March 1995

A Lexicalized Tree Adjoining Grammar for English

The XTAG Research Group
University of Pennsylvania

University of Pennsylvania Institute for Research in Cognitive Science Technical Report No. IRCS-95-03.

A Lexicalized Tree Adjoining Grammar for English

Abstract

This document describes a sizable grammar of English written in the TAG formalism and implemented for use with the XTAG system. This report and the grammar described herein supersedes the TAG grammar described in [Abeille` *et al.,* 1990]. The English grammar described in this report is based on the TAG formalism developed [Joshi *et al.,* 1975] which has been extended to include lexicalization ([Schabes *et al.*, 1988]), and unification-based feature structures ([Vijay Shanker and Joshi, 1991]). The grammar discussed in this report extends the grammar presented in [Abeille` *et al.,* in at least two ways. First, this grammar has more detailed linguistic analyses, and second, the grammar presented in this paper is fully implemented. The range of syntactic phenomena that can be handled is large and includes auxiliaries (including inversion), copula, raising and small clause constructions, topicalization, relative clauses, infinitives, gerunds, passives, adjuncts, it-clefts, wh-clefts, PRO contructions, noun-noun modifications, extraposition, determiner phrases, genitives, negation, noun-verb contractions, sentential adjuncts and imperatives. The XTAG grammar has been relatively stable since November 1993, although new analyses are still being added periodically.

Comments

University of Pennsylvania Institute for Research in Cognitive Science Technical Report No. IRCS-95-03.

The Institute For
Research In Cognitive
Science

A Lexicalized Tree Adjoining Grammar for English

by

The XTAG Research Group

University of Pennsylvania
3401 Walnut Street, Suite 400C
Philadelphia, PA 19104-6228

March 1995

Site of the NSF Science and Technology Center for
Research in Cognitive Science

University of Pennsylvania
Founded by Benjamin Franklin in 1740

IRCS Report 95-03

A Lexicalized Tree Adjoining Grammar for English

The XTAG Research Group

Institute for Research in Cognitive Science
University of Pennsylvania
3401 Walnut St., Suite 400C
Philadelphia, PA 19104-6228

xtag-request@linc.cis.upenn.edu

March 9, 1995

IRCS 95-03

Contents

List of Figures

Abstract

This document describes a sizable grammar of English written in the TAG formalism and implemented for use with the XTAG system. This report and the grammar described herein supersedes the TAG grammar described in [Abeillé *et al.*, 1990]. The English grammar described in this report is based on the TAG formalism developed in [Joshi *et al.*, 1975], which has been extended to include lexicalization ([Schabes *et al.*, 1988]), and unification-based feature structures ([Vijay-Shanker and Joshi, 1991]). The grammar discussed in this report extends the grammar presented in [Abeillé *et al.*, 1990] in at least two ways. First, this grammar has more detailed linguistic analyses, and second, the grammar presented in this paper is fully implemented. The range of syntactic phenomena that can be handled is large and includes auxiliaries (including inversion), copula, raising and small clause constructions, topicalization, relative clauses, infinitives, gerunds, passives, adjuncts, it-clefts, wh-clefts, PRO constructions, noun-noun modifications, extraposition, determiner phrases, genitives, negation, noun-verb contractions, sentential adjuncts and imperatives. The XTAG grammar has been relatively stable since November 1993, although new analyses are still being added periodically.

Acknowledgements

We are immensely grateful to Aravind Joshi for supporting this project.

The principal authors of the current document are: Tilman Becker, Christine Doran, Dania Egedi, Beth Ann Hockey, Seth Kulick and B. Srinivas. We are also grateful to Rajesh Bhatt for editorial help.

The following people have contributed to the development of grammars in the project: Anne Abeille, Rajesh Bhatt, Kathleen Bishop, Sharon Cote, Beatrice Daille, Christine Doran, Dania Egedi, Jason Frank, Caroline Heycock, Beth Ann Hockey, Daniel Karp, Seth Kulick, Young-Suk Lee, Patrick Martin, Megan Moser, Sabine Petillon, Yves Schabes, Victoria Tredinnick and Raffaella Zanuttini.

The XTAG system has been developed by: Tilman Becker, Richard Billington, Andrew Chalnick, Dania Egedi, Devtosh Khare, Albert Lee, David Magerman, Alex Mallet, Patrick Paroubek, Rich Pito, Gilles Prigent, Anoop Sarkar, Yves Schabes, B. Srinivas, Yuji Yoshiie and Martin Zaidel.

We would also like to thank Michael Hegarty, Lauri Karttunen, Anthony Kroch, Mitchell Marcus, Martha Palmer, Owen Rambow, Phillip Resnik, Beatrice Santorini and Mark Steedman.

In addition, Jeff Aaronson, Douglas DeCarlo, Mark-Jason Dominus, Mark Foster, Gaylord Holder, David Mageman, Ken Noble, Steven Shapiro and Ira Winston have provided technical support. Adminstrative support was provided by Carolyn Elken, Jodi Kerper, Christine Sandy and Trisha Yannuzzi.

This work was partially supported by ARO grant DAAL03-89-0031, ARPA grant N00014-90-J-1863, NSF STC grant DIR-8920230, and Ben Franklin Partnership Program (PA) grant 93S.3078C-6.

Part I

General Information

Chapter 1

Getting Around

This technical report presents the English XTAG grammar as implemented by the XTAG Research Group at the University of Pennsylvania. The technical report is organized into four parts, plus a set of appendices. Part 1 contains general information about the XTAG system and some of the underlying mechanisms that help shape the grammar. Chapter 2 contains an introduction to the formalism behind the grammar and parser, while Chapter 3 contains information about the entire XTAG system. Linguists interested solely in the grammar of the XTAG system may safely skip Chapters 2 and 3. Chapter 4 contains information on some of the linguistic principles that underlie the XTAG grammar, including the distinction between complements and adjuncts, and how case is handled.

The actual description of the grammar begins with Part 2, and is contained in the following three parts. Parts 2 and 3 contains information on the verb classes and the types of trees allowed within the verb classes, respectively, while Part 4 contains information on trees not included in the verb classes (e.g. NP's, PP's, various modifiers, etc). Chapter 5 of Part 2 contains a table that attempts to provide an overview of the verb classes and tree types by providing a graphical indication of which tree types are allowed in which verb classes. This has been cross-indexed to tree figures shown in the tech report. Chapter 6 contains an overview of all of the verb classes in the XTAG grammar. The rest of Part 2 contains more details on several of the more interesting verb classes, including ergatives, sentential subjects, sentential complements, small classes, ditransitives, and it-clefts.

Part 3 contains information on some of the tree types that are available within the verb classes. These tree types correspond to what would be transformations in a movement based approach. Not all of these types of trees are contained in all of the verb classes. The table (previously mentioned) in Part 2 contains a list of the tree types and indicates which verb classes each occurs in.

Part 4 focuses on the non-verb class trees in the grammar. NP's and determiners are presented in Chapter 18, while the various modifier trees are presented in Chapter 19. Auxiliary verbs, which are classed separate from the verb classes, are presented in Chapter 20, while certain types of conjunction are shown in Chapter 21. Other conjunctions, such as subordinating and discourse conjunction, are considered tree types, and as such are included in the chapter on adjunct clauses (section 15.2). Sentential complements of NP's and PP's are discussed in section 8.8.

Throughout the technical report, mention is occasionally made of changes or analyses that we hope to incorporate in the future. Appendix A details a list of these and other future work. The appendices also contain information on some of the nitty gritty details of the XTAG grammar, including the tree naming conventions (Appendix B), and a comprehensive list of the features used in the grammar (Appendix C). Appendix D contains an evaluation of the XTAG grammar, including comparisons with other wide coverage grammars.

Chapter 2

Feature-Based, Lexicalized Tree Adjoining Grammars

The English grammar described in this report is based on the TAG formalism ([Joshi *et al.*, 1975]), which has been extended to include lexicalization ([Schabes *et al.*, 1988]), and unification-based feature structures ([Vijay-Shanker and Joshi, 1991]). Tree Adjoining Languages (TALs) fall into the class of mildly context-sensitive languages, and as such are more powerful than context free languages. The TAG formalism in general, and lexicalized TAGs in particular, are well-suited for linguistic applications. As first shown by [Joshi, 1985] and [Kroch and Joshi, 1987], the properties of TAGs permit us to encapsulate diverse syntactic phenomena in a very natural way. For example, TAG's extended domain of locality and its factoring of recursion from local dependencies lead, among other things, to a localization of so-called unbounded dependencies.

2.1 TAG formalism

The primitive elements of the standard TAG formalism are known as elementary trees. ELEMENTARY TREES are of two types: initial trees and auxiliary trees (see Figure 2.1). In describing natural language, INITIAL TREES are minimal linguistic structures that contain no recursion, i.e. trees containing the phrasal structure of simple sentences, NP's, PP's, and so forth. Initial trees are characterized by the following: 1) all internal nodes are labeled by non-terminals, 2) all leaf nodes are labeled by terminals, or by non-terminal nodes marked for substitution. An initial tree is called an X-type initial tree if its root is labeled with type X.

Recursive structures are represented by AUXILIARY TREES, which represent constituents that are adjuncts to basic structures (e.g. adverbials). Auxiliary trees are characterized as follows: 1) all internal nodes are labeled by non-terminals, 2) all leaf nodes are labeled by terminals, or by non-terminal nodes marked for substitution, except for exactly one non-terminal node, called the foot node, which can only be used to adjoin the tree to another node[1], 3) the foot node has the same label as the root node of the tree.

[1] A null adjunction constraint (NA) is systematically put on the foot node of an auxiliary tree. This disallows adjunction of a tree onto the foot node itself.

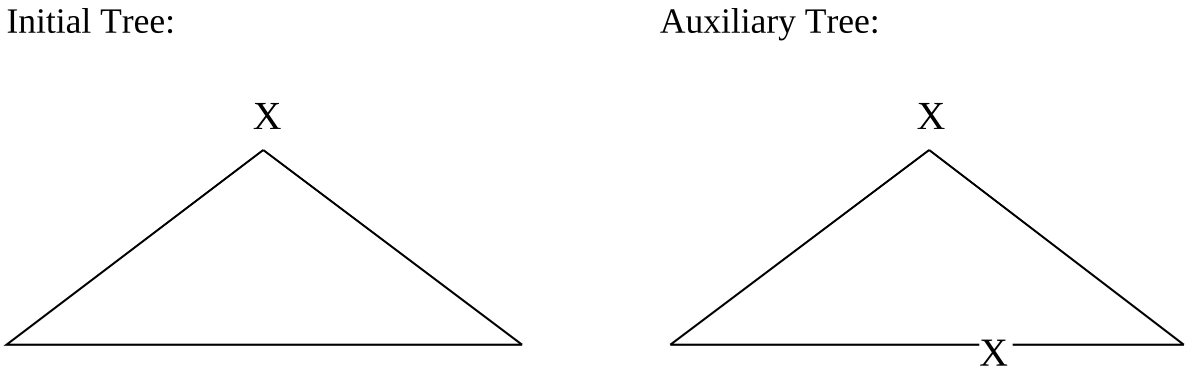


Figure 2.1: Elementary trees in TAG

There are two operations defined in the TAG formalism, substitution[2] and adjunction. In the substitution operation, the root node on an initial tree is merged into a non-terminal leaf node marked for substitution in another initial tree, producing a new tree. The root node and the substitution node must have the same name. Figure 2.2 shows two initial trees and the tree resulting from the substitution of one tree into the other.

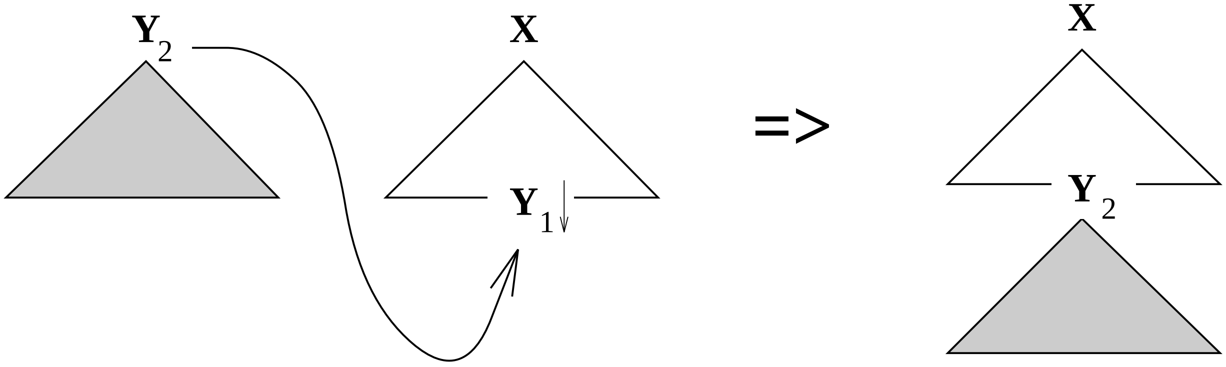


Figure 2.2: Substitution in TAG

In an adjunction operation, an auxiliary tree is grafted onto a non-terminal node anywhere in an initial tree. The root and foot nodes of the auxiliary tree must match the node at which the auxiliary tree adjoins. Figure 2.3 shows an auxiliary tree and an initial tree, and the tree resulting from an adjunction operation.

A TAG G is a collection of finite initial trees, I, and auxiliary trees, A. The tree set of a TAG G, $\mathcal{T}(G)$ is defined to be the set of all derived trees starting from S-type initial trees in I whose frontier consists of terminal nodes (all substitution nodes having been filled). The string language generated by a TAG, $\mathcal{L}(G)$, is defined to be the set of all terminal strings on the frontier of the trees in $\mathcal{T}(G)$.

[2] Technically, substitution is a specialized version of adjunction, but it is useful to make a distinction between the two.

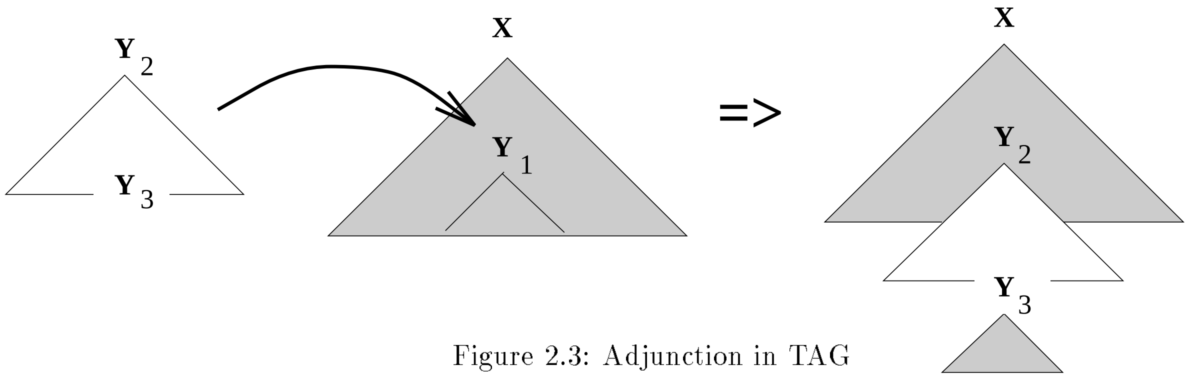


Figure 2.3: Adjunction in TAG

2.2 Lexicalization

'Lexicalized' grammars systematically associate each elementary structure with a lexical anchor. This means that in each structure there is a lexical item that is realized. It does not mean simply adding feature structures (such as head) and unification equations to the rules of the formalism. These resultant elementary structures specify extended domains of locality (as compared to CFGs) over which constraints can be stated.

Following [Schabes *et al.*, 1988] we say that a grammar is LEXICALIZED if it consists of 1) a finite set of structures each associated with a lexical item, and 2) an operation or operations for composing the structures. Each lexical item will be called the ANCHOR of the corresponding structure, which defines the domain of locality over which constraints are specified. Note then, that constraints are local with respect to their anchor.

Not every grammar is in a lexicalized form.[3] In the process of lexicalizing a grammar, the lexicalized grammar is required to be strongly equivalent to the original grammar, i.e. it must produce not only the same language, but the same structures or tree set as well.

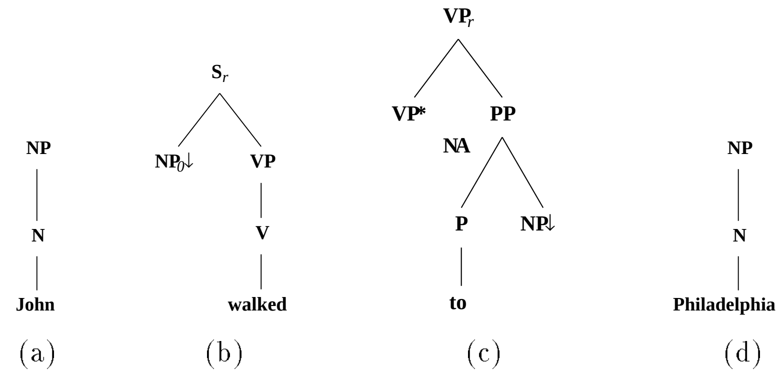


Figure 2.4: Lexicalized Elementary trees

[3] Notice the similarity of the definition of a lexicalized grammar with the off line parsability constraint ([Kaplan and Bresnan, 1983]). As consequences of our definition, each structure has at least one lexical item (its anchor) attached to it and all sentences are finitely ambiguous.

In Figure 2.4, which shows sample initial and auxiliary trees, substitution sites are marked by a ↓, and foot nodes are marked by an $*$. This notation is standard and is followed in the rest of this report.

2.3 Unification-based features

In a unification framework, a feature structure is associated with each node in an elementary tree. This feature structure contains information about how the node interacts with other nodes in the tree. It consists of a top part, which generally contains information relating to the supernode, and a bottom part, which generally contains information relating to the subnode. Substitution nodes, however, have only the top features, since the tree substituting in logically carries the bottom features.

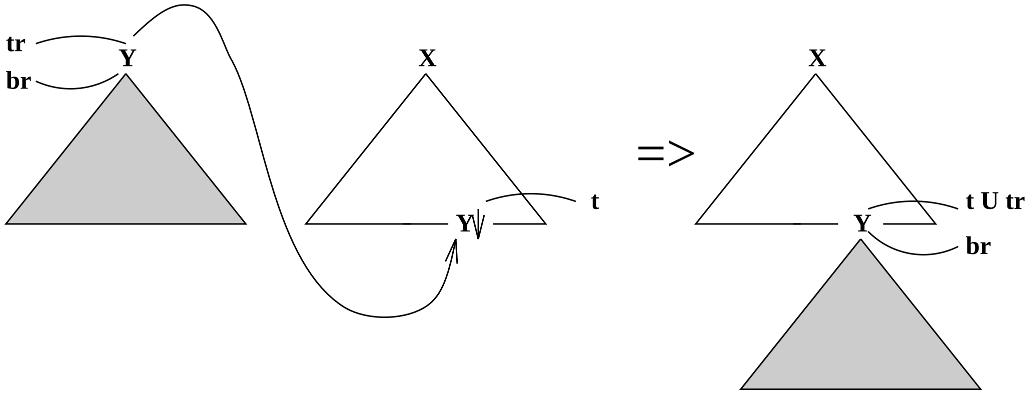


Figure 2.5: Substitution in FB-LTAG

The notions of substitution and adjunction must be augmented to fit within this new framework. The feature structure of a new node created by substitution inherits the union of the features of the original nodes. The top feature of the new node is the union of the top features of the two original nodes, while the bottom feature of the new node is simply the bottom feature of the top node of the substituting tree (since the substitution node has no bottom feature). Figure 2.5[4] shows this more clearly.

Adjunction is only slightly more complicated. The node being adjoined into splits, and its top feature unifies with the top feature of the root adjoining node, while its bottom feature unifies with the bottom feature of the foot adjoining node. Again, this is easier shown graphically, as in Figure 2.6[5].

The embedding of the TAG formalism in a unification framework allows us to dynamically specify local constraints that would have otherwise had to have been made statically within the trees. Constraints that verbs make on their complements, for instance, can be implemented through the feature structures. The notions of Obligatory and Selective Adjunction, crucial

[4] abbreviations in the figure: t=top feature structure, tr=top feature structure of the root, br=bottom feature structure of the root, U=unification

[5] abbreviations in the figure: t=top feature structure, b=bottom feature structure, tr=top feature structure of the root, br=bottom feature structure of the root, tf=top feature structure of the foot, bf=bottom feature structure of the foot, U=unification

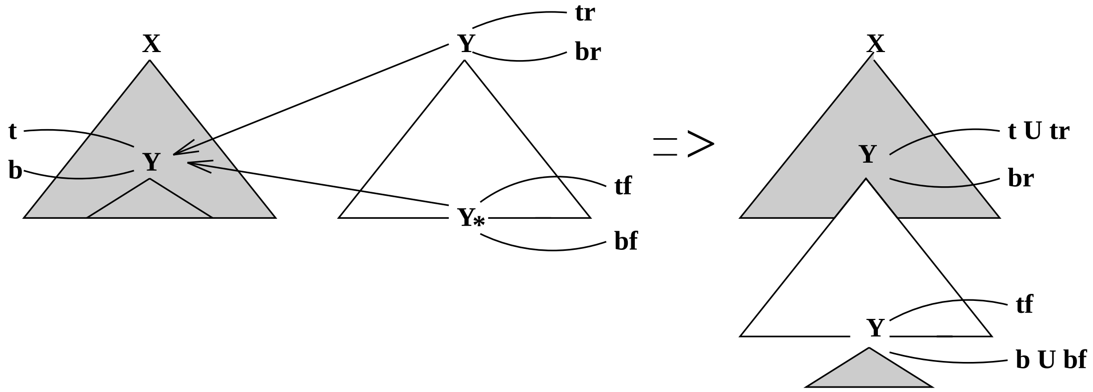
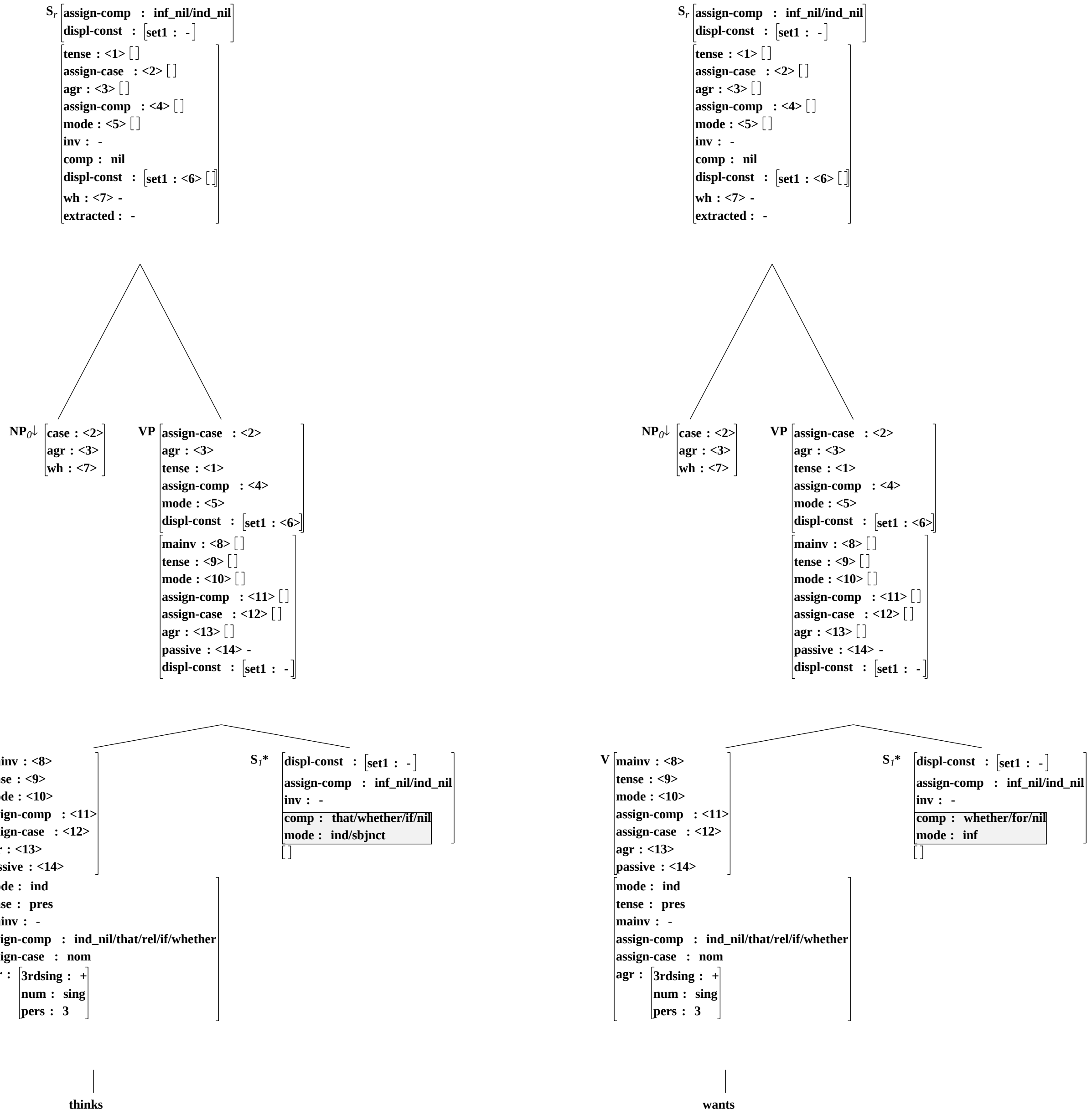


Figure 2.6: Adjunction in FB-LTAG

to the formation of lexicalized grammars, can also be handled through the use of features.[6] Perhaps more important to developing a grammar, though, is that the trees can serve as a schemata to be instantiated with lexical-specific features when an anchor is associated with the tree. To illustrate this, Figure 2.7 shows the same tree lexicalized with two different verbs, each of which instantiates the features of the tree according to its lexical selectional restrictions.

In Figure 2.7, the lexical item *thinks* takes an indicative sentential complement, as in the sentence *John thinks that Mary loves Sally. Want* takes a sentential complement as well, but an infinitive one, as in *John wants to love Mary.* This distinction is easily captured in the features and passed to other nodes to constrain which trees this tree can adjoin into, both cutting down the number of separate trees needed and enforcing conceptual Selective Adjunctions (SA).

[6] The remaining constraint, Null Adjunction (NA), must still be specified directly on a node.



think tree

want tree

Figure 2.7: Lexicalized Elementary Trees with Features

Chapter 3

Overview of the XTAG System

This section is derived in large part from the XTAG project notes ([Doran *et al.*, 1994]). An additional section on Corpus Parsing and Evaluation has not been replicated here (but see Appendix D). This section focuses on the various components that comprise the parser and English grammar in the XTAG system. Persons interested only in the linguistic analyses in the grammar may skip this section without loss of continuity, although we may occasionally refer back to the various components mentioned here.

3.1 System Description

Figure 3.1 shows the overall flow of the system when parsing a sentence. The input sentence is submitted to the **Morphological Analyzer** and the **Tagger**. The morphological analyzer retrieves the morphological information for each individual word in the sentence from the **Morphological Database**. The result is filtered in the **P.O.S. Blender** using the output of the **Trigram Tagger** to reduce the part of speech ambiguity of the words. The augmented sentence, with each word annotated with part of speech tags and morphological information, is input to the **Parser**, which then consults the **Syntactic Database** and the **Tree Database** to retrieve the appropriate tree structures for each word in the sentence. Information from the **Statistical Database**, along with a variety of heuristics, is used to reduce the number of trees selected. The parser then composes the structures to obtain the parse(s) of the sentence.

3.2 Morphological Analyzer

The morphology data was originally extracted from the Collins English Dictionary ([Hanks, 1979]) and Oxford Advanced Learner's Dictionary ([Hornby, 1974]) available through ACL-DCI ([Liberman, 1989]), and then cleaned up and augmented by hand ([Karp *et al.*, 1992]). The database consists of approximately 317,000 inflected items, along with their root forms and inflectional information (such as case, number, tense). Thirteen different parts of speech are differentiated: Noun, Proper Noun, Pronoun, Verb, Verb Particle, Adverb, Adjective, Preposition, Complementizer, Determiner, Conjunction, Interjection, and Noun/Verb Contraction. Nouns and Verbs are the largest categories, with approximately 213,000 and 46,500 inflected forms respectively. This information is maintained in database form for quick access. Retrieval

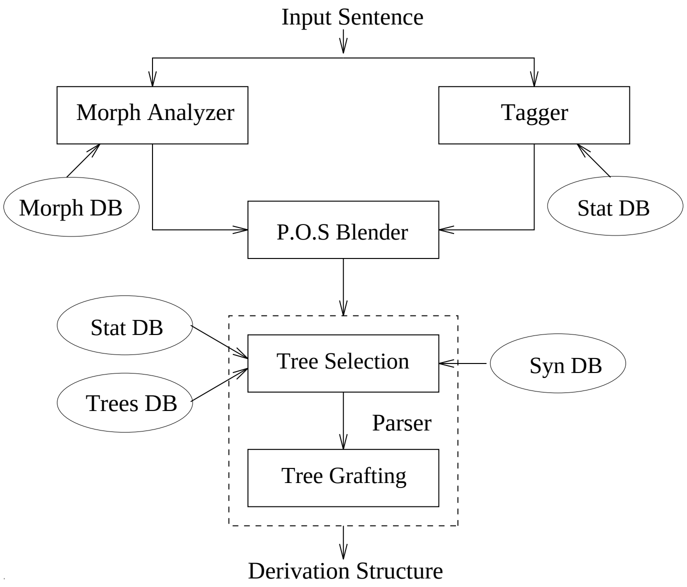


Figure 3.1: Overview of the XTAG system

time for a given inflected entry is approximately 0.6 msec.

