acquisition. *Social Development* 1:67–87.

Trask, Robert Lawrence. 1993. *A Dictionary of Grammatical Terms in Linguistics*. London and New York: Routledge.

Trask, Robert Lawrence. 2002. Review of Baker 2001. *The Human Nature Review* 2:77–81. Available online at http://human-nature.com/nibbs/02/trask.html.

Trueswell, John C., Michael K. Tanenhaus, and Susan M. Garnsey. 1992. Semantic influences on parsing: Use of thematic role information in syntactic disambiguation. *Journal of Memory and Language* 33.

Trueswell, John C., Michael K. Tanenhaus, and Christopher Kello. 1993. Verb-specific constraints in sentence processing: Separating effects of lexical preference from garden-paths. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 19:528–553.

Vijay-Shanker, K. 1987. *A Study of Tree-Adjoining Grammars.* PhD thesis, University of Pennsylvania.

Villavicencio, Aline, and Ann Copestake. 2002. On the nature of idioms. *LinGO Working Paper* No. 2002-04. Available online at: http://lingo.stanford.edu/.

Warner, Anthony. 1993. *English Auxiliaries: Structure and History.* Cambridge and New York: Cambridge University Press.

Warner, Anthony R. 2000. English auxiliaries without lexical rules. In R. Borsley (Ed.), *Syntax and Semantics Volume 32: The Nature and Function of Syntactic Categories*, 167–220, San Diego and London. Academic Press.

Wasow, Thomas. 1977. Transformations and the lexicon. In *Formal Syntax* (Culicover et al. 1977).

Wasow, Thomas. 1989. Grammatical theory. In *Foundations of Cognitive Science* (Posner 1989).

Webelhuth, Gert (Ed.). 1995. *Government and Binding Theory and the Minimalist Program.* Oxford: Basil Blackwell.

Weir, David. 1987. *Characterizing Mildly Context-Sensitive Grammar Formalisms.* PhD thesis, University of Pennsylvania.

Wilkes-Gibbs, Deanna. 1986. *Collaborative Proccesses of Language Use in Conversation.* PhD thesis, Stanford University.

Wood, Mary. 1993. *Categorial Grammars.* London and New York: Routledge.

XTAG, The Research Group. 2000. A lexicalized tree-adjoining grammar for English. Technical Report 98-18, Institute for Research in Cognitive Science, Philadelphia.

Zwicky, Arnold, and Geoffrey K. Pullum. 1983. Cliticiziation vs. inflection: English *n't*. *Language* 59:502–13.

Index

CSLI Lecture Notes
Number 152

The second edition of Syntactic Theory: A Formal Introduction expands and improves on a truly unique introductory syntax textbook. Like the first edition, it focuses on the development of precisely formulated grammars whose empirical predictions can be directly tested. There is considerable emphasis on prediction and evaluation of grammatical hypotheses, as well as on integrating syntactic hypotheses with matters of semantic analysis. The step-by-step introduction to a consistent grammar covering the core areas of English syntax is complemented by extensive problem sets drawing from a variety of languages. The student is introduced to the central topics in the syntax of English, induding: agreement, complementation, binding, coordination, imperatives, passives, control, raising, the auxiliary system, and the analysis of long distance dependency constructions. The theoretical perspective of the book is presented in the context of current models of language processing. The practical value of the constraint-based, lexicalist grammatical architecture proposed has already been demonstrated in computer language processing applications. The thoroughly reworked second edition includes revised and extended problem sets, updated analyses, additional examples, and more detailed exposition throughout.

Praise for the first edition of Syntactic Theory:

"A clear, thorough, and up-to-date introduction to a formal theory of grammar, it meticulously takes the reader through detailed analyses of many important aspects of English syntax, suggests applications to other languages, and evaluates the theoretical model employed from broad historical and intellectual perspectives." –Patrick Farrell, *Language* 77.

"*Syntactic Theory* sets a new standard for introductory syntax volumes that all future books should be measured against." –Gert Webelhuth, *Journal of Linguistics* 37.

"*Syntactic Theory* is, without a doubt, the best available introduction to unification-based syntactic theory." –Gregory Stump, University of Kentucky

"My undergraduate students loved this book." –Georgia Green, University of Illinois, Urbana-Champaign

Ivan A. Sag is Professor of Linguistics and Thomas Wasow is Professor of Lingusitics and Philosophy at Stanford University, where they also serve on the faculty of the Symbolic Systems Program. Emily M. Bender is Acting Assistant Professor in the Department of Linguistics at the University of Washington.

CSLI PUBLICATIONS
Center for the Study of Language and Information
Stanford, California

ISBN-13 ISBN 978-1-57586-400-6
ISBN-10 1-57586-400-2

9 781575 864006