3.3 Part of Speech Tagger

A trigram part of speech tagger ([Church, 1988]), trained on the Wall Street Journal Corpus, is incorporated in XTAG. The trigram tagger has been extended to output the N-best parts of speech sequences ([Soong and Huang, 1990]). XTAG uses this information to reduce the number of specious parses by filtering the possible parts of speech provided by the morphological analyzer for each word. When the correct part of speech sequence is returned, the time required to parse a sentence decreases by an average of 93%.

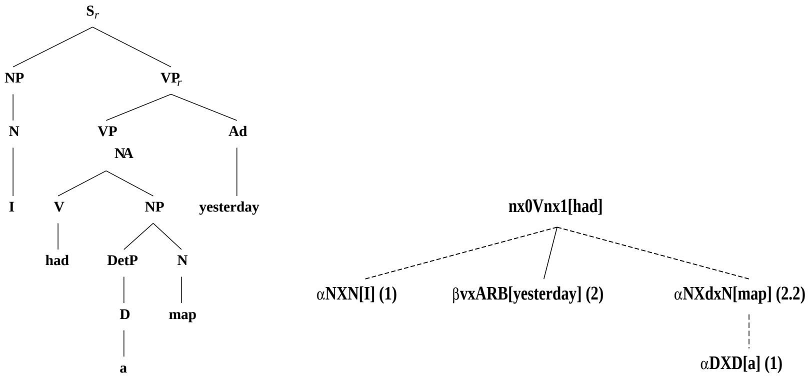
3.4 Parser

XTAG uses an Earley-style parser which has been extended to handle feature structures associated with trees ([Schabes, 1990]). The parser uses a general two-pass parsing strategy for lexicalized grammars ([Schabes *et al.*, 1988]). In the tree-selection pass, the parser uses the syntactic database entry for each lexical item in the sentence to select a set of elementary structures from the tree database. The tree-grafting pass composes the selected trees using substitution and adjunction operations to obtain the parse of the sentence. The output of the parser for the sentence *I had a map yesterday* is illustrated in Figure 3.2. The parse tree[1] represents the surface constituent structure, while the derivation tree represents the derivation history of the parse. The nodes of the derivation tree are the tree names anchored by the lexical items[2]. The composition operation is indicated by the nature of the arcs: a dashed line is used for

[1] The feature structures associated with each note of the parse tree are not shown here.

[2] Appendix B explains the conventions used in naming the trees.

substitution and a bold line for adjunction. The number beside each tree name is the address of the node at which the operation took place. The derivation tree can also be interpreted as a dependency graph with unlabeled arcs between words of the sentence.



Parse Tree

Derivation Tree

Figure 3.2: Output Structures from the Parser

Additional methods that take advantage of FB-LTAGs have been implemented to improve the performance of the parser. For instance, the span[3] of the tree and the position of the anchor in the tree are used to weed out unsuitable trees in the first pass of the parser. Statistical information about the usage frequency of the trees has been acquired by parsing corpora. This information has been compiled into a statistical database that is used by the parser. These methods speed the runtime by approximately 87%.

3.5 Syntactic Database

The syntactic database associates lexical items with the appropriate trees and tree families based on various selectional information. The syntactic database entries were originally extracted from the Oxford Advanced Learner's Dictionary ([Hornby, 1974]) and Oxford Dictionary for Contemporary Idiomatic English ([Cowie and Mackin, 1975]) available through ACL-DCI ([Liberman, 1989]), and then modified and augmented by hand ([Egedi and Martin, 1994]). There are more than 37,000 syntactic database entries. Selected entries from this database are shown in Table 3.1.

Each syntactic entry consists of an INDEX field, the uninflected form under which the entry is compiled in the database, an ENTRY field, which contains all of the lexical items that will anchor the associated tree(s), a POS field, which gives the part of speech for the lexical item(s) in the ENTRY field, and then either (but not both) a TREES or FAM field. The TREES field indicates a list of individual trees to be associated with the entry, while the FAM field indicates a list of tree families. A tree family may contain a number of trees. A syntactic entry may also contain a list of feature templates (FS) which expand out to feature equations to be placed in the specified tree(s). Any number of EX fields may be provided for example sentences. Note

[3] The SPAN of a tree is the number of terminals and non-terminals along its frontier.

that lexical items may have more than one entry and may select the same tree more than once, using different features to capture lexical idiosyncrasies.

INDEX:	have/26
ENTRY:	have
POS:	V
TREES:	βVvx
FS:	#VPr_ind, #VPr_past, #VPr_perfect+, #VP_ppart, #VP_pass-
EX:	he had died; we had died
INDEX:	have/50
ENTRY:	have
POS:	V
TREES:	βVvx
FS:	#VP_inf
EX:	John has to go to the store.
INDEX:	have/69
ENTRY:	NP0 have NP1
POS:	NP0 V NP1
FAM:	Tnx0Vnx1
FS:	#TRANS+
EX:	John has a problem.
INDEX:	map/1
ENTRY:	NP0 map out NP1
POS:	NP0 V PL NP1
FAM:	Tnx0Vplnx1
INDEX:	map/3
ENTRY:	map
POS:	N
TREES:	αN, αNXdxN, βNn
FS:	#N_wh-, #N_refl-
INDEX:	map/4
ENTRY:	map
POS:	N
TREES:	αNXN
FS:	#N_wh-, #N_refl-, #N_plur

Table 3.1: Example Syntactic Database Entries

The syntactic database is currently undergoing a series of changes designed to make it easier to use and update. In addition, the number of entries will be augmented to increase the coverage of the database, and the defaults used by the XTAG system will be accessible from the database itself. The format of the entries as seen in Table 3.1 will change slightly in the new version, but the same basic information will be included.

3.6 Tree Database

Trees in the English XTAG grammar fall into two conceptual classes. The smaller class consists of individual trees such as NP and adverb trees. The trees in this class are generally anchored by non-verbal lexical items. The larger class consists of trees that are grouped into tree families. These tree families represent subcategorization frames (see section 4.1). As of the end of 1994, there are 569 trees that compose 38 tree families, along with 67 individually selected trees in the tree database.

3.7 Statistics Database

The statistics database contains tree unigram frequencies which have been collected by parsing the Wall Street Journal, IBM manual, and ATIS corpora using the XTAG English grammar. The parser, augmented with the statistics database, assigns each word of the input sentence the top three most frequently used trees given the part of speech of the word. On failure the parser retries using all the trees suggested by the syntactic database for each word. The augmented parser has been observed to have a success rate of 50% without retries.

3.8 X-Interface

In addition to the parser and English grammar, XTAG provides a graphical interface for manipulating TAGs. The interface offers the following:

- Menu-based facility for creating and modifying tree files and loading grammar files.
- User controlled parser parameters, including the parsing of categories (S, embedded S, NP, DetP), and the use of the tagger (on/off/retry on failure).
- Storage/retrieval facilities for elementary and parsed trees as text files.
- The production of postscript files corresponding to elementary and parsed trees.
- Graphical displays of tree and feature data structures, including a scroll 'web' for large tree structures.
- Mouse-based tree editor for creating and modifying trees and feature structures.
- Hand combination of trees by adjunction or substitution for use in diagnosing grammar problems.

3.9 Computer Platform

XTAG was developed on the Sun SPARC station series, and has been tested on the Sun 4 and HP BOBCATs series 9000. It is available through anonymous ftp, and requires 20MB of disk space. Please send mail to xtag-request@linc.cis.upenn.edu for ftp instructions or more information. XTAG requires the following software to run:

- A machine running UNIX and X11R4 (or higher). Previous releases of X will not work. X11R4 is free software available from MIT.
- A Common Lisp compiler which supports the latest definition of Common Lisp (Steele's Common Lisp, second edition). XTAG has been tested with Lucid Common Lisp 4.0 and Allegro 4.0.1.
- CLX version 4 or higher. CLX is the lisp equivalent to the Xlib package written in C.
- Mark Kantrowitz's Lisp Utilities from CMU: logical-pathnames and defsystem.

The latest version of CLX (R5.0) and the CMU Lisp Utilities are provided in our ftp directory for your convenience. However, we ask that you refer to the appropriate source for updates.

The morphology database component ([Karp *et al.*, 1992]), no longer under licensing restrictions, is available as a separate system from the XTAG system. FTP instructions and more information can be obtained by mailing requests to lex-request@linc.cis.upenn.edu.

The syntactic database component is also available as a separate system ([Egedi and Martin, 1994]). The new format of the database is expected to be available in 1995. FTP instructions and more information can be obtained by mailing requests to lex-request@linc.cis.upenn.edu.

Chapter 4

Underview

The morphology, syntactic, and tree databases together comprise the English grammar. A lexical item that is not in the databases receives a default tree selection and features for its part of speech and morphology. In designing the grammar, a decision was made early on to err on the side of acceptance whenever there were conflicting opinions as to whether or not a construction is grammatical. In this sense, the XTAG English grammar functions better as an acceptor rather than a generator of English sentences. The range of syntactic phenomena that can be handled is large and includes auxiliaries (including inversion), copula, raising and small clause constructions, topicalization, relative clauses, infinitives, gerunds, passives, adjuncts, it-clefts, wh-clefts, PRO constructions, noun-noun modifications, extraposition, determiner phrases, genitives, negation, noun-verb contractions, sentential adjuncts and imperatives. The combination of large scale lexicons and wide phenomena coverage result in a robust system.

4.1 Subcategorization Frames

Elementary trees for non-auxiliary verbs are used to represent the linguistic notion of subcategorization frames. The anchor of the elementary tree subcategorizes for the other elements that appear in the tree, forming a clausal or sentential structure. Tree families group together trees belonging to the same subcategorization frame. Consider the following uses of the verb *buy*:

(1) Srini bought a book.

(2) Srini bought Beth a book.

In sentence (1), the verb *buy* subcategorizes for a direct object NP. The elementary tree anchored by *buy* is shown in Figure 4.1(a) and includes nodes for the NP complement of *buy* and for the NP subject. In addition to this declarative tree structure, the tree family also contains the trees that would be related to each other transformationally in a movement based approach, i.e passivization, imperatives, wh-questions, relative clauses, and so forth. Sentence (2) shows that *buy* also subcategorizes for a double NP object. This means that *buy* also selects the double NP object subcategorization frame, or tree family, with its own set of transformationally related sentence structures. Figure 4.1(b) shows the declarative structure for this set of sentence structures.

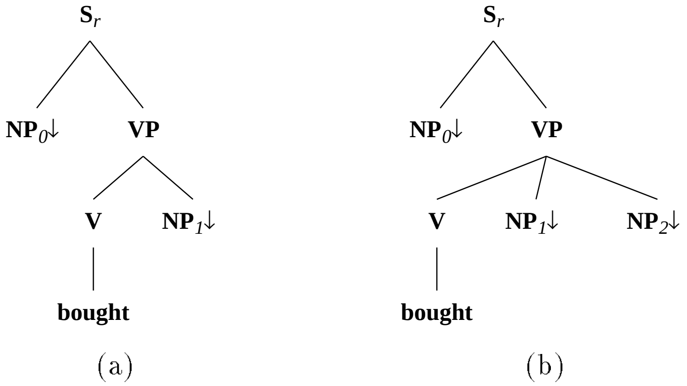


Figure 4.1: Different subcategorization frames for the verb *buy*

4.2 Complements and Adjuncts

Complements and adjuncts have very different structures in the XTAG grammar. Complements are included in the elementary tree anchored by the verb that selects them, while adjuncts do not originate in the same elementary tree as the verb anchoring the sentence, but are instead added to a structure by adjunction. The contrasts between complements and adjuncts have been extensively discussed in the linguistics literature and the classification of a given element as one or the other remains a matter of debate (see [Rizzi, 1990], [Larson, 1988], [Jackendoff, 1990], [Larson, 1990], [Cinque, 1990], [Obernauer, 1984], [Lasnik and Saito, 1984], and [Chomsky, 1986]). The guiding rule used in developing the XTAG grammar is whether or not the sentence is ungrammatical without the questioned structure.[1] Consider the following sentences:

(3) Srini bought a book.

(4) Srini bought a book at the bookstore.

(5) Srini arranged for a ride.

(6) ∗Srini arranged.

Prepositional phrases frequently occur as adjuncts, and when they are used as adjuncts they have the tree structure shown in Figure 4.2(a). This adjunction tree would adjoin into the tree shown in Figure 4.1(a) to generate sentence (4). There are verbs, however, such as *arrange*, *hunger* and *differentiate*, that take prepositional phrases as complements. Sentences (5) and (6) clearly show that the prepositional phrase are not optional for these verbs. For these sentences, the prepositional phrase will be an initial tree (as shown in Figure 4.2(b)) that substitutes into an elementary tree, such as the one anchored by the verb *arrange* in Figure 4.2(c).

Virtually all parts of speech, except for main verbs, function as both complements and adjuncts in the grammar. More information is available in this report on various parts of speech as complements: adjectives (e.g. section 6.13), nouns (e.g. section 6.2), and prepositions (e.g. section 6.11); and as adjuncts: adjectives (section 19.1), adverbs (section 19.4), nouns (section 19.2), and prepositions (section 19.3).

[1] Iteration of a structure can also be used as a diagnostic: *Srini bought a book at the bookstore on Walnut Street for a friend.*

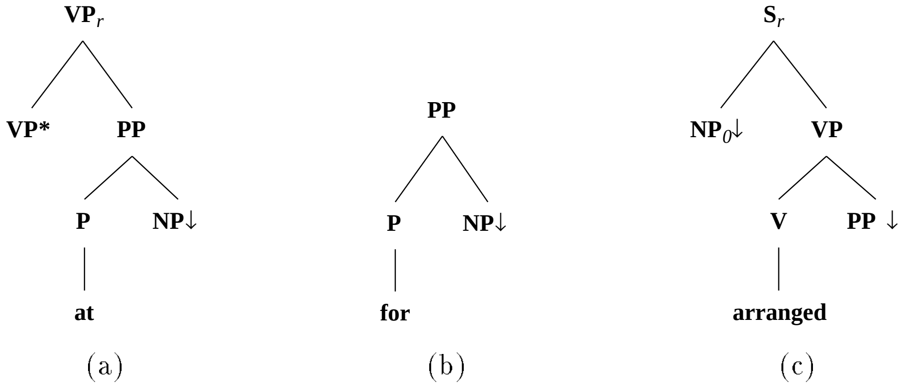


Figure 4.2: Trees illustrating the difference between Complements and Adjuncts

4.3 Non-S constituents

Although sentential trees are generally considered to be special cases in any grammar, insofar as they make up a 'starting category', it is the case that any initial tree constitutes a phrasal constituent. These initial trees may have substitution nodes that need to be filled (by other initial trees), and may be modified by adjunct trees, exactly as the trees rooted in S. Although grouping is possible according to the heads or anchors of these trees, we have not found any classification similar to the subcategorization frames for verbs that can be used by a lexical entry to 'group select' a set of trees. These trees are selected one by one by each lexical item, according to each lexical item's idiosyncrasies. The grammar described by this technical report places them into several files for ease of use, but these files do not constitute tree families in the way that the subcategorization frames do.

4.4 Case Assignment

4.4.1 Approaches to Case

4.4.1.1 Case in GB theory

GB (Government and Binding) theory proposes the following 'case filter' as a requirement on S-structure.[2]

> CASE FILTER Every overt NP must be assigned abstract case. [Haegeman, 1991]

Abstract case is taken to be universal. Languages with rich morphological case marking, such as Latin, and languages with very limited morphological case marking, like English, are all presumed to have full systems of abstract case that differ only in the extent of morphological realization.

In GB, abstract case is assigned to NP's by various case assigners, namely verbs, prepositions, and INFL. Verbs and prepositions are said to assign accusative case to NP's that they

[2] There are certain problems with applying the case filter as a requirement at the level of S-structure. These issues are not crucial to the discussion of the English XTAG implementation of case and so will not be discussed here. Interested readers are referred to [Lasnik and Uriagereka, 1988].

govern, and INFL assigns nominative case to NP's that it governs. These governing categories are constrained in where they can assign case by means of 'barriers' based on 'minimality conditions', although these are relaxed in 'exceptional case marking' situations. The details of the GB analysis are beyond the scope of this technical report, but see [Chomsky, 1986] for the original analysis or [Haegeman, 1991] for an overview. Let it suffice for us to say that the notion of abstract case and the case filter are useful in accounting for a number of phenomena including the distribution of nominative and accusative case, and the distribution of overt NP's and empty categories (such as PRO).

4.4.1.2 Minimalism and Case

A major conceptual difference between GB theories and Minimalism is that in Minimalism, lexical items carry their features with them rather than being assigned their features based on the nodes that they end up at. For nouns, this means that they carry case with them, and that their case is 'checked' when they are in SPEC position of AGR$_s$ or AGR$_o$, which subsequently disappears ([Chomsky, 1992]).

4.4.2 Case in XTAG

The English XTAG grammar adopts the notion of case and the case filter for many of the same reasons argued in the GB literature. However, in some respects the English XTAG grammar's implementation of case more closely resembles the treatment in Chomsky's Minimalism framework ([Chomsky, 1992]) than the system outlined in the GB literature ([Chomsky, 1986]). As in Minimalism, nouns in the XTAG grammar carry case with them, which is eventually 'checked'. However in the XTAG grammar, noun cases are checked against the case values assigned by the verb during the unification of the feature structures. Unlike Chomsky's Minimalism, there are no separate AGR nodes; the case checking comes from the verbs directly. Case assignment from the verb is more like the GB approach than the requirement of a SPEC-head relationship in Minimalism.

Most nouns in English do not have separate forms for nominative and accusative case, and so they are ambiguous between the two. Pronouns, of course, are morphologically marked for case, and each carries the appropriate case in its feature. Figures 4.3(a) and 4.3(b) show the NP tree anchored by a noun and a pronoun, respectively, along with the feature values associated with each word. Note that *books* simply gets the default case **nom/acc**, while *she* restricts the case to be **nom**.

4.4.3 Case Assigners

4.4.3.1 Prepositions

Case is assigned in the XTAG English grammar by two components - verbs and prepositions.[3] Prepositions assign accusative case (**acc**) through their <**assign-case**> feature, which is linked directly to the <**case**> feature of their objects. Figure 4.4(a) shows a lexicalized preposition tree, while Figure 4.4(b) shows the same tree with the NP tree from Figure 4.3(a) substituted

[3] *For* also assigns case as a complementizer. See section 8.5 for more details.

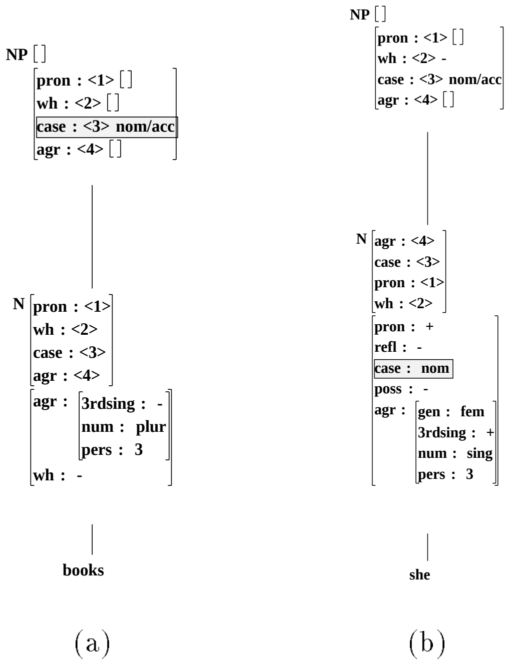


Figure 4.3: Lexicalized NP trees with case markings

into the NP position. Figure 4.4(c) is the tree in Figure 4.4(b) after unification has taken place. Note that the case ambiguity of *books* has been resolved to accusative case.

4.4.3.2 Verbs

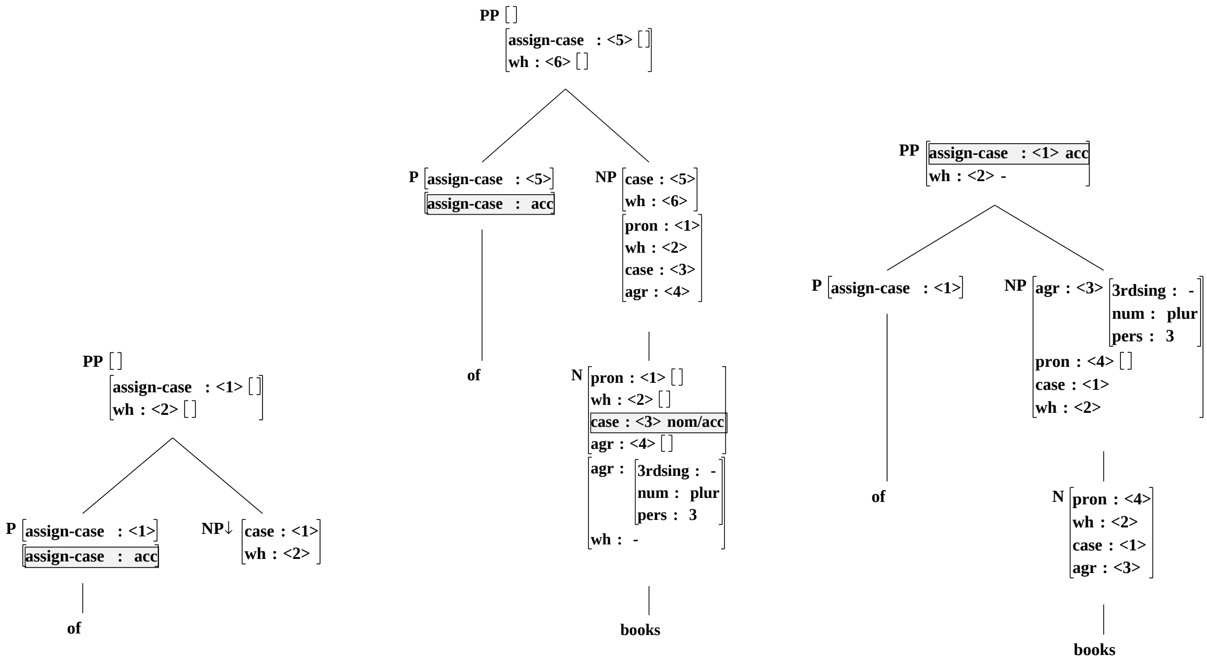
Verbs are the other part of speech in the XTAG grammar that can assign case. Because XTAG does not distinguish INFL and VP nodes, verbs must provide case assignment on the subject position in addition to the case assigned to their NP complements.

Assigning case to NP complements is handled by building the case values of the complements directly into the tree that the case assigner (the verb) anchors. Figures 4.5(a) and 4.5(b) show an S tree[4] that would be anchored[5] by a transitive and ditransitive verb, respectively. Note that the case assignments for the NP complements are already in the tree, even though there is not yet a lexical item anchoring the tree. Since every verb that selects these trees (and other trees in each respective subcategorization frame) assigns the same case to the complements, building case features into the tree has exactly the same result as putting the case feature value in each verb's lexical entry.

The case assigned to the subject position varies with verb form. Since the XTAG grammar treats the inflected verb as a single unit rather than dividing it into INFL and V nodes, case, along with tense and agreement, is expressed in the features of verbs, and must be passed in the appropriate manner. The trees in Figure 4.6 show the path of linkages that joins the <**assign-case**> feature of the V to the <**case**> feature of the subject NP. The morphological form of

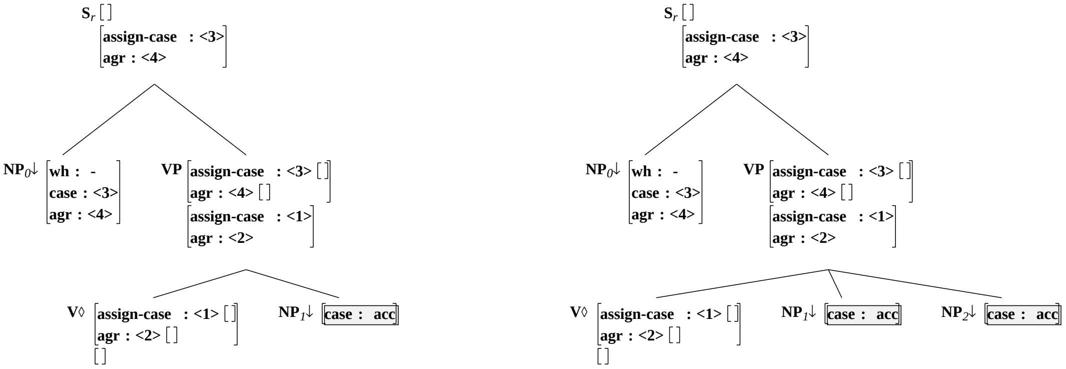
[4] Features not pertaining to this discussion have been taken out to improve readability and to make the trees easier to fit onto the page.

[5] The diamond marker (⋄) indicates the anchor(s) of a structure if the tree has not yet been lexicalized.



(a) (b) (c)

Figure 4.4: Assigning case in prepositional phrases



(a) (b)

Figure 4.5: Case assignment to NP arguments

the verb determines the value of the <**assign-case**> feature. Figures 4.6(a) and 4.6(b) show the same tree[6] anchored by different morphological forms of the verb *sing*, which give different values for the <**assign-case**> feature.

The adjunction of an auxiliary verb onto the VP node breaks the <**assign-case**> link from the main V, replacing it with a link from the auxiliary verb instead.[7] The progressive form of the verb in Figure 4.6(b) has the feature-value <**assign-case**>=**none**, but this is overridden

[6] Again, the feature structures shown have been restricted to those that pertain to the V/NP interaction.

[7] See section 20.1 for a more complete explanation of how this relinking occurs.

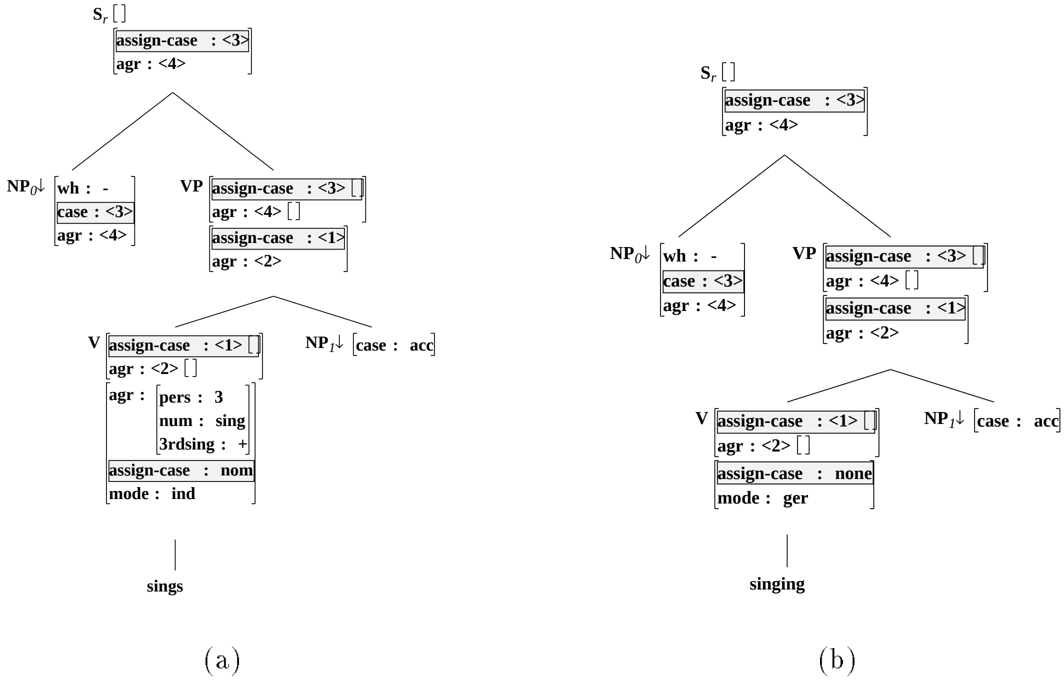


Figure 4.6: Assigning case according to verb form

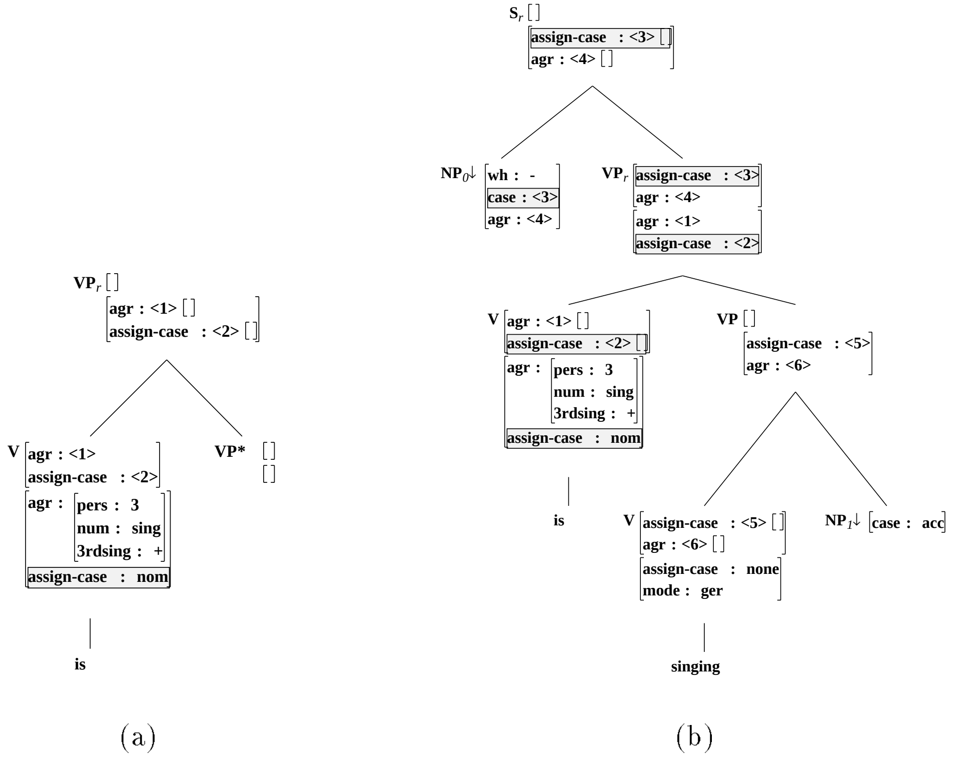


Figure 4.7: Proper case assignment with auxiliary verbs

by the adjunction of the appropriate form of the auxiliary word *be*. Figure 4.7(a) shows the lexicalized auxiliary tree, while Figure 4.7(b) shows it adjoined into the transitive tree shown in Figure 4.6(b). The case value passed to the subject NP is now **nom** (nominative).

4.4.4 PRO in a unification based framework

Most forms of a verb assign nominative case, although some forms, such as past participle, assign no case whatsoever. This is different than assigning case **none**, as the progressive form of the verb *sing* does in Figure 4.6(b). The distinction of a case **none** from no case is indicative of a divergence from the standard GB theory. In GB theory, the absence of case on an NP means that only PRO can fill that NP. With feature unification as is used in the FB-LTAG grammar, the absence of case on an NP means that *any* NP can fill it, regardless of its case. This is due to the mechanism of unification, in which if something is unspecified, it can unify with anything. Thus we have a specific case **none** to handle verb forms that in GB theory do not assign case. PRO is the only NP with case **none**. Verb forms that assign no case, as the past participle mentioned above, can do so because they cannot occur without an auxiliary verb which takes care of the case assignment. Note that although we are drawn to this treatment by our use of unification for feature manipulation, our treatment is very similar to the assignment of null case to PRO in [Chomsky and Lasnik, 1993]. [Watanabe, 1993] also proposes a very similar approach within Chomsky's Minimalist framework.[8]

[8] See section 8.1 for additional discussion of PRO.

Part II

Verb Classes

Chapter 5

Where to Find What

The two page table that follows gives an overview of what types of trees occur in various tree families with pointers to discussion in this report. An entry in a cell of the table indicates that the tree(s) for the construction named in the row header are included in the tree family named in the column header. Entries are of two types. If the particular tree(s) are displayed and/or discussed in this report the entry gives a page number reference to the relevant discussion or figure.[1] Otherwise, a $\surd$ indicates inclusion in the tree family but no figure or discussion related specifically to that tree in this report. Blank cells indicate that there are no trees for the construction named in the row header in the tree family named in the column header. The table below gives the expansion of abbreviations in the table headers.

[1] Since Chapter 6 has a brief discussion and a declarative tree for every tree family, page references are given only for other sections in which discussion or tree diagrams appear.

Abbreviation	Full Name
Sentential Comp. with NP	Sentential Complement with NP
Ditrans. Light Verbs w. PP Shift	Ditransitive Light Verbs with PP Shift
Ditrans. Light Verbs w/o PP Shift	Ditransitive Light Verbs without PP Shift
Adj. Sm. Cl. w. Sentential Subj.	Adjective Small Clause with Sentential Subject
NP Sm. Clause w. Sentential Subj.	NP Small Clause with Sentential Subject
PP Sm. Clause w. Sentential Subj.	PP Small Clause with Sentential Subject
Y/N question	Yes/No question
Wh-mov. NP complement	Wh-moved NP complement
Wh-mov. S comp.	Wh-moved S complement
Wh-mov. Adj comp.	Wh-moved Adjective complement
Wh-mov. object of a P	Wh-moved object of a P
Wh-mov. PP	Wh-moved PP
Topic. NP complement	Topicalized NP complement
Det. gerund	Determiner gerund
Rel. cl. on NP comp.	Relative clause on NP complement
Rel. cl. on PP comp.	Relative clause on PP complement
Rel. cl. on NP object of P	Relative clause on NP object of P
Pass. with wh-moved subj.	Passive with wh-moved subject (with and without *by* phrase)
Pass. w. wh-mov. ind. obj.	Passive with wh-moved indirect object (with and without *by* phrase)
Pass. w. wh-mov. obj. of the *by phrase*	Passive with wh-moved object of the *by* phrase
Pass. w. wh-mov. *by* phrase	Passive with wh-moved *by* phrase
Cl. S mod. (decl.)	Clausal S modifier (declarative)
Cl. VP mod. (decl.)	Clausal VP modifier (declarative)
Cl. S mod. (pass., w. *by* phrase)	Clausal S modifier (passive, with *by* phrase)
Cl. VP mod. (pass., w. *by* phrase)	Clausal VP modifier (passive, with *by* phrase)
Cl. S mod. (pass., w.out *by* phrase)	Clausal S modifier (passive, without *by* phrase)
Cl. VP mod.	Clausal VP modifier
Adj. Sm. Cl. w. Sent. Comp.	Adjective Small Clause with Sentential Complement
NP Sm. Cl. w. Sent. Comp.	NP Small Clause with Sentential Complement
PP Sm. Cl. w. Sent. Comp.	PP Small Clause with Sentential Complement

	Tree families														
Constructions	Transitive	Ditransitive with PP shift	Ditransitive	Ditransitive with PP	Sentential Comp. with NP	Intransitive Verb Particle	Transitive Verb Particle	Ditransitive Verb Particle	Intransitive with PP	Sentential Complement	Light Verbs	Ditrans. Light Verbs w. PP Shift	Adj. Sm. Cl. w. Sentential Subj.	NP Sm. Cl. w. Sentential Subj.	PP Sm. Cl. w. Sentential Subj.
Declarative	22	76	22	✓	✓	✓	✓	✓	19	61	✓	✓	✓	✓	✓
Passive	✓	✓	✓	✓	83		✓	✓				✓			
Y/N quest.															
Wh-moved subject	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓
Wh-mov. NP complement	86	✓	89	✓	✓		✓	✓							
Wh-mov. S comp.					✓					✓					
Wh-mov. Adj. comp.												✓			
Wh-mov. object of a P		✓		90					✓			✓			
Wh-mov. PP		✓		91					✓			✓			
Topic. NP comp.	✓	✓	✓	✓	✓		✓	✓							
Imperative	102	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
Det. gerund	104	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
NP gerund	104	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
Ergative	55														
Rel. cl. on subj.	94	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓			
Rel. cl. on NP comp.	94	✓	✓	✓	✓		✓	✓							✓
Rel. cl. on PP comp.		✓		✓	✓				✓						
Rel. cl. on NP object of P		✓		✓	✓				✓			✓			
Pass. w. wh-mov. subj.	✓	✓	✓	✓	✓		✓	✓				✓			
Pass. w. wh-mov. ind. obj.		✓	✓	✓	✓			✓				✓			
Pass. w. wh-mov. obj. of *by* phrase	✓	✓	✓	✓	✓		✓	✓				✓			
Pass. w. wh-mov. *by* phrase	✓	✓	✓	✓	✓		✓	✓							
Cl. S mod. (decl.)	96	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Cl. VP mod. (decl.)	96	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Cl. S mod. (pass. w. *by* phrase)	✓	✓	✓	✓	✓		✓	✓				✓			
Cl. VP mod. (pass. w. *by* phrase)	✓	✓	✓	✓	✓		✓	✓				✓			
Cl. S mod. (pass. wo. *by* phrase)	✓	✓	✓	✓	✓		✓	✓				✓			
Cl. VP mod. (pass. wo. *by* phrase)	✓	✓	✓	✓	✓		✓	✓				✓			

	Tree families															
Constructions	Adverb It-Cleft	NP It-Cleft	PP It-Cleft	Adj. Small Clause	Adj. Sm. Cl. w. Sent. Comp.	Equative *BE*	NP Small Clause	NP Sm. Cl. w. Sent. Comp.	PP Small Clause	PP Sm. Cl. w. Sent. Comp.	Exhaustive PP Sm. Cl.	Exhaustive PP Small Clause with Sent. Comp.	Intransitive	Intransitive with Adjective	Transitive Idioms	Sentential Subject
Declarative	✓	✓	79	72	✓	74	70	✓	70	✓	✓	✓	✓	✓	✓	64
Passive																
Y/N quest.	✓	✓	79													
Wh-moved subject				✓	✓		✓	✓	✓	✓	✓	✓	89	62	✓	✓
Wh-mov. NP complement		✓					✓									
Wh-mov. S comp.															✓	
Wh-mov. Adj. comp.	✓			✓	✓								91			
Wh-mov. object of a P										✓		✓				
Wh-mov. PP			✓						✓	✓		✓				
Topic. NP comp.		✓					✓									
Imperative	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Det. gerund	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
NP gerund	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Ergative																
Rel. cl. on subj.				✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	
Rel. cl. on NP comp.																
Rel. cl. on PP comp.																
Rel. cl. on NP object of P									✓	✓		✓				
Pass. w. wh-mov. subj.																
Pass. w. wh-mov. ind. obj.																
Pass. w. wh-mov. obj. of *by* phrase																
Pass. w. wh-mov. *by* phrase																
Cl. S mod. (decl.)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Cl. VP mod. (decl.)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Cl. S mod. (pass. w. *by* phrase)																
Cl. VP mod. (pass. w. *by* phrase)																
Cl. S mod. (pass. wo. *by* phrase)																
Cl. VP mod. (pass. wo. *by* phrase)																

Chapter 6

Verb Classes

Each main[1] verb in the syntactic lexicon selects at least one tree family[2] (subcategorization frame). Since the tree database and syntactic lexicon are already separated for space efficiency (see Chapter 3), each verb can efficiently select a large number of trees by specifying a tree family, as opposed to each of the individual trees. This approach allows for a considerable reduction in the number of trees that must be specified for any given verb or form of a verb.

There are currently 38 tree families in the system.[3] This chapter gives a brief description of each tree family and shows the corresponding declarative tree[4], along with any peculiar characteristics or trees. It also indicates which transformations are in each tree family, and gives the number of verbs that select that family.[5] A few sample verbs are given, along with example sentences.

6.1 Intransitive: Tnx0V

Description: This tree family is selected by verbs that do not require an object complement of any type. Adverbs, prepositional phrases and other adjuncts may adjoin on, but are not required for the sentences to be grammatical. 2,091 verbs select this family.

Examples: *eat*, *sleep*, *dance*
Al ate.
Seth slept.
Hyun danced.

Declarative tree: See Figure 6.1.

Other available trees: wh-moved subject, subject relative clause, imperative, determiner gerund, NP gerund.

[1] Auxiliary verbs are handled under a different mechanism. See Chapter 20 for details.

[2] See section 3.6 for explanation of tree families.

[3] An explanation of the naming convention used in naming the trees and tree families is available in Appendix B.

[4] Before lexicalization, the ◇ indicates the anchor of the tree.

[5] Numbers given are as of August 1994 and are subject to some change with further development of the grammar.

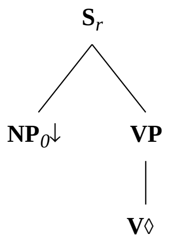


Figure 6.1: Declarative Intransitive Tree: αnx0V

6.2 Transitive: Tnx0Vnx1

Description: This tree family is selected by verbs that require only an NP object complement. The NP's may be complex structures, including gerund NP's and NP's that take sentential complements. This does not include light verb constructions (see sections 6.15 and 6.16). 4,335 verbs select the transitive tree family.

Examples: *eat*, *dance*, *take*, *like*
Al ate an apple.
Seth danced the tango.
Hyun is taking an algorithms course.
Anoop likes the fact that the semester is finished.

Declarative tree: See Figure 6.2.

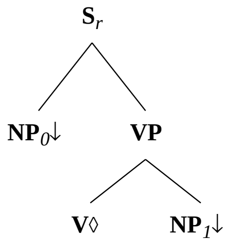


Figure 6.2: Declarative Transitive Tree: αnx0Vnx1

Other available trees: wh-moved subject, wh-moved object, subject relative clause, object relative clause, imperative, pre-sentential adjunct, post-sentential adjunct, determiner gerund, NP gerund, passive with *by* phrase, passive without *by* phrase, passive with wh-moved subject and *by* phrase, passive with wh-moved subject and no *by* phrase, passive with wh-moved object out of the *by* phrase, passive with wh-moved *by* phrase, passive with relative clause on subject and *by* phrase, passive with relative clause on subject and no *by* phrase, passive with relative clause on object on the *by* phrase, ergative, ergative with wh-moved subject, ergative with subject relative clause. In addition, two other trees that allow transitive verbs to function as adjectives (e.g. *the stopped truck*) are also in the family.

6.3 Transitive Idioms: Tnx0Vdn1

Description: This tree family is selected by idiomatic phrases in which the verb, determiner, and NP are all frozen (as in *He kicked the bucket.*). Only a limited number of transformations are allowed, as compared to the normal transitive tree family (see section 6.2). The analysis of idioms has not been done for idioms in general; this tree is included to illustrate how they could be handled in XTAG. Other idioms that have the same structure as *kick the bucket*, and that are limited to the same transformations would select this tree, while different tree families would be needed to handle other idioms. Note that *John kicked the bucket* is actually ambiguous, and would result in two parses - an idiom (meaning that John died), a simple transitive sentences (meaning that there is an physical bucket that John hit with his foot).

Examples: *kick the bucket*
Nixon kicked the bucket.

Declarative tree: See Figure 6.3.

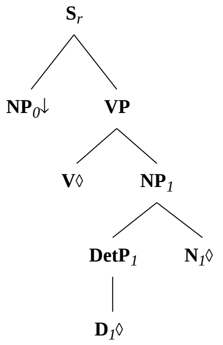


Figure 6.3: Declarative Transitive Idiom Tree: αnx0Vdn1

Other available trees: wh-moved subject, subject relative clause, imperative.

6.4 Ditransitive: Tnx0Vnx1nx2

Description: This tree family is selected by verbs that take exactly two NP complements. It does **not** include verbs that undergo the ditransitive verb shift (see section 6.6). The apparent ditransitive alternates involving verbs in this class and benefactive PP's (e.g. *John baked a cake for Mary*) are analyzed as transitives (see section 6.2) with a PP adjunct. Benefactives are taken to be adjunct PP's because they are optional (e.g. *John baked a cake* vs. *John baked a cake for Mary*). 54 verbs select the ditransitive tree family.

Examples: *ask*, *cook*, *win*
Christy asked Mike a question.
Doug cooked his father dinner.
Dania won her sister a stuffed animal.

Declarative tree: See Figure 6.4.

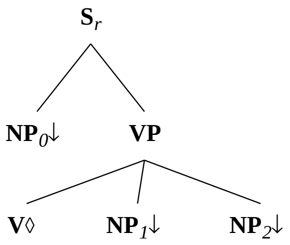


Figure 6.4: Declarative Ditransitive Tree: αnx0Vnx1nx2

Other available trees: wh-moved subject, wh-moved direct object, wh-moved indirect object, subject relative clause, direct object relative clause, indirect object relative clause, imperative, determiner gerund, NP gerund, passive with *by* phrase, passive without *by* phrase, passive with wh-moved subject and *by* phrase, passive with wh-moved subject and no *by* phrase, passive with wh-moved object out of the *by* phrase, passive with wh-moved *by* phrase, passive with wh-moved indirect object and *by* phrase, passive with wh-moved indirect object and no *by* phrase, passive with relative clause on subject and *by* phrase, passive with relative clause on subject and no *by* phrase, passive with relative clause on object of the *by* phrase, passive with relative clause on the indirect object and *by* phrase, passive with relative clause on the indirect object and no *by* phrase.

6.5 Ditransitive with PP: Tnx0Vnx1pnx2

Description: This tree family is selected by ditransitive verbs that take a noun phrase followed by a prepositional phrase. The preposition is not constrained. The preposition must be required and not optional - that is, the sentence must be ungrammatical with just the noun phrase (e.g. **John put the table*). No verbs, therefore, should select both this tree family and the transitive tree family (see section 6.2). This tree family is also distinguished from the ditransitive verbs, such as *give*, that undergo verb shifting (see section 6.6). There are 61 verbs that select this tree family.

Examples: *associate*, *put*, *refer*
Rostenkowski associated money with power.
He put his reputation on the line.
He referred all questions to his attorney.

Declarative tree: See Figure 6.5.

Other available trees: wh-moved subject, wh-moved direct object, wh-moved object of PP, wh-moved PP, subject relative clause, direct object relative clause, object of PP relative clause, imperative, determiner gerund, NP gerund, passive with *by* phrase, passive without *by* phrase, passive with wh-moved subject and *by* phrase, passive with wh-moved subject and no *by* phrase, passive with wh-moved object out of the *by* phrase, passive with wh-moved *by* phrase, passive with wh-moved object out of the PP and *by* phrase, passive

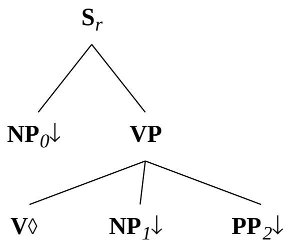


Figure 6.5: Declarative Ditransitive with PP Tree: αnx0Vnx1pnx2

with wh-moved object out of the PP and no *by* phrase, passive with wh-moved PP and *by* phrase, passive with wh-moved PP and no *by* phrase, passive with relative clause on subject and *by* phrase, passive with relative clause on subject and no *by* phrase, passive with relative clause on object of the *by* phrase, passive with relative clause on the object of the PP and *by* phrase, passive with relative clause on the object of the PP and no *by* phrase.

6.6 Ditransitive with PP shift: Tnx0Vnx1Pnx2

Description: This tree family is selected by ditransitive verbs that undergo a shift to a *to* prepositional phrase. These ditransitive verbs are clearly constrained so that when they shift, the prepositional phrase must start with *to*. This is in contrast to the Ditransitives with PP in section 6.5, in which verbs may appear in [NP V NP PP] constructions with a variety of prepositions. Both the dative shifted and non-shifted PP complement trees are included. 55 verbs select this family.

Examples: *give*, *promise*, *tell*
Bill gave Hillary flowers.
Bill gave flowers to Hillary.
Whitman promised the voters a tax cut.
Whitman promised a tax cut to the voters.
Pinnochino told Gepetto a lie.
Pinnochino told a lie to Gepetto.

Declarative tree: See Figure 6.6.

Other available trees: Non-shifted: wh-moved subject, wh-moved direct object, wh-moved indirect object, subject relative clause, direct object relative clause, indirect object relative clause, imperative, NP gerund, passive with *by* phrase, passive without *by* phrase, passive with wh-moved subject and *by* phrase, passive with wh-moved subject and no *by* phrase, passive with wh-moved object out of the *by* phrase, passive with wh-moved *by* phrase, passive with wh-moved indirect object and *by* phrase, passive with wh-moved indirect object and no *by* phrase, passive with relative clause on subject and *by* phrase, passive with relative clause on subject and no *by* phrase, passive with relative clause on object of the *by* phrase, passive with relative clause on the indirect object and *by* phrase, passive

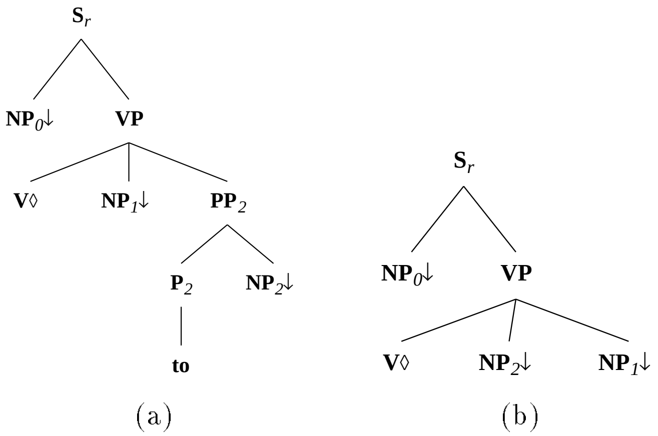


Figure 6.6: Declarative Ditransitive with PP shift Trees: αnx0Vnx1Pnx2 (a) and αnx0Vnx2nx1 (b)

with relative clause on the indirect object and no *by* phrase;
Shifted: wh-moved subject, wh-moved direct object, wh-moved object of PP, wh-moved PP, subject relative clause, direct object relative clause, object of PP relative clause, imperative, determiner gerund, NP gerund, passive with *by* phrase, passive without *by* phrase, passive with wh-moved subject and *by* phrase, passive with wh-moved subject and no *by* phrase, passive with wh-moved object out of the *by* phrase, passive with wh-moved *by* phrase, passive with wh-moved object out of the PP and *by* phrase, passive with wh-moved object out of the PP and no *by* phrase, passive with wh-moved PP and *by* phrase, passive with wh-moved PP and no *by* phrase, passive with relative clause on subject and *by* phrase, passive with relative clause on subject and no *by* phrase, passive with relative clause on object of the *by* phrase, passive with relative clause on the object of the PP and *by* phrase, passive with relative clause on the object of the PP and no *by* phrase.

6.7 Sentential Complement with NP: Tnx0Vnx1s2

Description: This tree family is selected by verbs that take both an NP and a sentential complement. The sentential complement may be infinitive or indicative. The type of clause is specified by each individual verb in its syntactic lexicon entry. A given verb may select more than one type of sentential complement. The declarative tree, and many other trees in this family, are auxiliary trees, as opposed to the more common initial trees. These auxiliary trees adjoin onto an S node in an existing tree of the type specified by the sentential complement. This is the mechanism by which TAGs are able to maintain long-distance dependencies (see Chapter 13), even over multiple embeddings (e.g. *What did Bill tell Mary that John said?*). 106 verbs select this tree family.

Examples: *beg*, *expect*, *tell*
Srini begged Mark to increase his disk quota.
Jim expected Beth to feed the dogs.
Beth told Jim that it was his turn.

Declarative tree: See Figure 6.7.

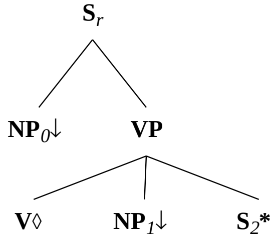


Figure 6.7: Declarative Sentential Complement with NP Tree: βnx0Vnx1s2

Other available trees: wh-moved subject, wh-moved object, wh-moved sentential complement, subject relative clause, object relative clause, imperative, determiner gerund, NP gerund, passive with *by* phrase before sentential complement, passive with *by* phrase after sentential complement, passive without *by* phrase, passive with wh-moved subject and *by* phrase before sentential complement, passive with wh-moved subject and *by* phrase after sentential complement, passive with wh-moved subject and no *by* phrase, passive with wh-moved object out of the *by* phrase, passive with wh-moved *by* phrase, passive with relative clause on subject and *by* phrase before sentential complement, passive with relative clause on subject and *by* phrase after sentential complement, passive with relative clause on subject and no *by* phrase.

6.8 Intransitive Verb Particle: Tnx0Vpl

Description: The trees in this tree family are anchored by both the verb and the verb particle. Both appear in the syntactic lexicon and together select this tree family. Intransitive verb particles can be difficult to distinguish from intransitive verbs with adverbs adjoined on. The main diagnostics for including verbs in this class are whether the meaning is compositional or not, and whether there is a transitive version of the verb/verb particle combination with the same or similar meaning. The existence of an alternate compositional meaning is a strong indication for a separate verb particle construction. There are 164 verb/verb particle combinations.

Examples: *add up*, *come out*, *sign off*
The numbers never quite added up.
John finally came out (of the closet).
I think that I will sign off now.

Declarative tree: See Figure 6.8.

Other available trees: wh-moved subject, subject relative clause, imperative, determiner gerund, NP gerund.

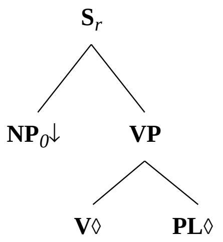


Figure 6.8: Declarative Intransitive Verb Particle Tree: αnx0Vpl

6.9 Transitive Verb Particle: Tnx0Vplnx1

Description: Verb/verb particle combinations that take an NP complement select this tree family. Both the verb and the verb particle are anchors of the trees. Particle movement has been taken as the diagnostic to distinguish verb particle constructions from intransitives with adjoined PP's. If the alleged particle is able to undergo particle movement, in other words appear both before and after the direct object, then it is judged to be a particle. Items that do not undergo particle movement are taken to be prepositions. In many, but not all, of the verb particle cases, there is also an alternate prepositional meaning in which the lexical item did not move. (e.g. *He looked up the number (in the phonebook). He looked the number up. Srini looked up the road (for Purnima's car). *He looked the road up.*) There are 841 verb/verb particle combinations.

Examples: *blow off*, *make up*, *pick out*
He blew off his linguistics class for the third time.
He blew his linguistics class off for the third time.
The dyslexic leprechaun made up the syntactic lexicon.
The dyslexic leprechaun made the syntactic lexicon up.
I would like to pick out a new computer.
I would like to pick a new computer out.

Declarative tree: See Figure 6.9.

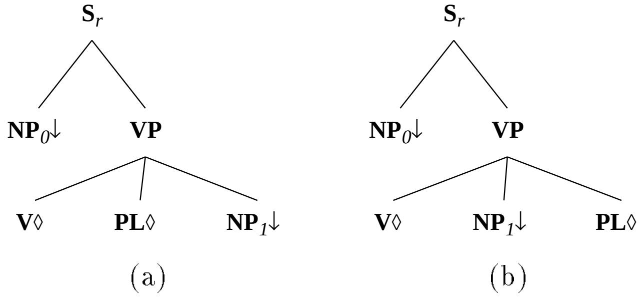


Figure 6.9: Declarative Transitive Verb Particle Tree: αnx0Vplnx1 (a) and αnx0Vnx1pl (b)

Other available trees: wh-moved subject with particle before the NP, wh-moved subject with particle after the NP, wh-moved object, subject relative clause with particle before the NP, subject relative clause with particle after the NP, object relative clause, imperative with particle before the NP, imperative with particle after the NP, determiner gerund with particle before the NP, NP gerund with particle before the NP, NP gerund with particle after the NP, passive with *by* phrase, passive without *by* phrase, passive with wh-moved subject and *by* phrase, passive with wh-moved subject and no *by* phrase, passive with wh-moved object out of the *by* phrase, passive with wh-moved *by* phrase, passive with relative clause on subject and *by* phrase, passive with relative clause on subject and no *by* phrase, passive with relative clause on object of the *by* phrase.

6.10 Ditransitive Verb Particle: Tnx0Vplnx1nx2

Description: Verb/verb particle combinations that select this tree family take 2 NP complements. Both the verb and the verb particle anchor the trees, and the verb particle can occur before, between, or after the noun phrases. Perhaps because of the complexity of the sentence, these verbs do not seem to have passive alternations (∗*A new bank account was opened up Michelle by me*). There are 4 verb/verb particle combinations that select this tree family. The exhaustive list is given in the examples.

Examples: *dish out*, *open up*, *pay off*, *rustle up*
I opened up Michelle a new bank account.
I opened Michelle up a new bank account.
I opened Michelle a new bank account up.

Declarative tree: See Figure 6.10.

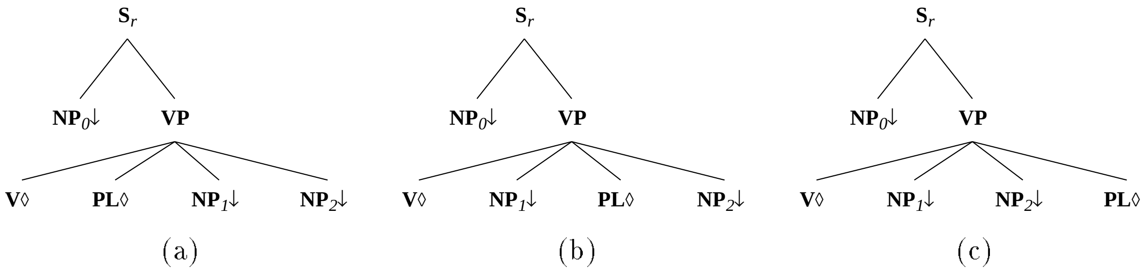


Figure 6.10: Declarative Ditransitive Verb Particle Tree: αnx0Vplnx1nx2 (a), αnx0Vnx1plnx2 (b) and αnx0Vnx1nx2pl (c)

Other available trees: wh-moved subject with particle before the NP's, wh-moved subject with particle between the NP's, wh-moved subject with particle after the NP's, wh-moved indirect object with particle before the NP's, wh-moved indirect object with particle after the NP's, wh-moved direct object with particle before the NP's, wh-moved direct object with particle between the NP's, subject relative clause with particle before the NP's, subject relative clause with particle between the NP's, subject relative clause with particle after the NP's, indirect object relative clause with particle before the NP's, indirect object relative clause with particle after the NP's, direct object relative clause with particle before the NP's, direct object relative clause with particle between the NP's, imperative with particle before the NP's, imperative with particle between the NP's, imperative with particle after the NP's, determiner gerund with particle before the NP's, NP gerund with particle before the NP's, NP gerund with particle between the NP's, NP gerund with particle after the NP's.

6.11 Intransitive with PP: Tnx0Vpnx1

Description: The verbs that select this tree family are not strictly intransitive, in that they **must** be followed by a prepositional phrase. Verbs that are intransitive and simply **can** be followed by a prepositional phrase do not select this family, but instead have the PP adjoin onto the intransitive sentence. Accordingly, there should be no verbs in both this class and the intransitive tree family (see section 6.1). The prepositional phrase is not restricted to being headed by any particular lexical item. Note that these are not transitive verb particles (see section 6.9), since the head of the PP does not move. 171 verbs select this tree family.

Examples: *grab*, *impinge*, *provide*
Seth grabbed for the brass ring.
The noise gradually impinged on Dania's thoughts.
A good host provides for everyone's needs.

Declarative tree: See Figure 6.11.

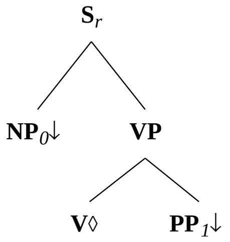


Figure 6.11: Declarative Intransitive with PP Tree: αnx0Vpnx1

Other available trees: wh-moved subject, wh-moved object of the PP, wh-moved PP, subject relative clause, object of the PP relative clause, imperative, determiner gerund, NP gerund, passive with *by* phrase, passive without *by* phrase, passive with wh-moved subject and *by* phrase, passive with wh-moved subject and no *by* phrase, passive with wh-moved *by* phrase, passive with relative clause on subject and *by* phrase, passive with relative clause on subject and no *by* phrase, passive with relative clause on object of the *by* phrase.

6.12 Sentential Complement: Tnx0Vs1

Description: This tree family is selected by verbs that take just a sentential complement. The sentential complement may be of type infinitive, indicative, or small clause (see Chapter 9). The type of clause is specified by each individual verb in its syntactic lexicon entry, and a given verb may select more than one type of sentential complement. The declarative tree, and many other trees in this family, are auxiliary trees, as opposed to the more common initial trees. These auxiliary trees adjoin onto an S node in an existing tree of the type specified by the sentential complement. This is the mechanism by which TAGs are able to maintain long-distance dependencies (see Chapter 13), even over multiple embeddings (e.g. *What did Bill think that John said?*). 318 verbs select this tree family.

Examples: *consider*, *think*

Dania considered the algorithm unworkable.
Srini thought that the program was working.

Declarative tree: See Figure 6.12.

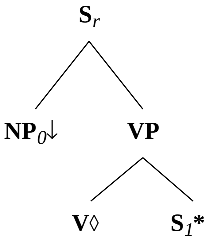


Figure 6.12: Declarative Sentential Complement Tree: βnx0Vs1

Other available trees: wh-moved subject, wh-moved sentential complement, subject relative clause, imperative, determiner gerund, NP gerund.

6.13 Intransitive with Adjective: Tnx0Va1

Description: The verbs that select this tree family take an adjective as a complement. The adjective may be regular, comparative, or superlative. It may also be formed from the special class of adjectives derived from the transitive verbs (e.g. *agitated, broken*). See section 6.2). Unlike the Intransitive with PP verbs (see section 6.11), some of these verbs may also occur as bare intransitives as well. This distinction is drawn because adjectives do not normally adjoin onto sentences, as prepositional phrases do. Other intransitive verbs can only occur with the adjective, and these select only this family. The verb class is also distinguished from the adjective small clauses (see section 6.20) because these verbs are not raising verbs. 34 verbs select this tree family.

Examples: *become*, *grow*, *smell*

The greenhouse became hotter.
The plants grew tall and strong.
The flowers smelled wonderful.

Declarative tree: See Figure 6.13.

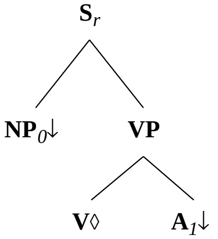


Figure 6.13: Declarative Intransitive with Adjective Tree: αnx0Va1

Other available trees: wh-moved subject, wh-moved adjective (*how*), subject relative clause, imperative, NP gerund.

6.14 Sentential Subject: Ts0Vnx1

Description: The verbs that select this tree family all take sentential subjects, and are often referred to as 'psych' verbs, since they all refer to some psychological state of mind. The sentential subject can be indicative (complementizer required) or infinitive (complementizer optional). 100 verbs that select this tree family.

Examples: *delight*, *impress*, *surprise*
That the tea had rosehips in it delighted Christy.
To even attempt a marathon impressed Dania.
For Jim to have walked the dogs surprised Beth.

Declarative tree: See Figure 6.14.

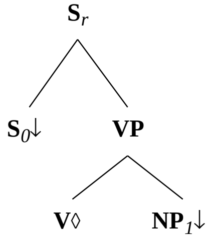


Figure 6.14: Declarative Sentential Subject Tree: αs0Vnx1

Other available trees: wh-moved subject, wh-moved object.

6.15 Light Verbs: Tnx0lVN1, Tnx0lVdxN1

Description: The verb/noun pairs that select these tree families are pairs in which the interpretation is non-compositional and the noun contributes argument structure to the predicate (e.g. *The man took a walk.* vs. *The man took a radio*). The verb and the noun occur together in the syntactic database, and both anchor the trees. The verbs in the light verb constructions are *do*, *give*, *have*, *make* and *take*. The noun following the light verb is (usually) in a bare infinitive form (*have a good cry*) and usually occurs with *a(n)*. However, we include deverbal nominals (*take a bath*, *give a demonstration*) as well. Constructions with nouns that do not contribute an argument structure (*have a cigarette*, *give* NP *a black eye*) are excluded. In addition to semantic considerations of light verbs, they differ syntactically from Transitive verbs (section 6.2) as well in that the noun in the light verb construction does not extract. Because the noun is an anchor in the tree, there are two different tree families representing nouns that require determiners and those that occur without them (see Chapter 18). There are 96 verb/noun pairs that select the light verb tree without determiners, and 242 that select the light verb tree with determiners.

Examples: *give groan*, *have discussion*, *make comment*
The audience gave a collective groan.
We had a big discussion about closing the libraries.
The professors made comments on the paper.

Declarative tree: See Figure 6.15.

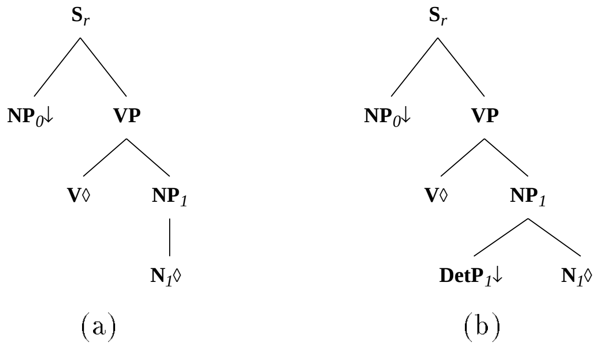


Figure 6.15: Declarative Light Verb Trees: αnx0lVN1 (a) and αnx0lVdxN1 (b)

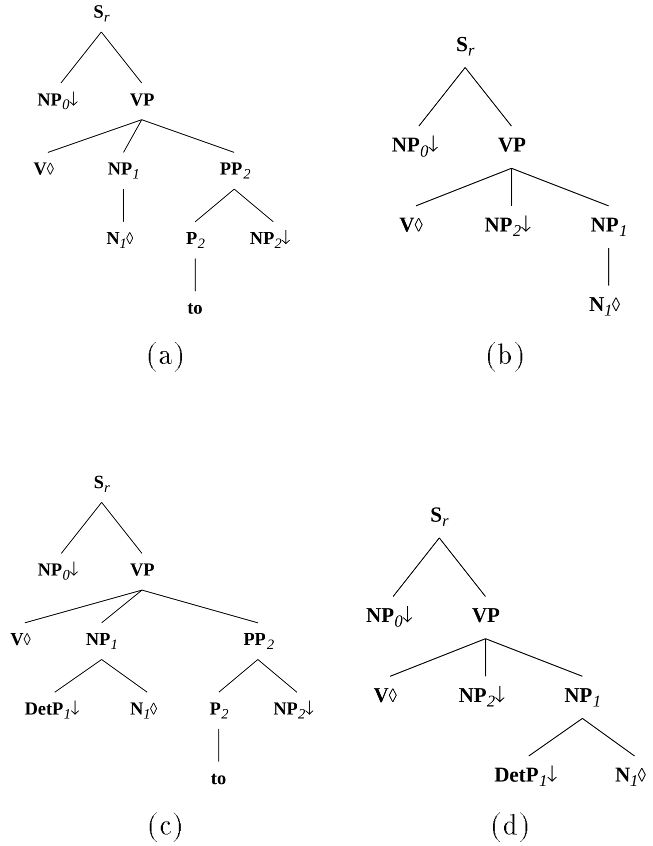
Other available trees: wh-moved subject, subject relative clause, imperative, determiner gerund, NP gerund.

6.16 Ditransitive Light Verbs with PP Shift: Tnx0lVN1Pnx2, Tnx0lVdxN1Pnx2

Description: The verb/noun pairs that select these tree families are pairs in which the interpretation is non-compositional and the noun contributes argument structure to the predicate (e.g. *Dania made Srini a cake.* vs. *Dania made Srini a loan*). The verb and the noun occur together in the syntactic database, and both anchor the trees. The verbs in these light verb constructions are *give* and *make*. The noun following the light verb is (usually) a bare infinitive form (e.g. *make a promise to Anoop*). However, we include deverbal nominals (e.g. *make a payment to Anoop*) as well. Constructions with nouns that do not contribute an argument structure are excluded. In addition to semantic considerations of light verbs, they differ syntactically from the Ditransitive with PP Shift verbs (see section 6.6) as well in that the noun in the light verb construction does not extract. Also, passivization is severely restricted. Because the noun is an anchor in the tree, there are two different tree families representing nouns that require determiners and those that occur without them (see Chapter 18). There are 10 verb/noun pairs that select the trees without determiners, and 18 that select the trees with determiners.

Examples: *give look*, *give wave*, *make promise*
Dania gave Carl a murderous look.
Amanda gave us a little wave as she left.
Dania made Doug a promise.

Declarative tree: See Figure 6.16.



(a) (b)

(c) (d)

Figure 6.16: Declarative Light Verbs with PP Tree: αnx0lVN1Pnx2 (a), αnx0lVnx2N1 (b), αnx0lVdxN1Pnx2 (c) and αnx0lVnx2dxN1 (d)

Other available trees: Non-shifted: wh-moved subject, wh-moved indirect object, subject relative clause, indirect object relative clause, imperative, NP gerund, passive with *by* phrase

Shifted: wh-moved subject, wh-moved object of PP, wh-moved PP, subject relative clause, object of PP relative clause, imperative, determiner gerund, NP gerund, passive with *by* phrase.

6.17 NP It-Cleft: TItVnx1s2

Description: This tree family is selected by *be* as the main verb and *it* as the subject. Together these two items serve as a multi-component anchor for the tree family. This tree family is used for it-clefts in which the clefted element is an NP and there are no gaps in the clause which follows the NP. The NP is interpreted as an adjunct of the following clause. See Chapter 11 for additional discussion.

Examples: *it be*
It was yesterday that we had the meeting.

Declarative tree: See Figure 6.17.

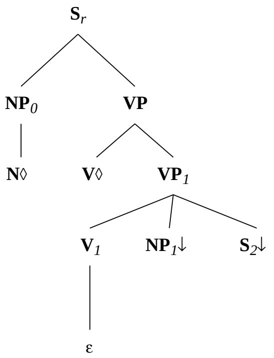


Figure 6.17: Declarative NP It-Cleft Tree: αItVnx1s2

Other available trees: inverted question, wh-moved object with *be* inverted, wh-moved object with *be* not inverted.

6.18 PP It-Cleft: TItVpnx1s2

Description: This tree family is selected by *be* as the main verb and *it* as the subject. Together these two items serve as a multi-component anchor for the tree family. This tree family is used for it-clefts in which the clefted element is an PP and there are no gaps in the clause which follows the PP. The PP is interpreted as an adjunct of the following clause. See Chapter 11 for additional discussion.

Examples: *it be*
It was at Kent State that the police shot all those students.

Declarative tree: See Figure 6.18.

Other available trees: inverted question, wh-moved prepositional phrase with *be* inverted, wh-moved prepositional phrase with *be* not inverted.

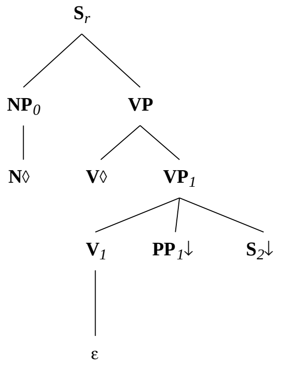


Figure 6.18: Declarative PP It-Cleft Tree: αItVnx1s2

6.19 Adverb It-Cleft: TItVad1s2

Description: This tree family is selected by *be* as the main verb and *it* as the subject. Together these two items serve as a multi-component anchor for the tree family. This tree family is used for it-clefts in which the clefted element is an adverb and there are no gaps in the clause which follows the adverb. The adverb is interpreted as an adjunct of the following clause. See Chapter 11 for additional discussion.

Examples: *it be*
It was reluctantly that Dania agreed to do the tech report.

Declarative tree: See Figure 6.19.

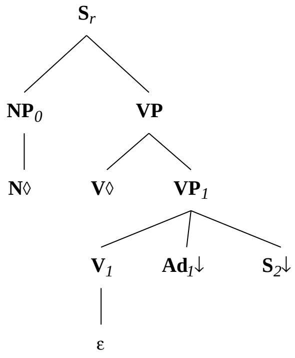


Figure 6.19: Declarative Transitive Tree: αItVad1s2

Other available trees: inverted question, wh-moved adverb *how* with *be* inverted, wh-moved adverb *how* with *be* not inverted.

6.20 Adjective Small Clause Tree: Tnx0Ax1

Description: These trees are not anchored by verbs, but by adjectives. They are explained in much greater detail in the section on small clauses (see section 9.3). This section is presented here for completeness. 3312 adjectives select this tree family.

Examples: *addictive*, *dangerous*, *wary*

Cigarettes are addictive.
Smoking cigarettes is dangerous.
John seems wary of the Surgeon General's warnings.

Declarative tree: See Figure 6.20.

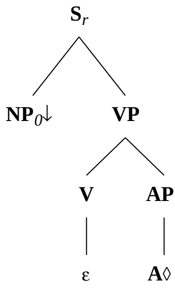


Figure 6.20: Declarative Adjective Small Clause Tree: αnx0Ax1

Other available trees: wh-moved subject, wh-moved adjective *how*, relative clause on subject, imperative, NP gerund.

6.21 Adjective Small Clause with Sentential Complement: Tnx0Ax1s2

Description: This tree family is selected by adjectives that take sentential complements. The sentential complements can be indicative or infinitive. Note that these trees are anchored by adjectives, not verbs. Most adjectives that take the Adjective Small Clause tree family (see section 6.20) take this family as well.[6] Small clauses are explained in much greater detail in section 9.3. This section is presented here for completeness. 3,229 adjectives select this tree family.

Examples: *able*, *curious*, *disappointed*

Christy was able to find the problem.
Christy was curious whether the new analysis was working.
Christy was sad that the old analysis failed.

Declarative tree: See Figure 6.21.

Other available trees: wh-moved subject, wh-moved adjective *how*, relative clause on subject, imperative, NP gerund.

[6] No great attempt has been made to go through and decide which adjectives should actually take this family and which should not.

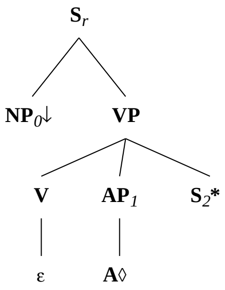


Figure 6.21: Declarative Adjective Small Clause with Sentential Complement Tree: αnx0Ax1s2

6.22 Adjective Small Clause with Sentential Subject: Ts0Ax1

Description: This tree family is selected by adjectives that take sentential subjects. The sentential subjects can be indicative or infinitive. Note that these trees are not anchored by adjectives, not verbs. Most adjectives that take the Adjective Small Clause tree family (see section 6.20) take this family as well.[7] Small clauses are explained in much greater detail in section 9.3. This section is presented here for completeness. 3,227 adjectives select this tree family.

Examples: *decadent*, *incredible*, *uncertain*
To eat raspberry chocolate truffle ice cream is decadent.
That Carl could eat a large bowl of it is incredible.
Whether he will actually survive the experience is uncertain.

Declarative tree: See Figure 6.22.

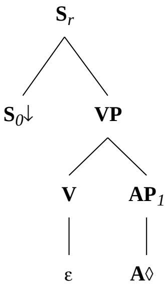


Figure 6.22: Declarative Adjective Small Clause with Sentential Subject Tree: αs0Ax1

Other available trees: wh-moved subject.

6.23 Equative *BE*: Tnx0BEnx1

Description: This tree family is selected only by the verb *be*. It is distinguished from the

[7] No great attempt has been made to go through and decide which adjectives should actually take this family and which should not.

predicative NP's (see section 6.24) in that two NP's are equated, and hence interchangeable (see Chapter 9 for more discussion on the English copula and predicative sentences). The XTAG analysis for equative *be* is explained in greater detail in section 9.4.

Examples: *be*

That man is my uncle.

Declarative tree: See Figure 6.23.

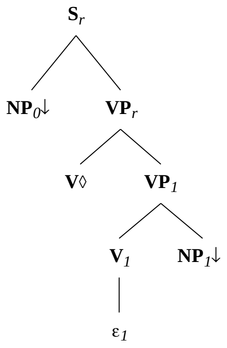


Figure 6.23: Declarative Equative *BE* Tree: αnx0BEnx1

Other available trees: inverted-question.

6.24 NP Small Clauses: Tnx0N1, Tnx0dxN1

Description: The trees in these tree families are not anchored by verbs, but by nouns. Because they are anchored by nouns, there are two different tree families representing nouns that require determiners and those that occur without them. Small clauses are explained in much greater detail in section 9.3. This section is presented here for completeness. 9,915 nouns select the tree family without determiners, and 9,888 nouns select the family with determiners.

Examples: *author*, *chair*, *dish*

Dania is an author.

That blue, warped-looking thing is a chair.

Those broken pieces were dishes.

Declarative tree: See Figure 6.24.

Other available trees: wh-moved subject, wh-moved object, relative clause on object, imperative, NP gerund.

6.25 NP with Sentential Complement Small Clause: Tnx0dxN1s1, Tnx0N1s1

Description: These tree families are selected by the small group of nouns that take sentential

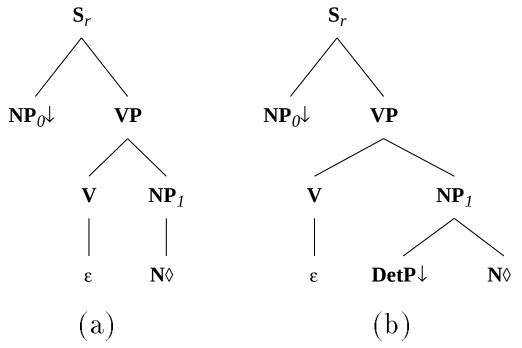


Figure 6.24: Declarative NP Small Clause Trees: αnx0N1 (a) and αnx0dxN1 (b)

complements by themselves (see section 8.8). The sentential complements can be indicative or infinitive, depending on the noun. Because the trees are anchored by nouns, there are two different tree families representing nouns that require determiners and those that occur without them. Small clauses in general are explained in much greater detail in the section 9.3. This section is presented here for completeness. 216 nouns collectively select these two families.

Examples: *admission*, *claim*, *vow*

The affidavits are admissions that they killed the sheep.
There is always the claim that they were insane at the time.
This is his vow to fight the charges.

Declarative tree: See Figure 6.25.

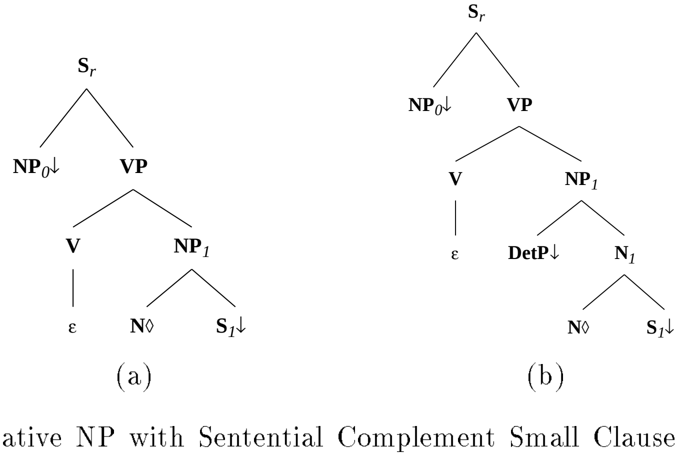


Figure 6.25: Declarative NP with Sentential Complement Small Clause Tree: αnx0N1s1 (a) and αnx0dxN1s1 (b)

Other available trees: wh-moved subject, wh-moved object, relative clause on object, imperative, NP gerund.

6.26 NP Small Clause with Sentential Subject: Ts0dxN1, Ts0N1

Description: These tree families are selected by nouns that take sentential subjects. The sentential subjects can be indicative or infinitive. Note that these trees are anchored by nouns, not verbs. Because they are anchored by nouns, there are two different tree families representing nouns that require determiners and those that occur without them. Most nouns that take the NP Small Clause tree family (see section 6.24) take this family as well.[8] Small clauses are explained in much greater detail in section 9.3. This section is presented here for completeness. 9,888 nouns select both the tree family with determiners and the tree family without determiners.

Examples: *dilemma*, *insanity*, *tragedy*

Whether to keep the job he hates is a dilemma.
For Bill to invest all of his money in worms is insanity.
That the worms died is a tragedy.

Declarative tree: See Figure 6.26.

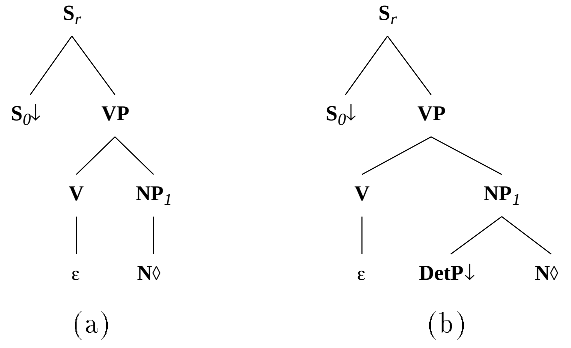


Figure 6.26: Declarative NP Small Clause with Sentential Subject Tree: αs0N1 (a) and αs0dxN1 (b)

Other available trees: wh-moved subject.

6.27 PP Small Clause: Tnx0Pnx1

Description: This family is selected by prepositions that can occur in small clause constructions. For more information on small clause constructions, see section 9.3. This section is presented here for completeness. 39 prepositions select this tree family.

Examples: *around*, *in*, *underneath*

Chris is around the corner.
Trisha is in big trouble.
The dog is underneath the table.

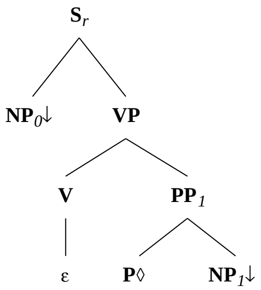


Figure 6.27: Declarative PP Small Clause Tree: αnx0Pnx1

Declarative tree: See Figure 6.27.

Other available trees: wh-moved subject, wh-moved object of PP, relative clause on subject, relative clause on object of PP, imperative, NP gerund.

6.28 Exhaustive PP Small Clause: Tnx0Px1

Description: This family is selected by **exhaustive** prepositions that can occur in small clauses. Exhaustive prepositions are prepositions that function as prepositional phrases by themselves. For more information on small clause constructions, please see section 9.3. The section is included here for completeness. 8 exhaustive prepositions select this tree family.

Examples: *abroad*, *below*, *outside*
Dr. Joshi is abroad.
The workers are all below.
Clove is outside.

Declarative tree: See Figure 6.28.

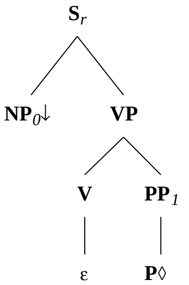


Figure 6.28: Declarative Exhaustive PP Small Clause Tree: αnx0Px1

[8] No great attempt has been made to go through and decide which nouns should actually take this family and which should not.

Other available trees: wh-moved subject, wh-moved PP, relative clause on subject, imperative, NP gerund.

6.29 PP Small Clause with Sentential Subject: Ts0Pnx1

Description: This tree family is selected by prepositions that take sentential subjects. The sentential subject can be indicative or infinitive. Small clauses are explained in much greater detail in section 9.3. This section is presented here for completeness. 39 prepositions select this tree family.

Examples: *beyond*, *unlike*

That Ken could forget to pay the taxes is beyond belief.

To explain how this happened is outside the scope of this discussion.

For Ken to do something right is unlike him.

Declarative tree: See Figure 6.29.

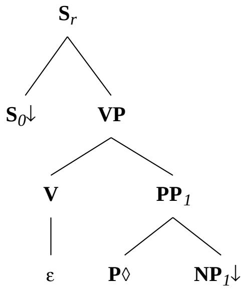


Figure 6.29: Declarative PP Small Clause with Sentential Subject Tree: αs0Pnx1

Other available trees: wh-moved subject, relative clause on object of the PP.

Chapter 7

Ergatives

Verbs in English that are termed ergative display the kind of alternation shown in the sentences in (7) below.

(7) The sun melted the ice.
The ice melted.

The pattern of ergative pairs as seen in (7) is for the object of the transitive sentence to be the subject of the intransitive sentence. The literature discussing such pairs is based largely on syntactic models that involve movement, particularly GB. Within that framework two basic approaches are discussed:

- **Derived Intransitive**
  The intransitive member of the ergative pair is derived through processes of movement and deletion from:
  - a transitive D-structure ([Burzio, 1986]); or
  - transitive lexical structure ([Hale and Keyser, 1986; Hale and Keyser, 1987])
- **Pure Intransitive**
  The intransitive member is intransitive at all levels of the syntax and the lexicon and is not related to the transitive member syntactically or lexically ([Napoli, 1988]).

Obviously, the Derived Intransitive approach's notions of movement in the lexicon or in the grammar cannot be represented as such in lexicalized tag. However, distinctions drawn in these arguments can be translated to the FB-LTAG framework. In the XTAG grammar the difference between these two approaches is not a matter of movement but rather a question of tree family membership. The relation between sentences represented in terms of movement in other frameworks is represented in XTAG by membership in the same tree family. Wh-questions and their indicative counterparts are one example of this. Adopting the Pure Intransitive approach suggested by [Napoli, 1988] would mean placing the intransitive ergatives in a tree family with other intransitive verbs and separate from the transitive variants of the same verbs. This would result in a grammar that represented intransitive ergatives as more closely related to other intransitives than to their transitive counterparts. The only hint of the relation between the intransitive ergatives and the transitive ergatives would be that ergative verbs would select both tree families. While this is a workable solution, it is an unattractive one for the English XTAG grammar because semantic coherence is implicitly associated with tree families in our analysis of other constructions. In particular, constancy in thematic role is represented by constancy in node names across sentence types within a tree family. For example, if the object of a declarative tree is NP_1 the subject of the passive tree(s) in that family will also be NP_1.

The analysis that has been implemented in the English XTAG grammar is an adaptation of the Derived Intransitive approach. The ergative verbs select one family, Tnx0Vnx1, that contains both transitive and intransitive trees. The <**trans**> feature appears on the intransitive ergative trees with the value − and on the transitive trees with the value +. This creates the two possibilities needed to account for the data.

- **intransitive ergative/transitive alternation.** These verbs have transitive and intransitive variants as shown in sentences (8) and (9).

  (8) The sun melted the ice cream.

  (9) The ice cream melted.

  In the English XTAG grammar, verbs with this behavior are left unspecified as to value for the <**trans**> feature. This lack of specification allows these verbs to anchor either type of tree in the Tnx0Vnx1 tree family because the unspecified <**trans**> value of the verb can unify with either + or − values in the trees.

- **transitive only.** Verbs of this type select only the transitive trees and do not allow intransitive ergative variants as in the pattern show in sentences (10) and (11).

  (10) Elmo borrowed a book.

  (11) ∗A book borrowed.

  The restriction to selecting only transitive trees is accomplished by setting the <**trans**> feature value to + for these verbs.

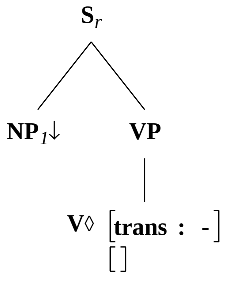


Figure 7.1: Ergative Tree: αEnx1V

The declarative ergative tree is shown in Figure 7.1 with the <**trans**> feature displayed. Note that the index of the subject NP indicates that it originated as the object of the verb.

Chapter 8

Sentential Subjects and Sentential Complements

In the XTAG grammar, arguments of a lexical item, including subjects, appear in the initial tree anchored by that lexical item. A sentential argument appears as an S node in the appropriate position within an elementary tree anchored by the lexical item that selects it. This is the case for sentential complements of verbs, prepositions and nouns and for sentential subjects. The distribution of complementizers in English is intertwined with the distribution of embedded sentences. A successful analysis of complementizers in English must handle both the cooccurrence restrictions between complementizers and various types of clauses, and the distribution of the clauses themselves, in both subject and complement positions.

8.1 S or VP complements?

Two comparable grammatical formalisms, Generalized Phrase Structure Grammar (GPSG) ([Gazdar *et al.*, 1985]) and Head-driven Phrase Structure Grammar (HPSG) ([Pollard and Sag, 1987]), have rather different treatments of sentential complements (S-comps). They both treat embedded sentences as VP's with subjects, which generates the correct structures but misses the generalization that S's behave similarly in both matrix and embedded environments, and VP's behave quite differently. Neither account has PRO subjects of infinitival clauses– they have subjectless VP's instead. GPSG has a complete complementizer system, which appears to cover the same range of data as our analysis. It is not clear what sort of complementizer analysis could be implemented in HPSG.

Following standard GB approach, the English XTAG grammar does not allow VP complements but treats verb-anchored structures without overt subjects as having PRO subjects. Thus, indicative clauses, infinitives and gerunds all have a uniform treatment as embedded clauses using the same trees under this approach. Furthermore, our analysis is able to preserve the selectional and distributional distinction between S's and VP's, in the spirit of GB theories, without having to posit 'extra' empty categories.[1] Consider the alternation between *that* and the null complementizer[2], shown in sentences (12) and (13).

[1] i.e. empty complementizers. We do have PRO and NP traces in the grammar.

[2] Although we will continue to refer to 'null' complementizers, in our analysis this is actually the absence of

(12) He hopes ∅ Muriel wins.

(13) He hopes that Muriel wins.

In GB both *Muriel wins* in (12) and *that Muriel wins* in (13) are CPs even though there is no overt complementizer to head the phrase in (12). Our grammar does not distinguish by category label between the the phrases that would be labeled in GB as IP and CP. We label both of these phrases S. The difference between these two levels is the presence or absence of the complementizer (or extracted WH constituent), and is represented in our system as a difference in feature values (here, of the <**comp**> feature), and the presence of the additional structure contributed by the complementizer or extracted constituent. This illustrates an important distinction in XTAG, that between features and node labels. Because we have a sophisticated feature system, we are able to make fine-grained distinctions between nodes with the same label which in another system might have to be realized by distinguishing node labels.

8.2 Complementizers and Embedded Clauses in English: The Data

Verbs selecting sentential complements (or subjects) place restrictions on their complements, in particular, on the form of the embedded verb phrase.[3] Furthermore, complementizers are constrained to appear with certain types of clauses, again, based primarily on the form of the embedded VP. For example, *hope* selects both indicative and infinitival complements. With an indicative complement, it may only have *that* or null as possible complementizers; with an infinitival complement, it may only have a null complementizer. Verbs that allow wh+ complementizers, such as *ask*, can take *whether* and *if* as complementizers. The possible combinations of complementizers and clause types is summarized in Table 8.1.

As can be seen in Table 8.1, sentential subjects differ from sentential complements in requiring the complementizer *that* for all indicative and subjunctive clauses. In sentential complements, *that* often varies freely with a null complementizer, as illustrated in (14)-(19).

(14) Christy hopes that Mike wins.

(15) Christy hopes Mike wins.

(16) Dania thinks that Newt is a liar.

(17) Dania thinks Newt is a liar.

(18) That Helms won so easily annoyed me.

(19) *Helms won so easily annoyed me.

a complementizer.

[3] Other considerations, such as the relationship between the tense/aspect of the matrix clause and the tense/aspect of a complement clause are also important but are not currently addressed in the current English XTAG grammar.

Complementizer:		that	whether	if	for	null
Clause type						
indicative	subject	Yes	Yes	No	No	No
	complement	Yes	Yes	Yes	No	Yes
infinitive	subject	No	Yes	No	Yes	Yes
	complement	No	Yes	No	Yes	Yes
subjunctive	subject	Yes	No	No	No	No
	complement	Yes	No	No	No	Yes
gerundive[4]	complement	No	No	No	No	Yes
base	complement	No	No	No	No	Yes
small clause	complement	No	No	No	No	Yes

Table 8.1: Summary of Complementizer and Clause Combinations

Another fact which must be accounted for in the analysis is that in infinitival clauses, the complementizer *for* must appear with an overt subject NP, whereas a complementizer-less infinitival clause never has an overt subject, as shown in (20)-(23). (See section 8.5 for more discussion of the case assignment issues relating to this construction.)

(20) To lose would be awful.

(21) For Penn to lose would be awful.

(22) *For to lose would be awful.

(23) *Penn to lose would be awful.

In addition, some verbs select <**wh**>=+ complements (either questions or clauses with *whether* or *if*) ([Grimshaw, 1990]):

(24) Jesse wondered who left.

(25) Jesse wondered if Barry left.

(26) Jesse wondered whether to leave.

(27) Jesse wondered whether Barry left.

(28) *Jesse thought who left.

(29) *Jesse thought if Barry left.

(30) *Jesse thought whether to leave.

(31) *Jesse thought whether Barry left.

[4] Most gerundive phrases are treated as NP's. In fact, all gerundive subjects are treated as NP's, and the only gerundive complements which receive a sentential parse are those for which there is no corresponding NP parse. This was done to reduce duplication of parses. See Chapter 17 for further discussion of gerunds.

8.3 Features Required

As we have seen above, clauses may be <**wh**>=+ or <**wh**>=–, may have one of several complementizers or no complementizer, and can be of various clause types. The XTAG analysis uses three features to capture these possibilities: <**comp**> for the variation in complementizers, <**wh**> for the question vs. non-question alternation and <**mode**>[5] for clause types. In addition to these three features, the <**assign-comp**> feature represents complementizer requirements of the embedded verb. More detailed discussion of the <**assign-comp**> feature appears below in the discussions of sentential subjects and of infinitives. The four features and their possible values are shown in Table 8.2.

Feature	Values
<**comp**>	that, if, whether, for, rel, nil
<**mode**>	ind, inf, subjnt, ger, base, ppart, nom/prep
<**assign-comp**>	that, if, whether, for, rel, ind_nil, inf_nil
<**wh**>	+,–

Table 8.2: Summary of Relevant Features

8.4 Distribution of Complementizers

Like other non-arguments, complementizers anchor an auxiliary tree (shown in Figure 8.1) and adjoin to elementary clausal trees. The auxiliary tree for complementizers is the only alternative to having a complementizer position 'built into' every sentential tree. The latter choice would mean having an empty complementizer substitute into every matrix sentence and a complementizerless embedded sentence to fill the substitution node. Our choice follows the XTAG principle that initial trees consist only of the arguments of the anchor[6] – the S tree does not contain a slot for a complementizer, and the βCOMP tree has only one argument, an S with particular features determined by the complementizer. Complementizers select the type of clause to which they adjoin through constraints on the <**mode**> feature of the S foot node in the tree shown in Figure 8.1. These features also pass up to the root node, so that they are 'visible' to the tree where the embedded sentence adjoins/substitutes.

The grammar handles the following complementizers: *that*, *whether*, *if*, *for*, and no complementizer, and the clause types: indicative, infinitival, gerundive, past participial, subjunctive and small clause (**nom/prep**). The <**comp**> feature in a clausal tree reflects the value of the complementizer if one has adjoined to the clause.

The <**comp**> and <**wh**> features receive their root node values from the particular complementizer which anchors the tree. The βCOMPs tree adjoins to an S node with the feature <**comp**>=**nil**; this feature indicates that the tree does not already **have** a complementizer adjoined to it.[7] We ensure that there are no stacked complementizers by requiring the foot node

[5] <**mode**> actually conflates several types of information, in particular verb form and mood.

[6] See section 4.2 for a discussion of the difference between complements and adjuncts in the XTAG grammar.

[7] Because root S's cannot have complementizers, the parser checks that the root S has <**comp**>=**nil** at the end of the derivation, when the S is also checked for a tensed verb.

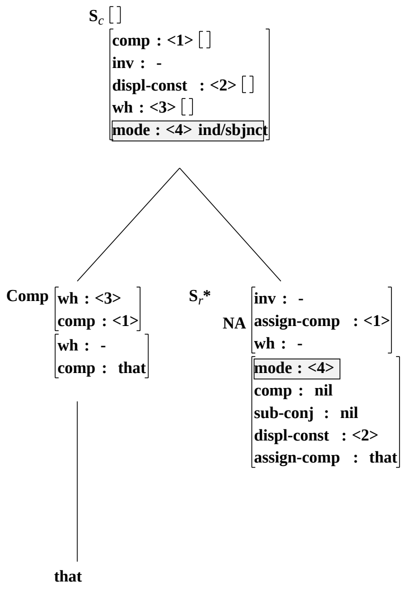


Figure 8.1: Tree βCOMPs, anchored by *that*

of βCOMPs to have <**comp**>=**nil**, as well as using the <**sub-conj**>=**nil** feature to prevent complementizers from adjoining above subordinating conjunctions.

8.5 Case assignment, *for* and the two *to*'s

The <**assign-comp**> feature is used to represent the requirements of particular types of clauses for particular complementizers. So while the <**comp**> feature represents constraints originating from the VP dominating the clause, the <**assign-comp**> feature represents constraints originating from the highest VP in the clause. <**assign-comp**> is used to control the appearance of subjects in infinitival clauses, to ensure the correct distribution of complementizers in sentential subjects, and to block 'that-trace' violations.

Examples (34), (35) and (36) show that an accusative case subject is obligatory in an infinitive clause if the complementizer *for* is present. The infinitive clauses in both (32) and (33) are analyzed in the English XTAG grammar as having PRO subjects. The apparent subject of *to win* in (32) is taken to be an object of the verb rather than the subject of the infinitive clause.

(32) Mike wants her to pass the exam.

(33) Christy wants to pass the exam.

(34) Mike wants for her to pass the exam.

(35) ∗Mike wants for she to pass the exam.

(36) ∗Christy wants for to pass the exam.

The *for-to* construction is particularly illustrative of the difficulties and benefits faced in using a lexicalized grammar. It is commonly accepted that *for* behaves as a case-assigning complementizer in this construction, assigning accusative case to the 'subject' of the clause since the infinitival verb does not assign case to its subject position. However, in our featurized grammar, the absence of a feature licenses anything, so we must have overt null case assigned by infinitives to ensure the correct distribution of PRO subjects. (See section 4.4 for more discussion of case assignment.) This null case assignment clashes with accusative case assignment if we simply add *for* as a standard complementizer, since NP's (including PRO) are drawn from the lexicon already marked for case. Thus, we must use the <**assign-comp**> feature to pass information about the verb up to the root of the embedded sentence. To capture these facts, two infinitive *to*'s are posited. One infinitive *to* has <**assign-case**>=**none** which forces a PRO subject, and <**assign-comp**>=**inf_nil** which prevents *for* from adjoining. The other infinitive *to* has no value at all for <**assign-case**> and has <**assign-comp**>=**for**, so that it can only occur with the complementizer *for*. In those instances *for* supplies the <**assign-case**> value and assigns accusative case to the overt subject.

8.6 Sentential Complements of Verbs

Tree families: Tnx0Vs1, Tnx0Vnx1s2, TItVnx1s2, TItVpnx1s2, TItVad1s2, Tnx0Ax1s2, Tnx0dxN1s1, Tnx0N1s1, Tnx0Pnx1s2, Tnx0Px1s2.

Verbs that select sentential complements restrict the <**mode**> and <**comp**> values for those complements. Since with very few exceptions[8] long distance extraction is possible from sentential complements, the S complement nodes are adjunction nodes. Figure 8.2 shows the declarative tree for sentential complements, anchored by *think*.

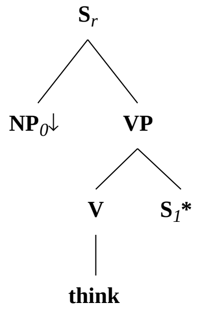


Figure 8.2: Sentential complement tree: βnx0Vs1

The need for an adjunction node rather than a substitution node at S_1 may not be obvious until one considers the derivation of sentences with long distance extractions. For example, the declarative in (37) is derived by adjoining the tree in Figure 8.3(b) to the S_1 node of the tree in Figure 8.3(a). Since there are no bottom features on S_1, the same final result could have been achieved with a substitution node at S_1.

(37) The emu thinks that the aardvark smells terrible.

[8] For example, long distance extraction is not possible from the S complement in it-clefts.

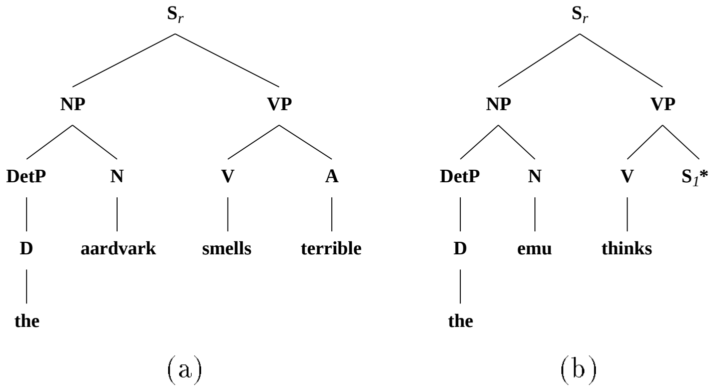


Figure 8.3: Trees for *The emu thinks that the aardvark smells terrible.*

However, adjunction is crucial in deriving sentences with long distance extraction, as in sentences (38) and (39).

(38) Who does the emu think smells terrible?

(39) Who did the elephant think the panda heard the emu say smells terrible?

The example in (38) is derived from the trees for *who smells terrible?* shown in Figure 8.4 and *the emu thinks* S shown in Figure 8.3(b), by adjoining the latter at the S_r node of the former.[9] This process is recursive, allowing sentences like (39). Such a representation has been shown by [Kroch and Joshi, 1985] to be well-suited for describing unbounded dependencies.

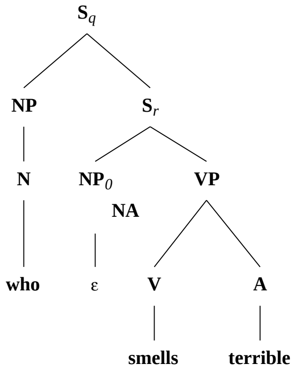


Figure 8.4: Tree for *Who smells terrible?*

In English, a complementizer may not appear on a complement with an extracted subject (the 'that-trace' configuration). This phenomenon is illustrated in (40)-(42):

(40) Which animal did the giraffe say that he likes?

(41) *Which animal did the giraffe say that likes him?

[9] See Chapter 20 for a discussion of do-support.

(42) Which animal did the giraffe say likes him?

These sentences are derived in XTAG by adjoining the tree for *did the giraffe say* S at the S_r node of the tree for either *which animal likes him* (to yield sentence (42)) or *which animal he likes* (to yield sentence (40)). That-trace violations are blocked by the presence of the feature **<assign-comp>=inf_nil/ind_nil** feature on the bottom of the S_r node of trees with extracted subjects, i.e. those used in sentences such as (41) and (42). This blocks (or 'filters') any other values of **<assign-comp>** projected by the verb, and ensures that no complementizer is able to adjoin at this node. Complementizers may adjoin normally to object extraction trees such as those used in sentence (40).

In the case of indirect questions, subjacency follows from the principle that a given tree cannot contain more than one wh-element. Extraction out of an indirect question is ruled out because a sentence like:

(43) $*$ Who_i do you wonder who_j e_j loves e_i ?

would have to be derived from the adjunction of *do you wonder* into $\textit{who}_i$ $\textit{who}_j$ e_j *loves* e_i, which is an ill-formed elementary tree.[10]

8.7 Sentential Subjects

Tree families: Ts0Vnx1, Ts0Ax1, Ts0dxN1, Ts0N1, Ts0Pnx1.

Verbs that select sentential subjects anchor trees that have an S node in the subject position rather than an NP node. Since extraction is not possible from sentential subjects, they are implemented as substitution nodes in the English XTAG grammar. Restrictions on sentential subjects, such as the required *that* complementizer for indicatives, are enforced by feature values specified on the S substitution node in the elementary tree.

Sentential subjects behave essentially like sentential complements, with a few exceptions. In general, all verbs which license sentential subjects license the same set of clause types. Thus, unlike sentential complement verbs which select particular complementizers and clause types, the matrix verbs licensing sentential subjects merely license the S argument. Information about the complementizer or embedded verb is located in the tree features, rather than in the features of each verb selecting that tree. Thus, all sentential subject trees have the same **<mode>**, **<comp>** and **<assign-comp>** values shown in Figure 8.5(a).

The major difference in clause types licensed by S-subjs and S-comps is that indicative S-subjs obligatorily have a complementizer (see examples in section 8.2). The **<assign-comp>** feature is used here to license a null complementizer for infinitival but not indicative clauses. **<assign-comp>** has the same possible values as **<comp>**, with the exception that the **nil** value is 'split' into **ind-nil** and **inf-nil**. This difference in feature values is illustrated in Figure 8.5.

Another minor difference is that *whether* but not *if* is grammatical with S-subjs.[11] Thus, **if** is not among the **<comp>** values allowed in S-subjs. The final difference from S-comps is that

[10] This does not mean that elementary trees with more than one gap should be ruled out across the grammar. Such trees might be required for dealing with parasitic gaps or gaps in coordinated structures.

[11] Some speakers also find *if* as a complementizer only marginally grammatical in S-comps.

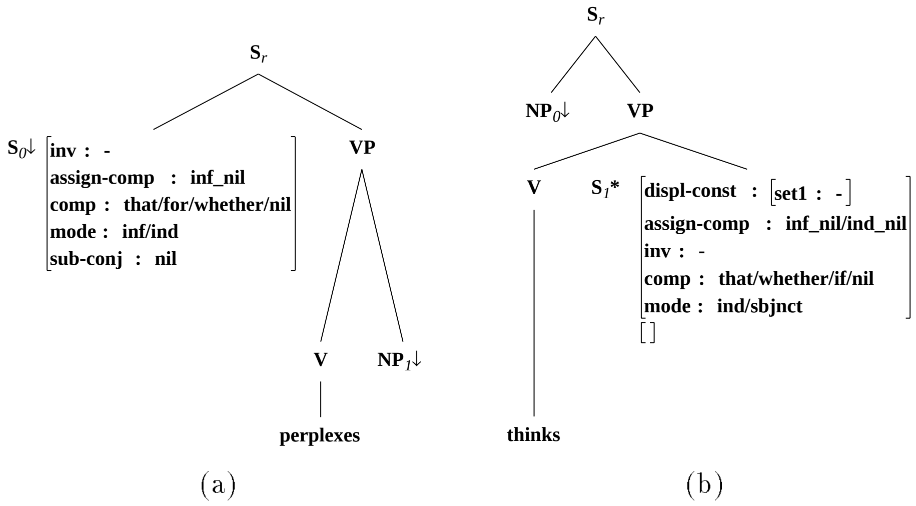


Figure 8.5: Comparison of <**assign-comp**> values for sentential subjects: αs0Vnx1 (a) and sentential complements: βnx0Vs1 (b)

there are no S-subjs with <**mode**>=**ger**. As noted in footnote 4 of this chapter, gerundive complements are only allowed when there is no corresponding NP parse. In the case of gerundive S-subjs, there is always an NP parse available.

8.8 Nouns and Prepositions taking Sentential Complements

Trees: αNXNs, αNXdxNs, αPXPs, βvxPs, βPss, βnxPx.

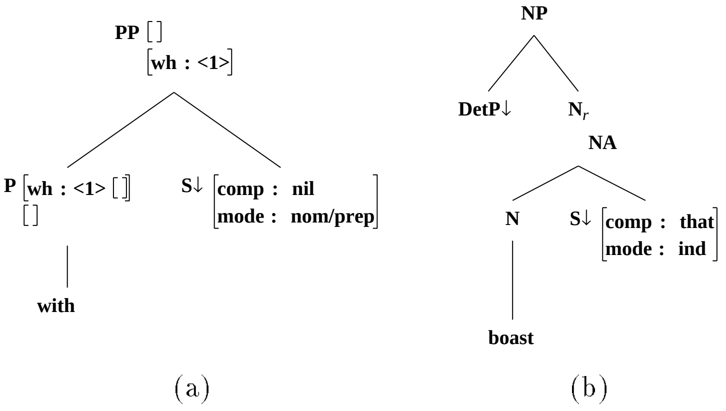


Figure 8.6: Sample trees for preposition: αPXPs (a) and noun: βNXdxNs (b) taking sentential complements

Prepositions and nouns can also select sentential complements, using the trees listed above. These trees use the <**mode**> and <**comp**> features as shown in Figure 8.6. For example, the noun *boast* takes only indicative complements with *that*, while the preposition *with* takes small clause complements, as seen in sentences (44)-(47).

(44) Beth's boast that Clove was a smart dog....

(45) *Beth's boast that Clove a smart dog....

(46) Dania wasn't getting any sleep with Doug sick.

(47) *Dania wasn't getting any sleep with Doug was sick.

Chapter 9

The English Copula, Raising Verbs, and Small Clauses

The English copula, raising verbs, and small clauses are all handled in XTAG by a common analysis based on sentential clauses headed by non-verbal elements. Since there are a number of different analyses in the literature of how these phenomena are related (or not), we will present first the data for all three phenomena, then various analyses from the literature, finishing with the analysis used in the English XTAG grammar.[1]

9.1 Usages of the copula, raising verbs, and small clauses

9.1.1 Copula

The verb *be* as used in sentences (48)-(50) is often referred to as the COPULA. It can be followed by a noun, adjective, or prepositional phrase.

(48) Carl is a jerk.

(49) Carl is upset.

(50) Carl is in a foul mood.

Although the copula may look like a main verb at first glance, its syntactic behavior follows the auxiliary verbs rather than main verbs. In particular,

- Copula *be* inverts with the subject.

  (51) Is Beth writing her dissertation?
  Is Beth upset?
  $*$Wrote Beth her dissertation?

- Copula *be* occurs to the left of the negative marker *not*.

[1] This chapter is strongly based on [Heycock, 1991]. Sections 9.1 and 9.2 are greatly condensed from her paper, while the description of the XTAG analysis in section 9.3 is an updated and expanded version.

(52) Beth is not writing her dissertation.
Beth is not upset.
*Beth wrote not her dissertation.

- Copula *be* can contract with the negative marker *not.*

(53) Beth isn't writing her dissertation.
Beth isn't upset.
*Beth wroten't her dissertation.

- Copula *be* can contract with pronominal subjects.

(54) She's writing her dissertation.
She's upset.
*She'ote her dissertation.

- Copula *be* occurs to the left of adverbs in the unmarked order.

(55) Beth is often writing her dissertation.
Beth is often upset.
*Beth wrote often her dissertation.

Unlike all the other auxiliaries, however, copula *be* is not followed by a verbal category (by definition) and therefore must be the rightmost verb. In this respect, it is like a main verb.

The semantic behavior of the copula is also unlike main verbs. In particular, any semantic restrictions or roles placed on the subject come from the complement phrase (NP, AP, PP) rather than from the verb, as illustrated in sentences (56) and (57). Because the complement phrases predicate over the subject, these types of sentences are often called PREDICATIVE sentences.

(56) The bartender was garrulous.

(57) ?The cliff was garrulous.

9.1.2 Raising Verbs

Raising verbs are the class of verbs that share with the copula the property that the complement, rather than the verb, places semantic constraints on the subject.

(58) Carl seems a jerk.
Carl seems upset.
Carl seems in a foul mood.

(59) Carl appears a jerk.
Carl appears upset.
Carl appears in a foul mood.

The raising verbs are similar to auxiliaries in that they order with other verbs, but they are unique in that they can appear to the left of the infinitive, as seen in the sentences in (60). They cannot, however, invert or contract like other auxiliaries (61), and they appear to the right of adverbs (62).

(60) Carl seems to be a jerk.
Carl seems to be upset.
Carl seems to be in a foul mood.

(61) *Seems Carl to be a jerk?
*Carl seemn't to be upset.
*Carl'ems to be in a foul mood.

(62) Carl often seems to be upset.
*Carl seems often to be upset.

9.1.3 Small Clauses

One way of describing small clauses is as predicative sentences without the copula. Since matrix clauses require tense, these clausal structures appear only as embedded sentences. They occur as complements of certain verbs, each of which may allow one type of small clause but not another, depending on its lexical idiosyncrasies.

(63) I consider [Carl a jerk].
I consider [Carl upset].
?I consider [Carl in a foul mood].

(64) I prefer [Carl in a foul mood].
??I prefer [Carl upset].

9.2 Various Analyses

9.2.1 Main Verb Raising to INFL + Small Clause

In [Pollock, 1989] the copula is generated as the head of a VP, like any main verb such as *sing* or *buy*. Unlike all other main verbs[2], however, *be* moves out of the VP and into Infl in a tensed sentence. This analysis aims to account for the behavior of *be* as an auxiliary in terms of inversion, negative placement and adverb placement, while retaining a sentential structure in which *be* heads the main VP at D-Structure and can thus be the only verb in the clause.

Pollock claims that the predicative phrase is not an argument of *be*, which instead he assumes to take a small clause complement, consisting of a node dominating an NP and a predicative AP, NP or PP. The subject NP of the small clause then raises to become the subject of the sentence. This accounts for the failure of the copula to impose any selectional restrictions on the subject. Raising verbs such as *seem* and *appear*, presumably, take the same type of small clause complement.

9.2.2 Auxiliary + Null Copula

In [Lapointe, 1980] the copula is treated as an auxiliary verb that takes as its complement a VP headed by a passive verb, a present participle, or a null verb (the true copula). This verb may

[2] with the exception of *have* in British English. See footnote 1 in Chapter 20.

then take AP, NP or PP complements. The author points out that there are many languages that have been analyzed as having a null copula, but that English has the peculiarity that its null copula requires the co-presence of the auxiliary *be*.

9.2.3 Auxiliary + Predicative Phrase

In GPSG ([Gazdar *et al.*, 1985], [Sag *et al.*, 1985]) the copula is treated as an auxiliary verb that takes an X^2 category with a + value for the head feature [PRD] (predicative). AP, NP, PP and VP can all be [+PRD], but a Feature Co-occurrence Restriction guarantees that a [+PRD] VP will be headed by a verb that is either passive or a present participle.

GPSG follows [Chomsky, 1970] in adopting the binary valued features [V] and [N] for decomposing the verb, noun, adjective and preposition categories. In that analysis, verbs are [+V,−N], nouns are [−V,+N], adjectives [+V,+N] and prepositions [−V,−N]. NP and AP predicative complements generally pattern together; a fact that can be stated economically using this category decomposition. In neither [Sag *et al.*, 1985] nor [Chomsky, 1970] is there any discussion of how to handle the complete range of complements to a verb like *seem*, which takes AP, NP and PP complements, as well as infinitives. The solution would appear to be to associate the verb with two sets of rules for small clauses, leaving aside the use of the verb with an expletive subject and sentential complement.

9.2.4 Auxiliary + Small Clause

In [Moro, 1990] the copula is treated as a special functional category - a lexicalization of tense, which is considered to head its own projection. It takes as a complement the projection of another functional category, Agr (agreement). This projection corresponds roughly to a small clause, and is considered to be the domain within which predication takes place. An NP must then raise out of this projection to become the subject of the sentence: it may be the subject of the AgrP, or, if the predicate of the AgrP is an NP, this may raise instead. In addition to occurring as the complement of *be*, AgrP is selected by certain verbs such as *consider*. It follows from this analysis that when the complement to *consider* is a simple AgrP, it will always consist of a subject followed by a predicate, whereas if the complement contains the verb *be*, the predicate of the AgrP may raise to the left of *be*, leaving the subject of the AgrP to the right.

(65) John$_i$ is [$_{AgrP}$ t_i the culprit].

(66) The culprit$_i$ is [$_{AgrP}$ John t_i].

(67) I consider [$_{AgrP}$ John the culprit].

(68) I consider [John$_i$ to be [$_{AgrP}$ t_i the culprit]].

(69) I consider [the culprit$_i$ to be [$_{AgrP}$ John t_i]].

Moro does not discuss a number of aspects of his analysis, including the nature of Agr and the implied existence of sentences without VP's.

9.3 XTAG analysis

The XTAG grammar provides a uniform analysis for the copula, raising verbs and small clauses by treating the maximal projections of lexical items that can be predicated as predicative clauses, rather than simply noun, adjective and prepositional phrases. The copula adjoins in for matrix clauses, as do the raising verbs. Certain other verbs (such as *consider*) can take the predicative clause as a complement, without the adjunction of the copula, to form the embedded small clause.

The structure of a predicative clause, then, is roughly as seen in (70)-(72) for NP's, AP's and PP's. The XTAG trees corresponding to these structures[3] are shown in Figures 9.1(a), 9.1(b), and 9.1(c), respectively.

(70) $[_S$ NP $[_{VP}$ N $\ldots]]$

(71) $[_S$ NP $[_{VP}$ A $\ldots]]$

(72) $[_S$ NP $[_{VP}$ P $\ldots]]$

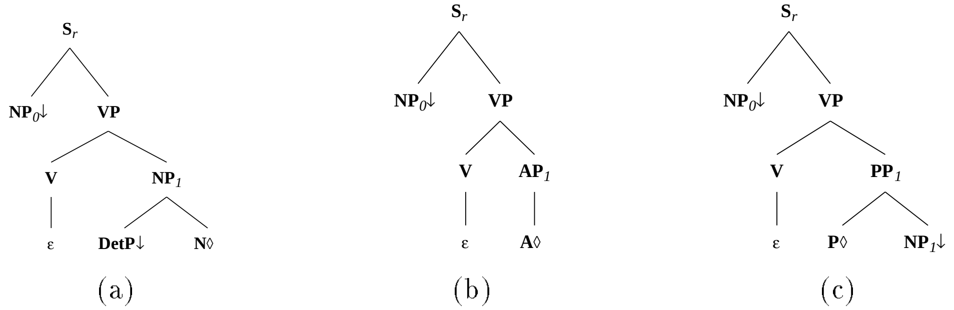


Figure 9.1: Predicative trees: αnx0dxN1 (a), αnx0Ax1 (b) and αnx0Pnx1 (c)

The copula *be* and raising verbs all get the basic auxiliary tree as explained in the section on auxiliary verbs (section 20.1). Unlike the raising verbs, the copula also selects the inverted auxiliary tree set. Figure 9.2 shows the basic auxiliary tree anchored by the copula *be*. The <**mode**> feature is used to distinguish the predicative constructions so that only the copula and raising verbs adjoin onto the predicative trees. The auxiliary tree in Figure 9.2 will look the same when anchored by the raising verbs, except possibly for the value of <**mode**> on the foot node, VP.

There are two possible values of <**mode**> that correspond to the predicative trees, **nom** and **prep**. They correspond to a modified version of the four-valued [N,V] feature described in section 9.2.3. The **nom** value corresponds to [N+], selecting the NP and AP predicative clauses. As mentioned earlier, they often pattern together with respect to constructions using predicative clauses. The remaining prepositional phrase predicative clauses, then, correspond to the **prep** mode.

[3] There are actually two other predicative trees in the XTAG grammar. Another predicative noun phrase tree is needed for noun phrases without determiners, as in the sentence *They are firemen*, and another prepositional phrase tree is needed for exhaustive prepositional phrases, such as *The workers are below.*

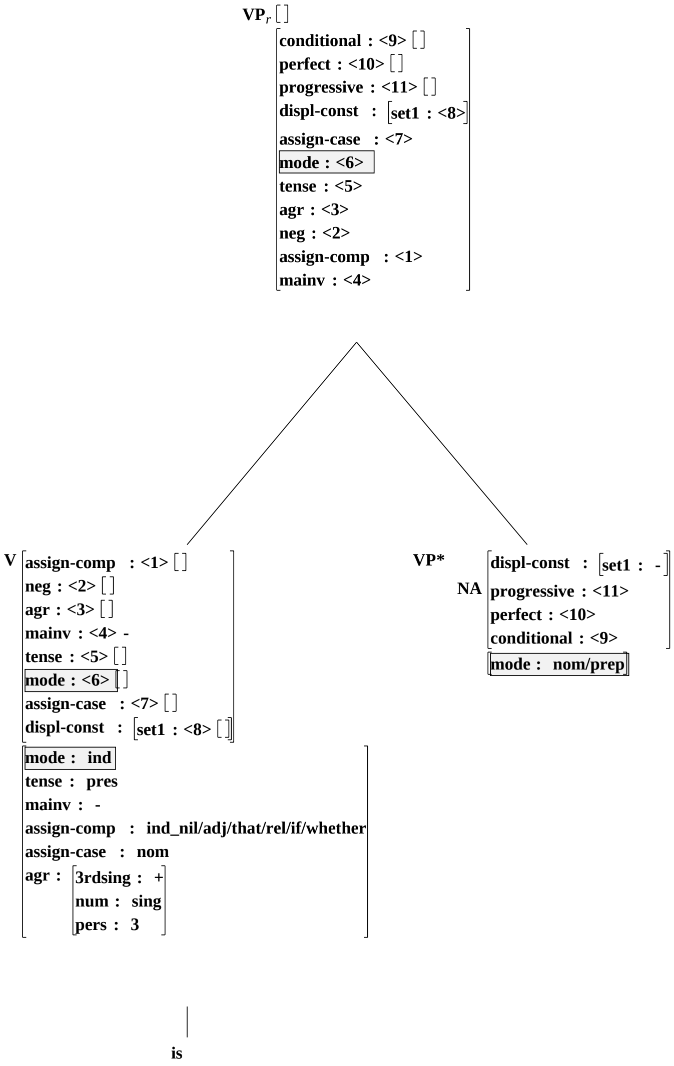


Figure 9.2: Copula auxiliary tree: βVvx

Figure 9.3 shows the predicative adjective tree from Figure 9.1(b) now anchored by *upset* and with the features visible. As mentioned, <**mode**>=**nom** on the VP node prevents auxiliaries other than the copula or raising verbs from adjoining into this tree. In addition, it prevents the predicative tree from occurring as a matrix clause. Since all matrix clauses in XTAG must be mode indicative (**ind**) or imperative (**imp**), a tree with <**mode**>=**nom** or <**mode**>=**prep** must have an auxiliary verb (the copula or a raising verb) adjoin in to make it <**mode**>=**ind**.

The distribution of small clauses as embedded complements to some verbs is also managed through the mode feature. Verbs such as *consider* and *prefer* select trees that take a sentential complement, and then restrict that complement to be <**mode**>=**nom** and/or <**mode**>=**prep**, depending on the lexical idiosyncrasies of that particular verb. Many verbs that don't take small clause complements do take sentential complements that are <**mode**>=**ind**,

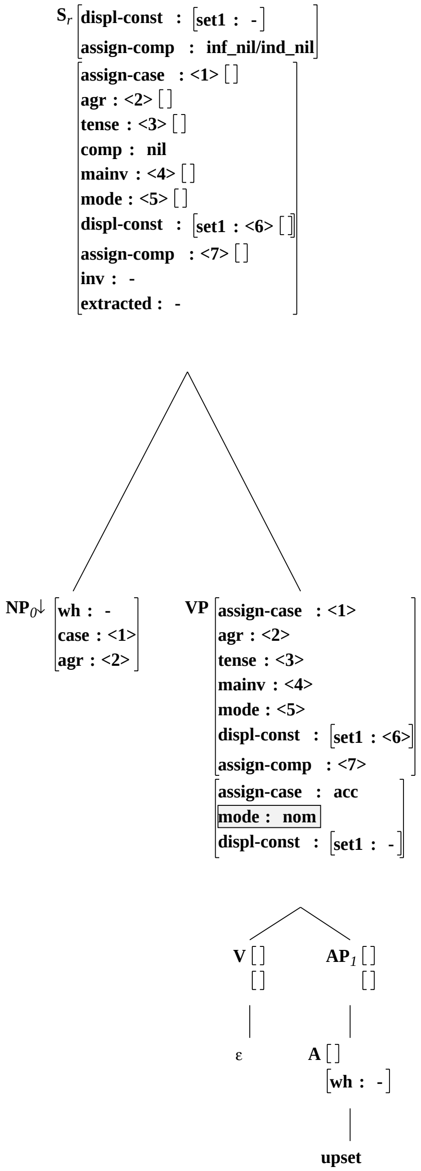


Figure 9.3: Predicative AP tree with features: αnx0Ax1

which includes small clauses with the copula already adjoined. Hence, as seen in sentence sets (73)-(75), *consider* takes only small clause complements, *prefer* takes both **prep** (but not **nom**) small clauses and indicative clauses, while *feel* takes only indicative clauses.

(73) She considers Carl a jerk.
?She considers Carl in a foul mood.
∗She considers that Carl is a jerk.

(74) ∗She prefers Carl a jerk.

She prefers Carl in a foul mood.
She prefers that Carl is a jerk.

(75) *She feels Carl a jerk.
*She feels Carl in a foul mood.
She feels that Carl is a jerk.

Figure 9.4 shows the tree anchored by *consider* that takes the predicative small clauses.

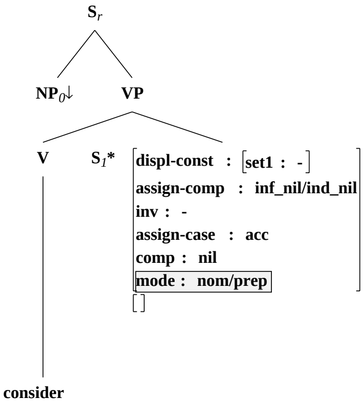


Figure 9.4: *Consider* tree for embedded small clauses

9.4 Non-predicative *BE*

The examples with the copula that we have given seem to indicate that *be* is always followed by a predicative phrase of some sort. This is not the case, however, as seen in sentences such as (76)-(81). The noun phrases in these sentences are not predicative. They do not take raising verbs, and they do not occur in embedded small clause constructions.

(76) My teacher is Mrs. Wayman.

(77) Doug is the man with the glasses.

(78) *My teacher seems Mrs. Wayman.

(79) *Doug appears the man with the glasses.

(80) *I consider [my teacher Mrs. Wayman].

(81) *I prefer [Doug the man with the glasses].

In addition, the subject and complement can exchange positions in these type of examples but not in sentences with predicative *be*. Sentence (82) has the same interpretation as sentence (77) and differs only in the positions of the subject and complement NP's. Similar sentences, with a predicative *be*, are shown in (83) and (84). In this case, the sentence with the exchanged NP's (84) is ungrammatical.

(82) The man with the glasses is Doug.

(83) Doug is a programmer.

(84) $*$A programmer is Doug.

The non-predicative *be* in (76) and (77), also called EQUATIVE BE, patterns differently, both syntactically and semantically, from the predicative usage of *be*. Since these sentences are clearly not predicative, it is not desirable to have a tree structure that is anchored by the NP, AP, or PP, as we have in the predicative sentences. In addition to the conceptual problem, we would also need a mechanism to block raising verbs from adjoining into these sentences (while allowing them for true predicative phrases), and prevent these types of sentence from being embedded (again, while allowing them for true predicative phrases).

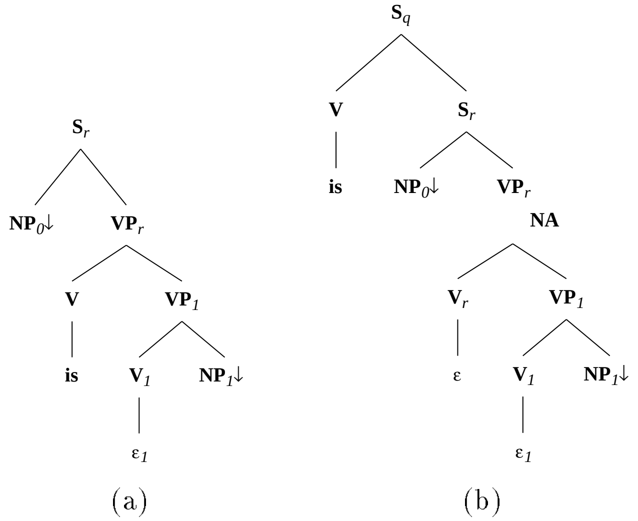


Figure 9.5: Equative *BE* trees: αnx0BEnx1 (a) and αInvnx0BEnx1 (b)

Although non-predicative *be* is not a raising verb, it does exhibit the auxiliary verb behavior set out in section 9.1.1. It inverts, contracts, and so forth, as seen in sentences (85) and (86), and therefore can not be associated with any existing tree family for main verbs. It requires a separate tree family that includes the tree for inversion. Figures 9.5(a) and 9.5(b) show the declarative and inverted trees, respectively, for equative *be*.

(85) Is my teacher Mrs. Wayman?

(86) Doug isn't the man with the glasses.

Chapter 10

Ditransitive constructions and dative shift

Verbs such as *give* and *put* that require two objects, as shown in examples (87)-(90), are termed ditransitive.

(87) Christy gave a cannoli to Beth Ann.

(88) ∗Christy gave Beth Ann.

(89) Christy put a cannoli in the refrigerator.

(90) ∗Christy put a cannoli.

The indirect objects *Beth Ann* and *refrigerator* appear in these examples in the form of PP's. Within the set of ditransitive verbs there is a subset that also allow two NP's as in (91). As can be seen from (91) and (92) this two NP, or double-object, construction is grammatical for *give* but not for *put*.

(91) Christy gave Beth Ann a cannoli

(92) ∗Christy put the refrigerator the cannoli.

The alternation between (87) and (91) is known as dative shift.[1] In order to account for verbs with dative shift the English XTAG grammar includes structures for both variants in the tree family Tnx0Vnx1Pnx2. The declarative trees for the shifted and non-shifted alternations are shown in Figure 10.1.

The indexing of nodes in these two trees represents the fact that the semantic role of the indirect object (NP_2) in Figure 10.1(a) is the same as that of the direct object (NP_2) in Figure 10.1(b) (and vice versa). This use of indexing is consistent with our treatment of other constructions such as passive and ergative.

Verbs that do not show this alternation and have only the NP PP structure (e.g. *put*) select the tree family Tnx0Vnx1pnx2. Unlike the Tnx0Vnx1Pnx2 family, the Tnx0Vnx1pnx2 tree family does not contain trees for the NP NP structure. Other verbs such as *ask* allow only the NP NP structure as shown in (93) and (94).

[1] In languages similar to English that have overt case marking indirect objects would be marked with dative case. It has also been suggested that for English the preposition *to* serves as a dative case marker.

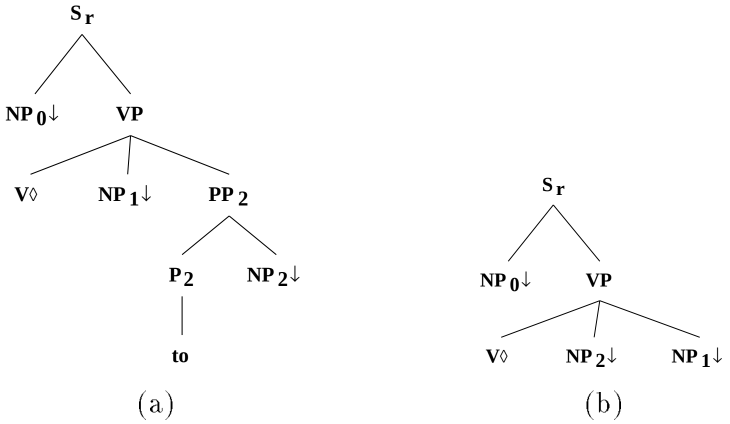


Figure 10.1: Dative shift trees: αnx0Vnx1Pnx2 (a) and αnx0Vnx2nx1 (b)

(93) Beth Ann asked Srini a question.

(94) ∗Beth Ann asked a question to Srini.

Verbs that only allow the NP NP structure select the tree family Tnx0Vnx1nx2. This tree family does not have the trees for the NP PP structure.

Notice that in Figure 10.1(a) the preposition *to* is built into the tree. There are other apparent cases of dative shift with *for*, such as in (95) and (96), that we have taken to be structurally distinct from the cases with *to*.

(95) Beth Ann baked Dusty a biscuit.

(96) Beth Ann baked a biscuit for Dusty.

[McCawley, 1988] notes:

> A "*for-dative*" expression in underlying structure is external to the V with which it is combined, in view of the fact that the latter behaves as a unit with regard to all relevant syntactic phenomena.

In other words, the *for* PP's that appear to undergo dative shift are actually adjuncts, not complements. Examples (97) and (98) demonstrate that PP's with *for* are optional while ditransitive *to* PP's are not.

(97) Beth Ann made dinner.

(98) ∗Beth Ann gave dinner.

Consequently, in the XTAG grammar the apparent dative shift with *for* PP's is treated as Tnx0Vnx1nx2 for the NP NP structure, and as a transitive plus an adjoined adjunct PP for the NP PP structure. Hence the fact that the preposition *to* is built into the tree and is the only preposition allowed in dative shift constructions correctly accounts for the observed patterns.

[McCawley, 1988] also notes that the *to* and *for* cases differ with respect to passivization; the indirect objects with *to* may be the subjects of corresponding passives while the alleged indirect objects with *for* can not, as in sentences (99)-(102).

(99) Beth Ann gave Clove dinner.

(100) Clove was given dinner (by Beth Ann).

(101) Beth Ann made Clove dinner.

(102) ?Clove was made dinner (by Beth Ann).

However, we believe that this to be incorrect, and that the indirect objects in the *for* case are allowed to be the subjects of passives, as in sentences (103)-(104). The apparent strangeness of sentence (102) is caused by interference from other interpretations of *Clove was made dinner.*

(103) Dania baked Doug a cake.

(104) Doug was baked a cake by Dania.

Chapter 11

It-clefts

There are several varieties of it-clefts in English. All of the it-clefts have four major components:

- **the dummy subject:** *it*,
- **the main verb:** *be*,
- **the clefted element:** A constituent (XP) compatible with any gap in the clause,
- **the clause:** A clause (e.g. S) with or without a gap.

Examples of it-clefts are shown in (105)-(108).

(105) It was $[_{XP}$ here $_{XP}]$ $[_{S}$ that the ENIAC was created. $_{S}]$

(106) It was $[_{XP}$ at MIT $_{XP}]$ $[_{S}$ that colorless green ideas slept furiously. $_{S}]$

(107) It is $[_{XP}$ happily $_{XP}]$ $[_{S}$ that Seth quit Reality. $_{S}]$

(108) It was $[_{XP}$ there $_{XP}]$ $[_{S}$ that she would have to enact her renunciation. $_{S}]$

The clefted element can be of a number of categories, for example NP, PP or adverb. The clause can also be of several types. The English XTAG grammar currently has a separate analysis for only a subset of the 'specificational' it-clefts[1], in particular the ones without gaps in the clause (e.g. (107) and (108)). It-clefts that have gaps in the clause, such as (105) and (106) are currently handled as relative clauses. Although arguments have been made against treating the clefted element and the clause as a constituent ([Delahunty, 1984]), the relative clause approach does capture the restriction that the clefted element must fill the gap in the clause, and does not require any additional trees.

In the 'specificational' it-cleft without gaps in the clause, the clefted element has the role of an adjunct with respect to the clause. For these cases the English XTAG grammar requires additional trees. These it-cleft trees are in separate tree families because, although some researchers (e.g. [Akmajian, 1970]) derived it-clefts through movement from other sentence types, most current researchers (e.g. [Delahunty, 1984], [Knowles, 1986], [Gazdar *et al.*, 1985],

[1] See e.g. [Ball, 1991], [Delin, 1989] and [Delahunty, 1984] for more detailed discussion of types of it-clefts.

[Delin, 1989] and [Sornicola, 1988]) favor base-generation of the various cleft sentences. Placing the it-cleft trees in their own tree families is consistent with the the current preference for base generation, since in the XTAG English grammar, structures that would be related by transformation in a movement-based account will appear in the same tree family. Like the base-generated approaches, the placement of it-clefts in separate tree families makes the claim that there is no derivational relation between it-clefts and other sentence types.

The three it-cleft tree families are virtually identical except for the category label of the clefted element. Figure 11.1 shows the declarative tree and an inverted tree for the PP It-cleft tree family.

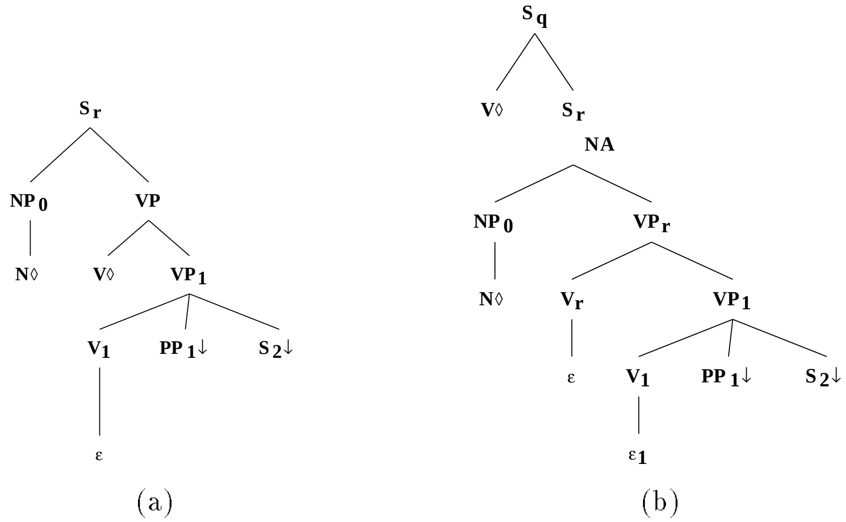


Figure 11.1: It-cleft with PP clefted element: αItVpnx1s2 (a) and αInvItVpnx1s2 (b)

The extra layer of tree structure in the VP represents that, while *be* is a main verb rather than an auxiliary in these cases, it retains some auxiliary properties. The VP structure for the equative/it-cleft-*be* is identical to that obtained after adjunction of predicative-*be* into small-clauses.[2] The inverted tree in Figure 11.1(b) is necessary because of *be*'s auxiliary-like behavior. Although *be* is the main verb in it-clefts, it inverts like an auxiliary. Main verb inversion cannot be accomplished by adjunction as is done with auxiliaries and therefore must be built into the tree family. The tree in Figure 11.1(b) is used for yes/no questions such as (109).

(109) Was it in the forest that the wolf talked to the little girl?

[2] For additional discussion of equative or predicative-*be* see Chapter 9.

Part III

Sentence Types

Chapter 12

Passives

In passive constructions such as (110), the subject NP is interpreted as having the same role as the direct object NP in the corresponding declarative (111).

(110) **An airline buy-out bill** was approved by the House. (WSJ)

(111) The House approved **an airline buy-out bill**.

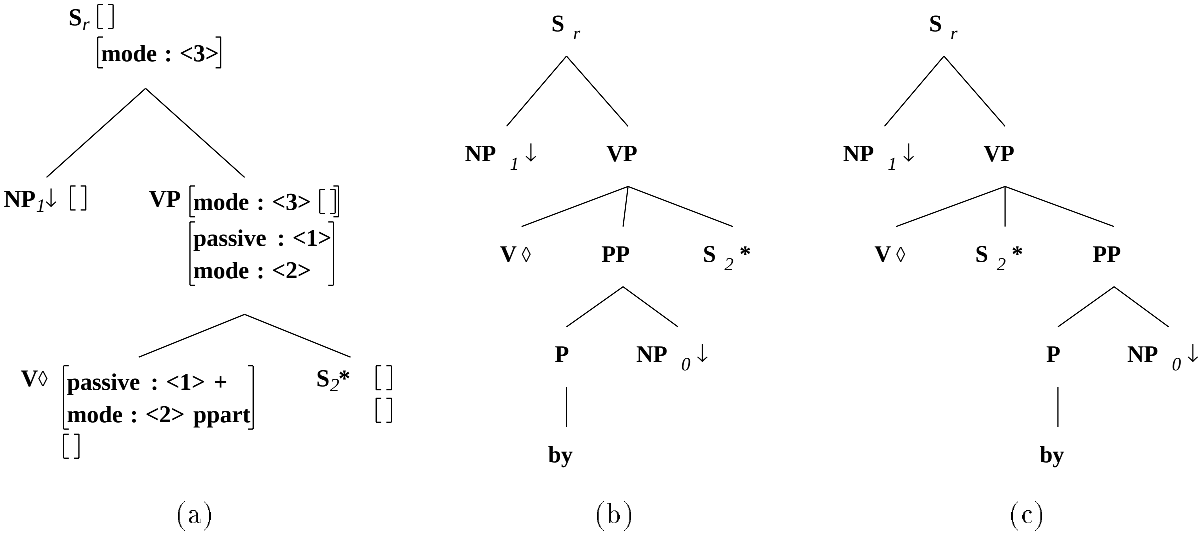


Figure 12.1: Passive trees in the Sentential Complement with NP tree family: βnx1Vs2 (a), βnx1Vbynx0s2 (b) and βnx1Vs2bynx0 (c)

In a movement analysis, the direct object is said to have moved to the subject position. The original declarative subject is either absent in the passive or is in a *by* headed PP (*by* phrase). In the English XTAG grammar, passive constructions are handled by having separate trees within the appropriate tree families. Passive trees are found in most tree families that have a direct object in the declarative tree (the light verb tree families, for instance, do not contain passive trees). Passive trees occur in pairs - one tree with the *by* phrase, and another without it. Variations in the location of the *by* phrase are possible if a subcategorization includes other arguments such as a PP or an indirect object. Additional trees are required for these variations. For example, the Sentential Complement with NP tree family has three passive trees, shown in Figure 12.1: one without the *by*-phrase (Figure 12.1(a)), one with the *by* phrase before the sentential complement (Figure 12.1(b)), and one with the *by* phrase after the sentential complement (Figure 12.1(c)).

Figure 12.1(a) also shows the feature restrictions imposed on the anchor[1]. Only verbs with **<mode>=ppart** (i.e. verbs with passive morphology) can anchor this tree. The **<mode>** feature is also responsible for requiring that passive *be* adjoin into the tree to create a matrix sentence. Since a requirement is imposed that all matrix sentences must have **<mode>=ind/imp**, an auxiliary verb that selects **<mode>=ppart** and **<passive>=+** (such as *was*) must adjoin (see Chapter 20 for more information on the auxiliary verb system).

[1] A reduced set of features are shown for readability.

Chapter 13

Extraction

The discussion in this chapter covers constructions that are analyzed as having wh-movement in GB, in particular, wh-questions and topicalization. Relative clauses, which could also be considered extractions, are discussed in Chapter 14.

Extraction involves a constituent appearing in a linear position to the left of the clause with which it is interpreted. One clause argument position is empty. For example, the position filled by *frisbee* in the declarative in sentence (112) is empty in sentence (113). The wh-item *what* in sentence (113) is of the same syntactic category as *frisbee* in sentence (112) and fills the same role with respect to the subcategorization.

(112) Clove caught a frisbee.

(113) What$_i$ did Clove catch ϵ_i?

The English XTAG grammar represents the connection between the extracted element and the empty position with co-indexing (as does GB). The <**trace**> feature is used to implement the co-indexing. In extraction trees in XTAG, the 'empty' position is filled with an ϵ. The extracted item always appears in these trees as a sister to the the S$_r$ tree, with both dominated by a S$_q$ root node. The S$_r$ subtrees in extraction trees have the same structure as the declarative tree in the same tree family. The additional structure in extraction trees of the S$_q$ and NP nodes roughly corresponds to the CP and Spec of CP positions in GB.

All sentential trees with extracted components (this does not include relative clause trees) are marked <**extracted**>=+ at the top S node, while sentential trees with no extracted components are marked <**extracted**>=–. Items that take embedded sentences, such as nouns, verbs and some prepositions can place restrictions on whether the embedded sentence is allowed to be extracted or not. For instance, sentential subjects and sentential complements of nouns and prepositions are not allowed to be extracted, while certain verbs may allow extracted sentential complements and others may not (e.g. sentences (114)-(117)).

(114) The jury wondered [who killed Nicole].

(115) The jury wondered [who Simpson killed].

(116) The jury thought [Simpson killed Nicole].

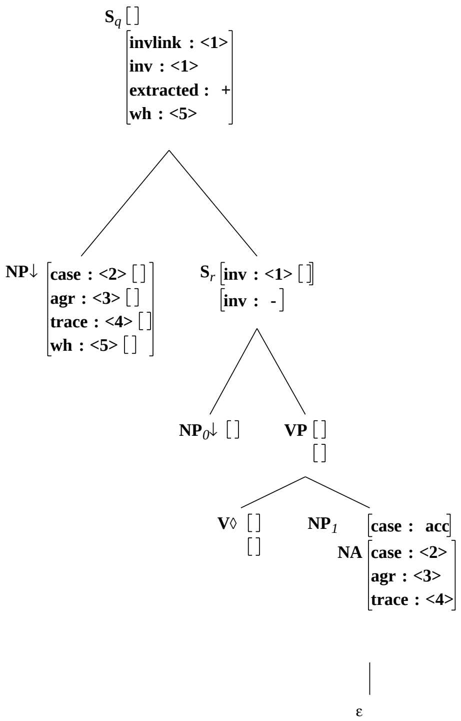


Figure 13.1: Transitive tree with object extraction: αW1nx0Vnx1

(117) ∗The jury thought [who did Simpson kill]?

The tree that is used to derive (115) in the English XTAG grammar is shown in Figure 13.1[1]. The important features of extracted trees are:

- The subtree that has S_r as its root is identical to the declarative tree or a non-extracted passive tree, except for having one NP position in the VP filled by ϵ.
- The root S node is S_q, which dominates NP and S_r.
- The <**trace**> feature of the ϵ filled NP is co-indexed with the <**trace**> feature of the NP daughter of S_q.
- The <**case**> and <**agr**> features are passed from the empty NP to the extracted NP. This is particularly important for extractions from subject NP's, since <**case**> can continue to be assign from the verb to the subject NP position, and from there passed to the extracted NP.
- The <**inv**> feature of S_r is co-indexed to the <**wh**> feature of NP through the use of the <**invlink**> feature in order to force subject-auxiliary inversion where needed (see section 13.1 for more discussion of the <**inv**>/<**wh**> co-indexing and the use of these trees for topicalization).

[1] Features not pertaining to this discussion have been taken out to improve readability.

13.1 Topicalization and the value of the <inv> feature

Our analysis of topicalization uses the same trees as wh-extraction. For any NP complement position a single tree is used for both wh-questions and for topicalization from that position. Wh-questions have subject-auxiliary inversion and topicalizations do not. This difference between the constructions is captured by equating the values of the S_r's <**inv**> feature and the extracted NP's <**wh**> feature. This means that if the extracted item is a wh-expression, as in wh-questions, the value of <**inv**> will be + and an inverted auxiliary will be forced to adjoin. If the extracted item is a non-wh, <**inv**> will be – and no auxiliary adjunction will occur. An additional complication is that inversion only occurs in matrix clauses, so the values of <**inv**> and <**wh**> should only be equated in matrix clauses and not in embedded clauses. In the English XTAG grammar, appropriate equating of the <**inv**> and <**wh**> features is accomplished using the <**invlink**> feature and a restriction imposed on the root S of a derivation. In particular, in extraction trees that are used for both wh-questions and topicalizations, the value of the <**inv**> feature for the top of the S_r node is co-indexed to the value of the <**inv**> feature on the bottom of the S_q node. On the bottom of the S_q node the <**inv**> feature is co-indexed to the <**invlink**> feature. The <**wh**> feature of the extracted NP node is co-indexed to the value of the <**wh**> feature on the bottom of S_q. The linking between the value of the S_q <**wh**> and the <**invlink**> features is imposed by a condition on the final root node of a derivation (i.e. the top S node of a matrix clause) requires that <**invlink**>=<**wh**>. For example, the tree in Figure 13.1 is used to derive both (118) and (119).

(118) John, I like.

(119) Who do you like?

For the question in (119), the extracted item *who* has the feature value <**wh**>=+, so the value of the <**inv**> feature on VP is also + and an auxiliary, in this case *do*, is forced to adjoin. For the topicalization (118) the values for *John*'s <**wh**> feature and for S_q's <**inv**> feature are both – and no auxiliary adjoins.

13.2 Extracted subjects

The extracted subject trees provide for sentences like (120)-(122), depending on the tree family with which it is associated.

(120) Who left?

(121) Who wrote the paper?

(122) Who was happy?

Wh-questions on subjects differ from other argument extractions in not having subject-auxiliary inversion. This means that in subject wh-questions the linear order of the constituents is the same as in declaratives so it is difficult to tell whether the subject has moved out of position or not (see [Heycock and Kroch, 1993] for arguments for and against moved subject).

The English XTAG treatment of subject extractions assumes the following:

- Syntactic subject topicalizations don't exist; and
- Subjects in wh-questions are extracted rather than in situ.

The assumption that there is no syntactic subject topicalization is reasonable in English since there is no convincing syntactic evidence and since the interpretability of subjects as topics seems to be mainly affected by discourse and intonational factors rather than syntactic structure. As for the assumption that wh-question subjects are extracted, these questions seem to have more similarities to other extractions than to the two cases in English that have been considered in situ wh: multiple wh questions and echo questions. In multiple wh questions such as sentence (123), one of the wh-items is blocked from moving sentence initially because the first wh-item already occupies the location to which it would move.

(123) Who ate what?

This type of 'blocking' account is not applicable to subject wh-questions because there is no obvious candidate to do the blocking. Similarity between subject wh-questions and echo questions is also lacking. At least one account of echo questions ([Hockey, 1994]) argues that echo questions are not ordinary wh-questions at all, but rather focus constructions in which the wh-item is the focus. Clearly, this is not applicable to subject wh-questions. So it seems that treating subject wh-questions similarly to other wh-extractions is more justified than an in situ treatment.

Given these assumptions, there must be separate trees in each tree family for subject extractions. The declarative tree cannot be used even though the linear order is the same because the structure is different. Since topicalizations are not allowed, the <**wh**> feature for the extracted NP node is set in these trees to +. The lack of subject-auxiliary inversion is handled by the absence of the <**invlink**> feature. Without the presence of this feature, the <**wh**> and <**inv**> are never linked, so inversion can not occur. Like other wh-extractions, the S_q node is marked <**extracted**>=+ to constrain the occurrence of these trees in embedded sentences. The tree in Figure 13.2 is an example of a subject wh-question tree.

13.3 Wh-moved NP complement

Wh-questions can be formed on every NP object or indirect object that appears in the declarative tree or in the passive trees, as seen in sentences (124)-(129). A tree family will contain one tree for each of these possible NP complement positions. Figure 13.3 shows the two extraction trees from the ditransitive tree family for the extraction of the direct (Figure 13.3(a)) and indirect object (Figure 13.3(b)).

(124) Dania asked Beth a question.

(125) Who_i did Dania ask ϵ_i a question?

(126) What_i did Dania ask Beth ϵ_i?

(127) Beth was asked a question by Dania.

(128) Who_i was Beth asked a question by ϵ_i??

(129) What_i was Beth asked ϵ_i? by Dania?

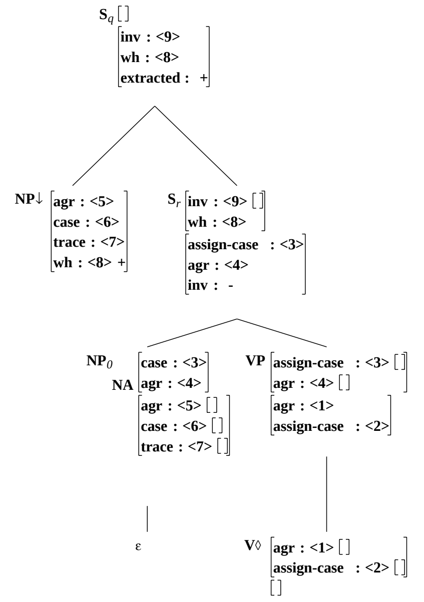
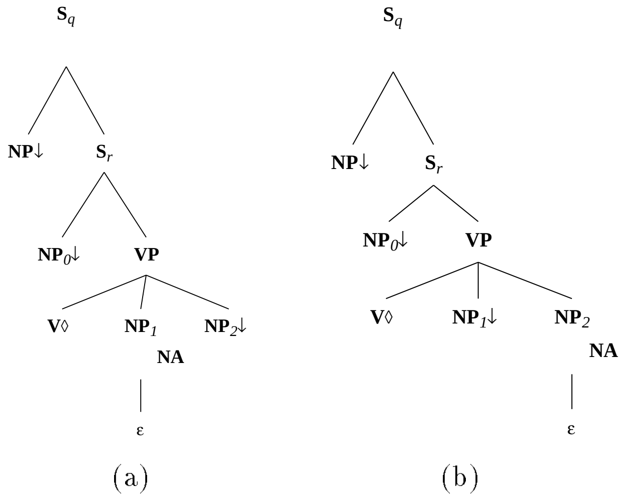


Figure 13.2: Intransitive tree with subject extraction: αW0nx0V



(a) (b)

Figure 13.3: Ditransitive trees with direct object: αW1nx0Vnx1nx2 (a) and indirect object extraction: αW2nx0Vnx1nx2 (b)

13.4 Wh-moved object of a P

Wh-questions can be formed on the NP object of a complement PP as in sentence (130).

(130) [Which dog]$_i$ did Beth Ann give a bone to ϵ_i?

The *by* phrases of passives behave like complements and can undergo the same type of extraction, as in (131).

(131) [Which dog]$_i$ was the frisbee caught by ϵ_i?

Tree structures for this type of sentence are very similar to those for the wh-extraction of NP complements discussed in section 13.3 and have the identical important features related to tree structure and trace and inversion features. The tree in Figure 13.4 is an example of this type of tree. Topicalization of NP objects of prepositions is handled the same way as topicalization of complement NP's.

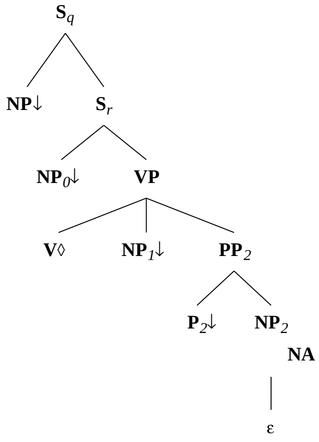


Figure 13.4: Ditransitive with PP tree with the object of the PP extracted: αW2nx0Vnx1pnx2

13.5 Wh-moved PP

Like NP complements, PP complements can be extracted to form wh-questions, as in sentence (132).

(132) [To which dog]$_i$ did Beth Ann throw the frisbee ϵ_i?

As can be seen in the tree in Figure 13.5, extraction of PP complements is very similar to extraction of NP complements from the same positions.

The PP extraction trees differ from NP extraction trees in having a PP rather than an NP left daughter node under S$_q$ and in having the ϵ fill a PP rather than an NP position in the VP. In other respects these PP extraction structures behave like the NP extractions, including being used for topicalization.

13.6 Wh-moved S complement

Except for the node label on the extracted position, the trees for wh-questions on S complements look exactly like the trees for wh-questions on NP's in the same positions. This is because there is no separate wh-lexical item for clauses in English, so the item *what* is ambiguous between representing a clause or an NP. To illustrate this ambiguity notice that the question in (133) could be answered by either a clause as in (134) or an NP as in (135). The extracted NP in these trees is constrained to be <**wh**>=+, since sentential complements can not be topicalized.

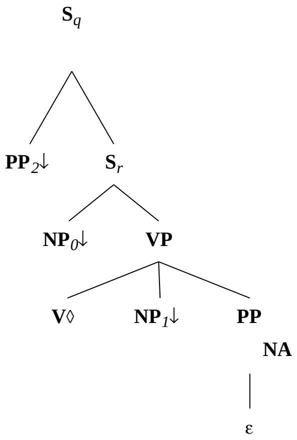


Figure 13.5: Ditransitive with PP with PP extraction tree: αpW2nx0Vnx1pnx2

(133) What does Clove want?

(134) for Beth Ann to play frisbee with her

(135) a biscuit

13.7 Wh-moved Adjective complement

In subcategorizations that select an adjective complement, that complement can be questioned in a wh-question, as in sentence (136).

(136) How$_i$ did he feel ϵ_i?

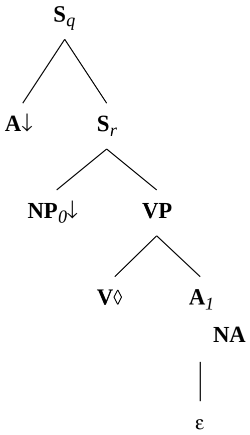


Figure 13.6: Predicative Adjective tree with extracted adjective: αWA1nx0Va1

The tree families with adjective complements include trees for such adjective extractions that are very similar to the wh-extraction trees for other categories of complements. The adjective position in the VP is filled by an ϵ and the trace feature of the adjective complement and of the adjective daughter of S_q are co-indexed. The extracted adjective is required to be <**wh**>=+[2], so no topicalizations are allowed. An example of this type of tree is shown in Figure 13.6.

[2] *How* is the only <**wh**>=+ adjective currently in the XTAG English grammar.

Chapter 14

Relative Clauses

Relative clauses are NP modifiers. For relative clauses on arguments, an argument in the clause is extracted, and the NP head (the portion of the NP being modified by the relative clause) is interpreted as having the same role in the clause as the extracted item. For example in (137) *export exhibitions* is the head NP and is modified by the relative clause ϵ *included high-tech items. Export exhibitions* is interpreted as the subject of the relative clause which is missing an overt subject.

(137) export exhibitions that included high-tech items

Relative clauses are represented in the English XTAG grammar by auxiliary trees that adjoin to NP's. These trees are anchored by the verb in the clause and appear in the appropriate tree families for the various verb subcategorizations. Within a tree family there will be groups of relative clause trees based on the declarative tree and each passive tree. Within each of these groups, there is a separate relative clause tree corresponding to each possible argument that can be extracted from the clause. The relationship between the extracted position and the head NP is captured by co-indexing the <**trace**> features of the extracted NP and the NP foot node in the relative clause tree. Representative examples from the transitive tree family are shown with a relevant subset of their features in Figures 14.1(a) and 14.1(b).

Our treatment of relative clauses allows a single tree to provide the structure for various relative clause types. For example, the tree shown in Figure 14.1(a) is used for all of the relative clauses shown in sentences (138)-(143).

(138) the man that Muriel likes

(139) the man who Muriel likes

(140) the man Muriel likes

(141) what Muriel likes

(142) the book for Muriel to read

(143) the book Muriel is reading

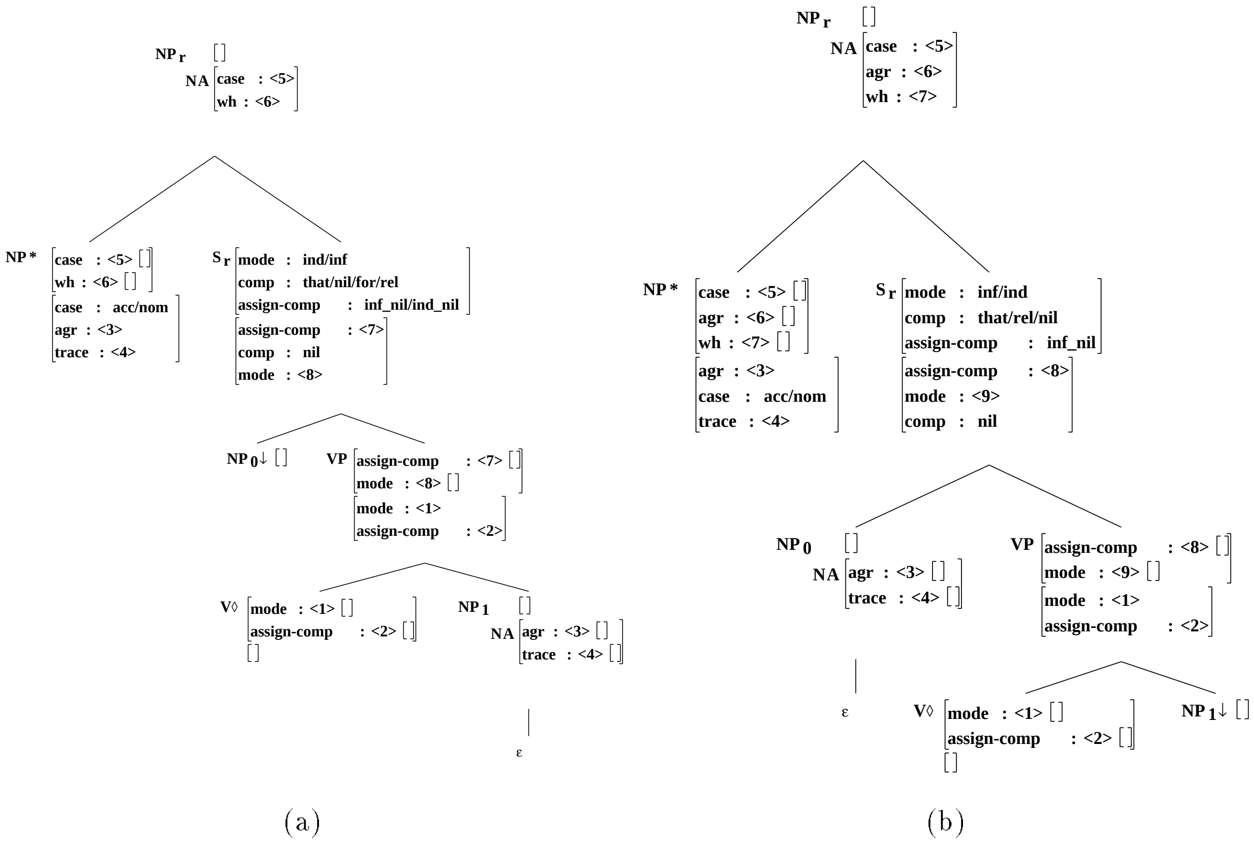


Figure 14.1: Relative clause trees in the transitive tree family: βN1nx0Vnx1 (a) and βN0nx0Vnx1 (b)

This variety of clause types is achieved through combinations of different clause types using the <**mode**> feature, different complementizers using the <**comp**> and <**assign-comp**> features and <**wh**>=+ or <**wh**>=– NP heads. A detailed discussion of how the <**mode**>, <**comp**> and <**assign-comp**> features are used to account for embedded clauses in general can be found in Chapter 8.

The relative pronouns *who* and *which* are treated as complementizers restricted to relative clauses. Their treatment as complementizers is consistent with our treatment of the complementizers *that* and *for* in other embedded clause environments as well as in relative clauses. Like other complementizers, the relative complementizers use the tree in Figure 14.2.

The relative complementizers, *who* and *which*, have **rel** as their value for the feature <**comp**>. This feature value insures that *who* and *which* do not adjoin onto sentential complements, subjects or adjunct modifiers because only relative clause trees allow complementizers with the value **rel**. Relative clause trees such as the one in Figure 14.1(a) also allow other complementizers with the appropriate clause type. For example, in sentence (142) the infinitive relative clause with an overt subject requires the complementizer *for* just as an infinitive with an overt subject would in other embedded clauses. Similarly, the adjunction of the complementizer *that* is optional in indicative relative clauses with non-subject extractions, such as in sentences (138) and (140), just as it is in sentential complements. The same system of features,

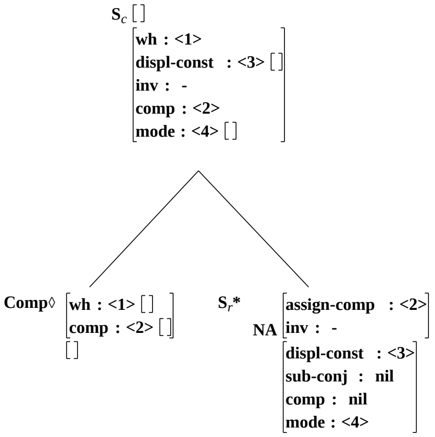


Figure 14.2: Complementizer tree: βCOMPs

<**comp**>, <**mode**> and <**assign-comp**>, is used in all cases of embedded clauses, including relative clauses, to insure the proper cooccurrence of complementizers and clause types.

Under this account, free relatives as in sentence (141) require no additional mechanisms. They are simply <**wh**>=+ NP heads with complementizerless relative clauses. For example, the clause *Mary likes* ϵ, using the tree in Figure 14.1(a), adjoins onto the NP *what* to derive *what Mary likes* ϵ.

The English XTAG grammar does not contain any syntactic distinction between restrictive and non-restrictive relatives because we believe this to be a semantic and/or pragmatic difference.

Chapter 15

Adjunct Clauses

In each tree family, there is a pair of indicative adjunct clause trees, exemplified below in Figure 15.1 with a transitive verb.

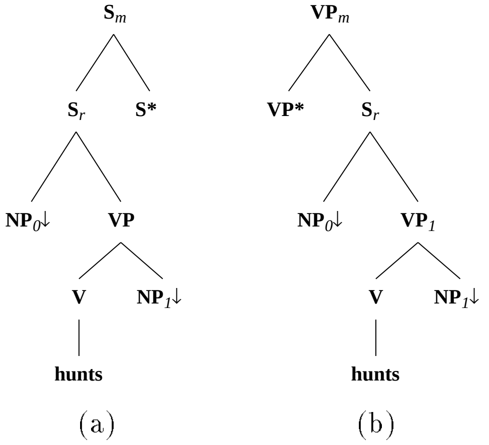


Figure 15.1: Auxiliary Trees for Sentence Initial: βnx0Vnx1s (a) and Sentence Final: βvxnx0Vnx1 (b) Adjunct Clauses

Sentence-initial adjuncts adjoin at the root S of the matrix clause (Figure 15.1(a)), while sentence-final adjuncts adjoin at a VP node (Figure 15.1(b)). In this, the XTAG analysis follows the findings on the attachment sites of adjunct clauses for conditional clauses ([Iatridou, 1991]) and for infinitival clauses ([Browning, 1987]). One compelling argument is based on Binding Condition C effects. As can be seen from examples (144)-(146) below, no Binding Condition violation occurs when the adjunct is sentence initial, but the subject of the matrix clause clearly governs the adjunct clause when it is in sentence final position and co-indexation of the pronoun with the subject of the adjunct clause is impossible.

(144) Unless she$_i$ hurries, Mary$_i$ will be late for the meeting.

(145) $*$She$_i$ will be late for the meeting unless Mary$_i$ hurries.

(146) Mary$_i$ will be late for the meeting unless she$_i$ hurries.

Tree families with direct objects also contain a pair for the passive trees, and the transitive family (Tnx0Vnx1) contains a pair for the ergative trees. All of these trees are anchored by the main verb of the adjunct clause, and adjoin either at S or VP to the matrix clause. Subordinating conjunctions adjoin to these sentential adjunct trees, as described in section 15.2 below. If no conjunction adjoins, only certain modes are licensed for the adjunct clause. These are described immediately below.

15.1 "Bare" Adjunct Clauses

As described in Chapter 8 on sentential complements and complementizers, the features <**mode**> and <**assign-comp**> are used to control the occurrence of complementizers with the various clause types. This same mechanism is used here to ensure the correct distribution of 'bare' (i.e. conjunction-less) adjunct clauses. Three values of <**mode**> are licensed: **inf** (infinitive), **ger** (gerundive) and **ppart** (past participal, licensed only for passive adjuncts).[1] They interact with complementizers as follows:

- Participial complements do not license any complementizers:[2]

  (147) [Destroyed by the fire], the building still stood.

  (148) The fire raged for days [destroying the building].

  (149) *[That destroyed by the fire], the building still stood.

- Infinitival adjuncts, including purpose clauses, are licensed both with and without the complementizer *for.*

  (150) Harriet bought a Mustang [to impress Eugene].

  (151) [To impress Harriet], Eugene dyed his hair.

  (152) Traffic stopped [for Harriet to cross the street].

15.2 Clauses Conjoined with Subordinating Conjunctions

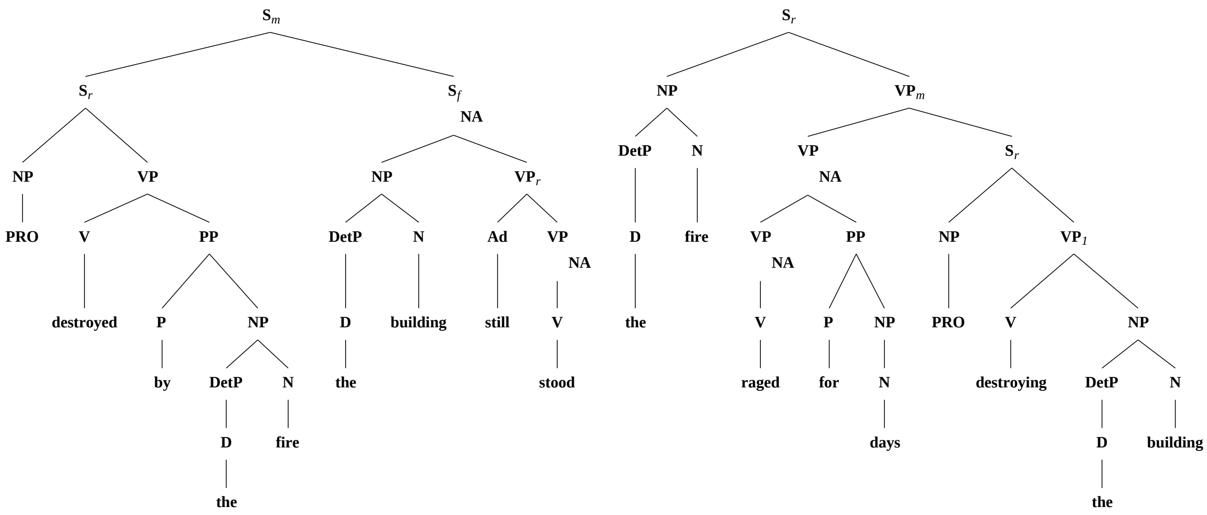
Subordinating conjunctions anchor one of the four auxiliary trees shown in Figure 15.3.[3] The tree in Figure 15.3(a) is selected by a great majority of subordinating conjunctions. Figure 15.3(b) is anchored by multi-word conjunctions.[4] The list of multi-word conjunctions was extracted from [Quirk *et al.*, 1985], and includes *as if*, *in order*, and *for all (that).* The remaining two trees, seen in Figures 15.3(c) and 15.3(d), handle the three word conjunctions in (153)

[1] We considered allowing bare indicative clauses, such as *He died that other may live*, but these were considered too archaic to be worth the additional ambiguity they would add to the grammar.

[2] While these sound a bit like extraposed relative clauses (see [Kroch and Joshi, 1987]), those move only to the right and adjoin to S; as these clauses are equally grammatical both sentence-initially and sentence-finally, we are analyzing them as adjunct clauses.

[3] There is some amount of overlap between subordinating conjunctions and prepositions. Items which were already in the grammar as prepositions were not added as subordinating conjunctions where this would have resulted in duplicate analyses.

[4] UCONJ means 'unanalyzed' conjunction, i.e. both words are not conjunctions themselves, but together they form a complex subordinating conjunction.



(a) (b)

Figure 15.2: Sample Participial Adjuncts

and (154) respectively. Thus, the former has two Conj anchors and an adverb substitution node, while the latter has three anchors. This multi-anchor treatment is very similar to that proposed for idioms in [Abeille and Schabes, 1989], and the analysis of light verbs in the XTAG grammar (see section 6.15).

(153) *as recently/quickly/*etc. *as* + indicative complement

(154) *as soon as* + participial complement

Each of these trees adjoins at the interior S_r node of the S and VP sentential adjunct trees described above and shown in Figure 15.1. Subordinating conjunctions are grouped into classes, based on the type of clause to which they may adjoin and whether they allow a complementizer to also adjoin to the clause. Each class instantiates a value for the <**sub-conj**> feature at the root S, which prevents subordinating conjunctions from stacking. They also instantiate values of the <**mode**> and <**comp**> features of the foot S. The <**mode**> value constrains the types of clauses the subordinating conjunction may adjoin to and the <**comp**> value constrains the complementizers which may adjoin below it. These classes are:

- IND1: Indicative clause with optional *that* complementizer, e.g. *in order*, *so.*

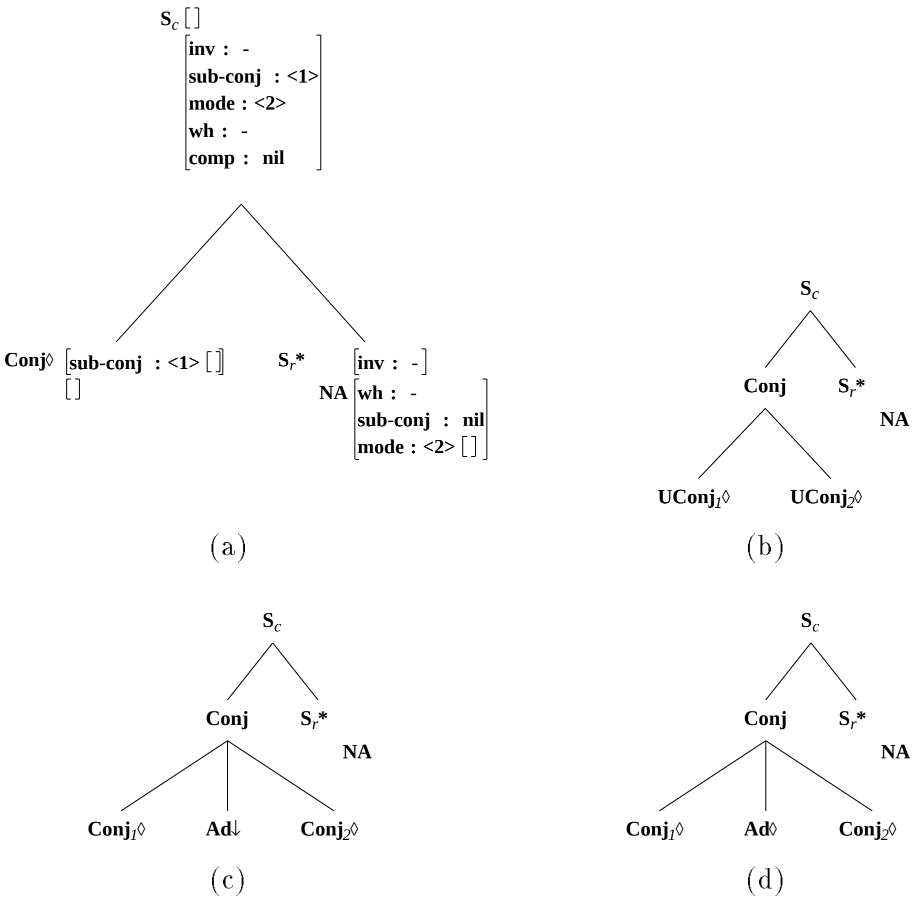
  (155) He died so (that) others could live.

- IND2: Indicative clause, no complementizer possible, e.g. *in case*, *because.*

  (156) Because Bill ate their lettuce the rabbits are sad.

  (157) ∗Because that Bill ate their lettuce the rabbits are sad.

- IND3: Asymmetric versions of coordinating conjuncts *and* and *but*; indicative clause, no complementizer possible, only allowed in sentence-final clausal adjunct trees.



(a) (b)

(c) (d)

Figure 15.3: Trees Anchored by Subordinating Conjunctions: βCONJs (a), βUCONJUCONJs (b), βCONJarbCONJs (c) and βCONJARBCONJs

(158) Paddington opened the closet and his galoshes were inside.

(159) ∗And his galoshes were inside Paddington opened the closet.

- INF1: Infinitival clause, no complementizer; only *so as*, *as if* and *as though.*

(160) As if he had planned it, the door suddenly opened.

(161) ∗As if for Bill he had planned it, the door suddenly opened.

- INF2: Infinitival clause optional *for* complementizer; only *in order.*

(162) Max picked the lettuce in order to eat it.

(163) Max picked the lettuce in order for us to eat it.

- GER: Participial (<**mode**>=**ger** or **ppart**) complement, no complementizer possible, e.g. *although*, *even if*, *when.*

(164) Drawn recently, the pictures are valuable.

(165) Max ate spinach, impressing Mary.

These auxiliary trees are also used to do 'discourse' coordination, as in sentence (166). All subordinating conjunctions which can conjoin indicative clauses may also adjoin to root matrix sentences, as seen in the derived tree in Figure 15.4.

(166) And Truffula trees are what everyone needs! [Seuss, 1971]

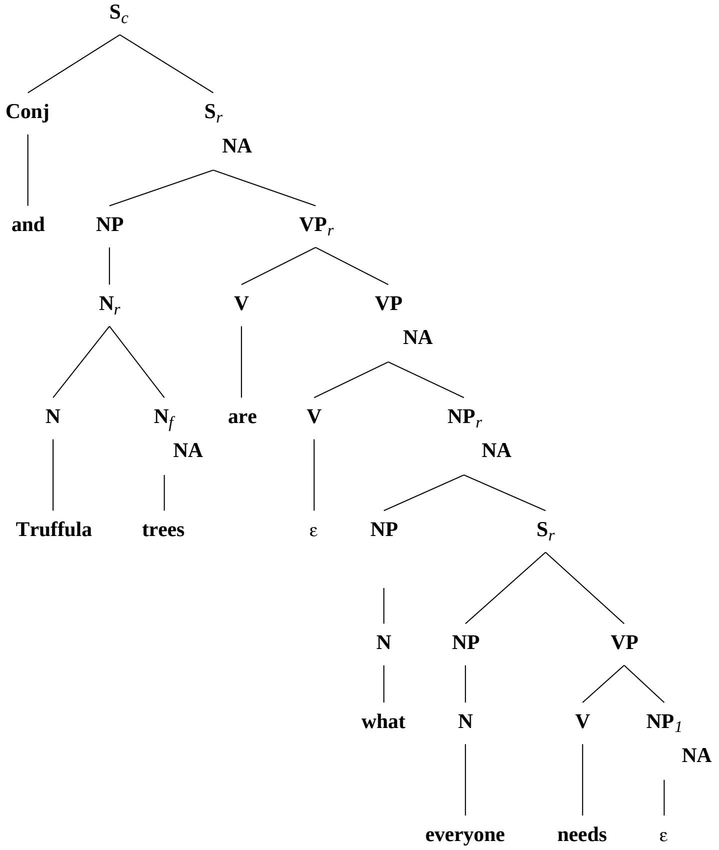


Figure 15.4: Example of discourse conjunction, from Seuss' *The Lorax*

Chapter 16

Imperatives

Imperatives in English do not require overt subjects. The subject in imperatives is second person, i.e. *you*, whether it is overt or not, as is clear from the verbal agreement and the interpretation. Imperatives with overt subjects can be parsed using the trees already needed for declaratives. The imperative cases in which the subject is not overt are handled by the imperative trees discussed in this section.

The imperative trees in English XTAG grammar are identical to the declarative tree except that the NP_0 subject position is filled by an ϵ, the NP_0 <**agr pers**> feature is set in the tree to the value **2nd** and the <**mode**> feature on the root node has the value **imp**. The value for <**agr pers**> is hardwired into the epsilon node and insures the proper verbal agreement for an imperative. The <**mode**> value of **imp** on the root node is recognized as a valid mode for a matrix clause. The **imp** value for <**mode**> also allows imperatives to be blocked from appearing as embedded clauses. Figure 16.1 is the imperative tree for the transitive tree family.

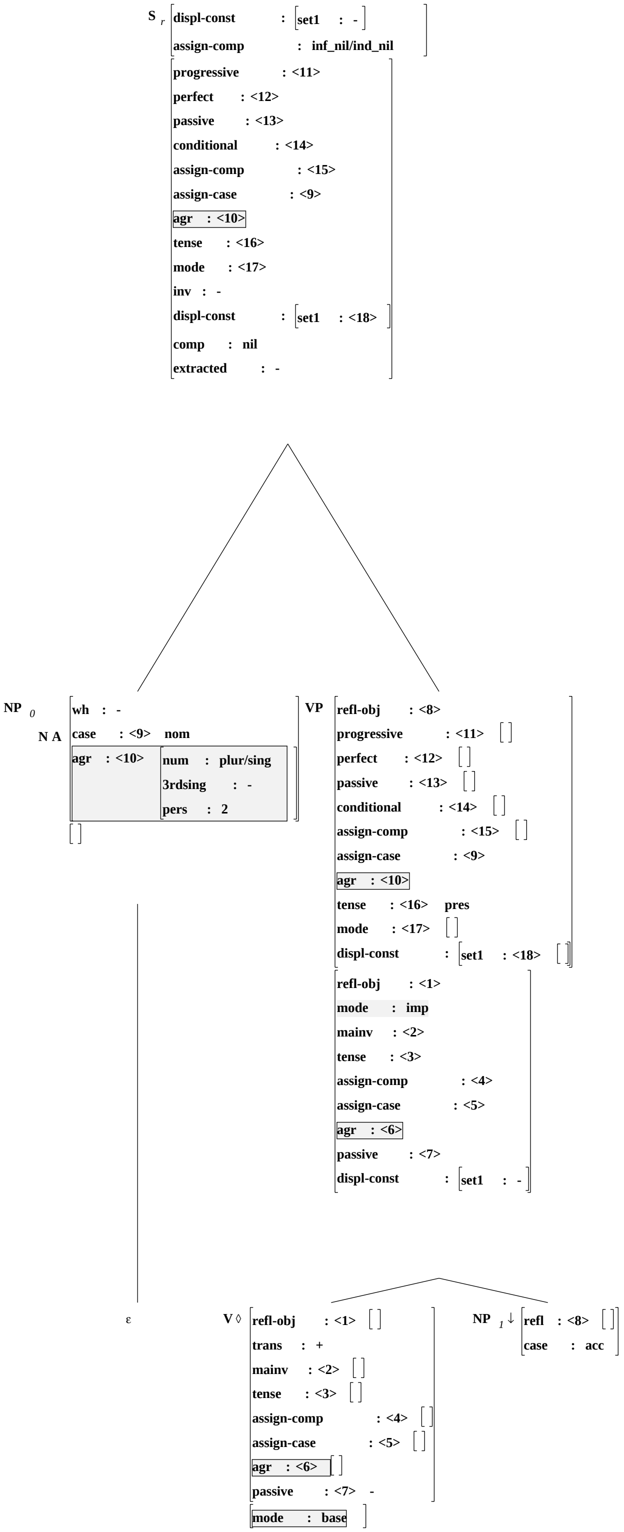


Figure 16.1: Transitive imperative tree: αInx0Vnx1

Chapter 17

Gerund NP's

The puzzle over gerunds in the linguistics literature has been that they seem to have both NP and clausal properties. That is to say they seem to have certain clausal properties but occur in positions typically occupied by noun phrases. The bold face portions of examples (167)-(169) show examples of gerunds as subjects in (167) and (168), and as the object of a preposition in (169).

(167) And **avoiding such losses** will take a monumental effort.

(168) **Mr. Nolen's nocturnal wandering** doesn't make him a weirdo.

(169) Are private markets approving of **Washington's bashing of Wall Street**?

In the English XTAG grammar we adopt a position similar to that of [Rosenbaum, 1967] and [Emonds, 1970] - that gerunds are NP's exhaustively dominating a clause. In particular, we found that any place an NP is allowed, a gerundive clause is also allowed, and no cases in which a verb subcategorized for gerundive clauses, but not NP's.

Our implementation includes at least two gerundive trees in each tree family (see Figure 17.1). The gerund trees in a tree family have basically the form of the declarative for that family but have NP as the category of their top node. The Determiner Gerund tree in Figure 17.1(a) has an initial DetP and instantiates the direct object as a PP. It is used for gerunds such as the one in bold face in sentence (170).

(170) Some think **the rapid selling of bonds** has a way to go.

Notice that the modification of *selling of bonds* by the adjective *rapid* supports the choice of N as the label for the node dominating V and PP_1.

The NP gerund tree in Figure 17.1(b) has exactly the same structure as the declarative transitive tree except for the root node label and for feature values. In particular, the verb is required to be <**mode**>=**ger**, and the subject is required to be <**case**>=**acc/none/gen**, i.e. either an accusative, PRO or genitive NP. The whole NP formed by the gerund can itself have either nominative or accusative case. The NP gerund tree is used for gerunds such as the one in bold face in sentence (167) and (168).

One question that arises with respect to gerunds is whether there is anything special about their distribution as compared to other types of NP's. In fact, it appears that gerund NP's

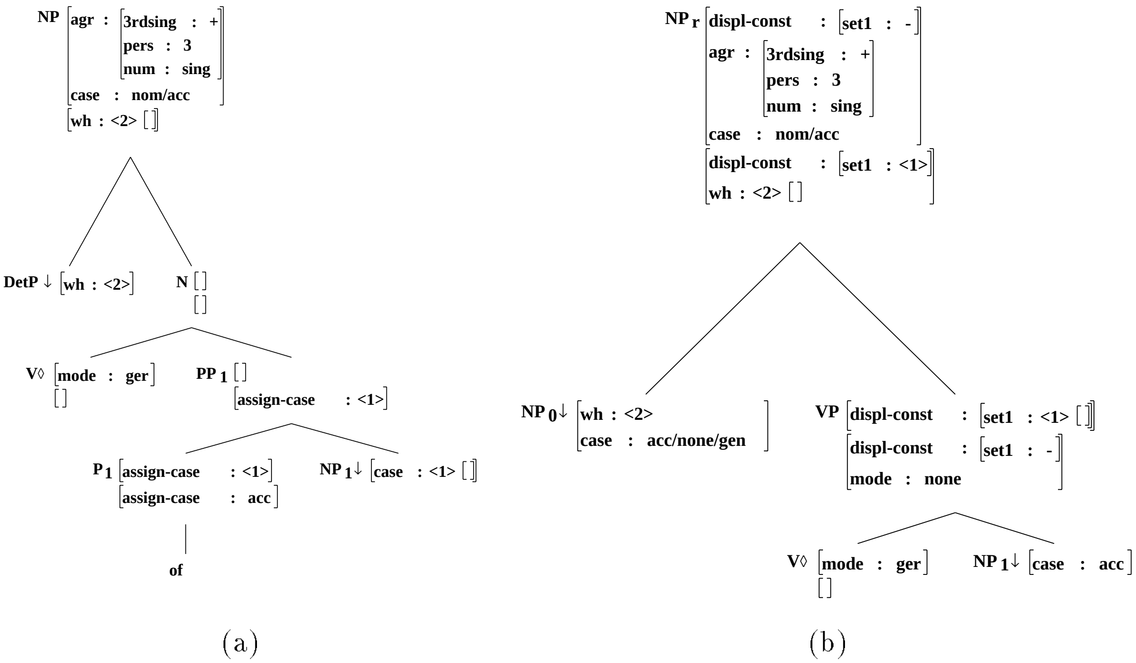


Figure 17.1: Gerund trees from the transitive tree family: αDnx0Vnx1 (a) and αGnx0Vnx (b)

can occur in any NP position. Some verbs might not seem to be very accepting of gerund NP arguments, as in (171), but we believe this to be a semantic incompatibility rather than a syntactic problem since the same structures are fine with other lexical items.

(171) ?[$_{NP}$John's tinkering$_{NP}$] ran.

(172) [$_{NP}$John's tinkering$_{NP}$] worked.

By having the root node of gerund trees be NP, the gerunds have the same distribution as any other NP in the English XTAG grammar without doing anything exceptional. The clause structure is captured by the form of the trees and by inclusion in the tree families.

Part IV

Other Constructions

Chapter 18

Determiners and Noun Phrases

Previous approaches to syntactic determiner ordering (e.g. [Quirk *et al.*, 1985]) have simply divided determiners into subcategories (predet, det, postdet). This type of approach is inadequate because it allows ungrammatical sequences like $*$*all what no*, and misses the finer distinctions among particular determiners. These finer distinctions are modeled very naturally in a lexicalized grammar formalism such as FB-LTAG in which pieces of syntactic structure and features representing linguistic properties are associated with individual lexical items.

In the English XTAG grammar,[1] there are two kinds of basic noun phrases (NP), those that take determiner phrases[2] and those that do not. Nouns that take (or require) determiners have a DetP substitution site. Complex DetP's are formed by having determiners adjoin onto each other. There are two basic determiner trees: an initial tree and an auxiliary tree. Figure 18.1 shows the initial and auxiliary trees anchored by the determiner *these*. Since any single determiner can function as a complete DetP,[3] every determiner selects the initial tree in Figure 18.1(a). Determiners that can modify other determiners also select the auxiliary tree in Figure 18.1(b).

The current grammar includes a DetP substitution node in the NP but having determiners adjoin on has also been proposed in the literature ([Abeillé, 1990]). The correct ordering of determiners and reasonable coverage is possible with either approach. In fact, the core of our analysis is based on the features and would essentially be the same with adjoined DetP's. We are currently considering whether there would be any compelling advantages of an adjunction analysis for the XTAG grammar.

In the XTAG grammar, features are crucial to ordering determiners correctly. We have identified eight features which are sufficient to order the determiners. These features are: **definiteness**, **quantity**, **cardinality**, **genitive**, **decreasing**, **constancy**, **wh** and **agr**. These features have all been previously proposed as semantic properties of determiners. The semantic definitions underlying the features are given below.

Definiteness: Possible Values [+/−].

A function f is definite iff f is non-trivial and whenever $f(s) \neq \emptyset$ then it is always the

[1] This chapter is a shortened version of [Hockey and Egedi, 1994], which contains a more extensive discussion of this analysis.

[2] Henceforth DetP's, not to be confused with DP's as in the DP Hypothesis.

[3] By definition. Our main criteria in classifying something as a determiner was that it be able to stand alone with a noun to form an NP.

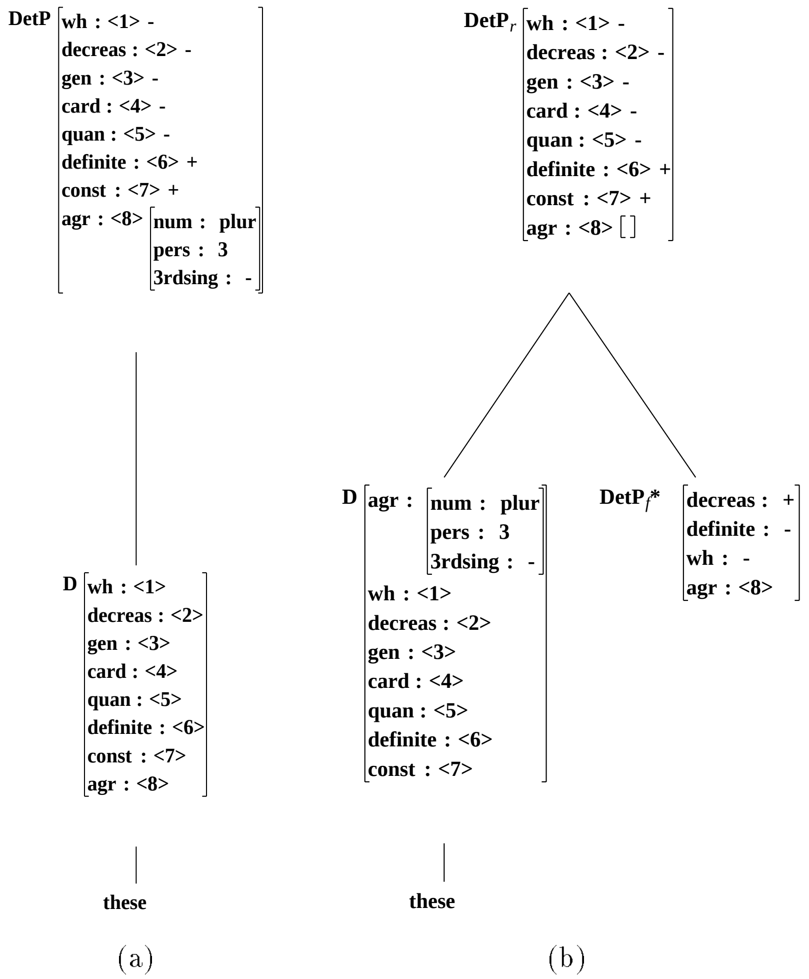


Figure 18.1: Determiner Trees with Features: αDXD (a) and βDdx (b)

intersection of one or more individuals. [Kennan and Stavi, 1986]

Quantity: Possible Values [+/−].
If A and B are sets denoting an NP and associated predicate, respectively; E is a domain in a model M, and F is a bijection from M_1 to M_2, then we say that a determiner satisfies the constraint of quantity if $\mathrm{Det}_{E_1}\mathrm{AB} \leftrightarrow \mathrm{Det}_{E_2}\mathrm{F(A)F(B)}$. [Partee *et al.*, 1990]

Cardinality: Possible Values [+/−].
A determiner D is cardinal iff D $\in$ cardinal numbers ≥ 1.

Genitive: Possible Values [+/−].
Possessive pronouns and the possessive morpheme (*'s*) are marked **gen+**; all other nouns are **gen−**.

Decreasing: Possible Values [+/−].
A set of Q properties is decreasing iff whenever s$\leq$t and t$\in$Q then s$\in$Q. A function f is decreasing iff for all properties f(s) is a decreasing set.

A non-trivial NP (one with a Det node) is decreasing iff its denotation in any model is decreasing. [Kennan and Stavi, 1986]

Constancy: Possible Values [+/−].

If A and B are sets denoting an NP and associated predicate, respectively, and E is a domain, then we say that a determiner displays constancy if $(A \cup B) \subseteq E \subseteq E'$ then $\mathrm{Det}_E AB \leftrightarrow \mathrm{Det}_{E'} AB$. Modified from [Partee *et al.*, 1990]

Wh: Possible Values [+/−].

Interrogative determiners are **wh+**; all other determiners are **wh−**.

Agreement: Possible Values [3sg, 3pl, 3sgpl].

Although English does not have the morphological marking of determiners for case, gender or number, we hold that most determiners in English are semantically marked for number.

The initial determiner tree in Figure 18.1(a) shows the appropriate feature values for the determiner *these*, while Table 18.1 shows the corresponding feature values of several other common determiners.

Det	defin	quan	card	gen	wh	decreas	const	agr
all	+	+	−	−	−	−	+	3pl
this	+	−	−	−	−	−	+	3sg
that	+	−	−	−	−	−	+	3sg
what	+	−	−	−	+	−	+	3sgpl
the	+	−	−	−	−	−	+	3sgpl
every	+	+	−	−	−	−	+	3sg
each	+	+	−	−	−	−	+	3sg
any	−	+	−	−	−	−	+	3sg
a	−	+	−	−	−	−	+	3sg
no	+	+	−	−	−	−	+	3sgpl
few	−	+	−	−	−	+	−	3pl
many	−	+	−	−	−	−	−	3pl
GEN	+	−	−	+	−	−	+	
CARD	+	+	+	−	−	−	+	3pl[4]
PART	−		−	−	−	−	+	

Table 18.1: Determiner Features

In addition to the features that represent their own properties, determiners that select the auxiliary tree have features to represent the selectional restrictions these determiners impose on the determiners they modify. The selectional restriction features of a determiner appear on the DetP foot node of the auxiliary tree that the determiner anchors. The DetP_f node in the auxiliary tree in Figure 18.1(b) shows the selectional feature restriction imposed by *these*,[5] while Table 18.2 shows the corresponding selectional feature restrictions of several other determiners.

[4] except *one* which is 3sg.

[5] In addition to this tree, *these* would also anchor another auxiliary tree that adjoins onto <**card**>=+ determiners.

Det	defin	quan	card	gen	wh	decreas	const	agr
all	+	−	− +		−			
this	−		+		−	+		
that	−		+		−	+		
what	−		+		−	+		
the	−		+		−		−	
every	−		+		−	+		
each	−		+		−	+		
any	−		+		−	+		
a	−				−	+		
many	only nouns							
no	only nouns							
GEN	only nouns							
CARD	only nouns							
PART					−			

Table 18.2: Selectional Restrictions Imposed by Determiners

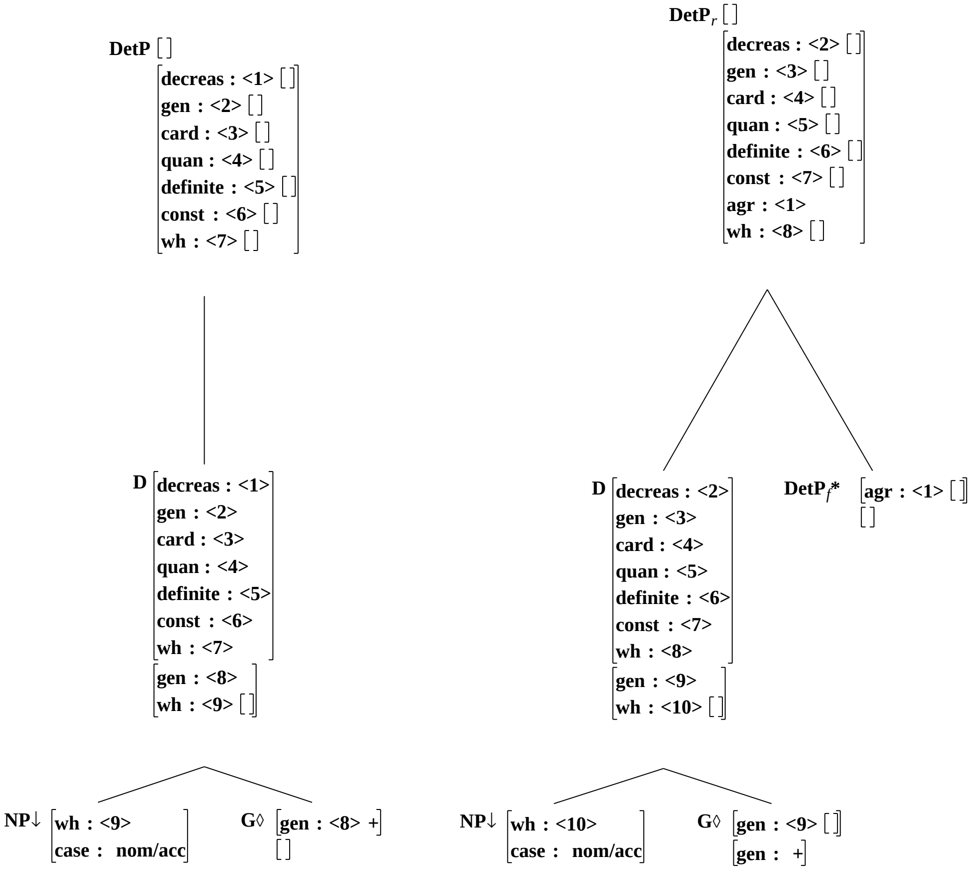
18.1 Wh and Agr Features

A determiner with a <**wh**>=+ feature is always the leftmost determiner since no determiners can adjoin onto it. The presence of a wh+ determiner makes the entire NP wh+, so this feature is always passed through to the NP node, unlike other features which are considered internal to the determiner system.

The <**agr**> feature functions differently from most of the features in the determiner sequencing system. Notice that in the auxiliary tree in Figure 18.1(b), the <**agr**> feature is the only feature not passed from the D node to the root DetP node, but is passed instead from the foot DetP to the root DetP. In the determiner system, the <**agr**> feature is generally propagated from the rightmost determiner (i.e. the one closest to the noun), although some adjoining determiners require that they also agree with that determiner. This distinction is captured in XTAG by having each determiner specify in its lexical entry whether or not its agreement feature is passed to the root DetP (i.e. from the D node to the DetP$_r$ node).

18.2 Genitive Constructions

There are two kinds of genitive constructions: genitive pronouns, and genitive NP's (which have an explicit genitive marker, *'s*, associated with them). It is clear from examples such as *her favorite artist prefers oils* vs. ∗*favorite artist prefers oils* that genitive pronouns function as determiners and as such, they sequence with the rest of the determiners. The features for the genitives are the same as for other determiners, and are given in Table 18.1. No <**agr**> is specified however, since the number and person of the genitive will depend on its particular form (e.g. *my* vs. *their*). Genitives are not required to agree with either the determiners or the nouns that they modify.



(a) (b)

Figure 18.2: Initial: αDXnxG (a) and Auxiliary: βnxGdx (b) Genitive Determiner Trees

Genitive NP's are particularly interesting because they are potentially recursive structures. Complex NP's can easily be embedded in a determiner phrase.

(173) [[[John]'s friend from high school]'s uncle]'s mother came to town.

There are two things to note in sentence (173). One is that in embedded NP's, the genitive morpheme comes at the end of the NP phrase, even if the head of the NP is at the beginning of the phrase. The other is that the determiner of an embedded NP can also be a genitive NP, hence the possibility of recursive structures.

In the XTAG grammar, the genitive marker, *'s*, is separated from the lexical item that it is attached to and given its own category (G). In this way, we can allow the full complexity of

NP's to come from the existing NP system, including any recursive structures. The two trees in Figure 18.2 demonstrate how easily the complexity of genitive NP's are captured in XTAG. As with the standard determiner trees, there are two trees - one for the determiner that stands alone and one for a determiner that adjoins onto another.

18.3 Partitive Constructions

Partitive constructions (e.g. *some kind of*, *all of*) are another type of complex determiner construction. Partitive constructions interact with other determiners. Since they can modify the noun itself (*[a certain kind of] machine*), or modify other determiners (*[some parts of] these machines*), the partitive construction has both an initial and auxiliary tree that are anchored by the preposition *of*. The partitive trees are shown in Figure 18.3.

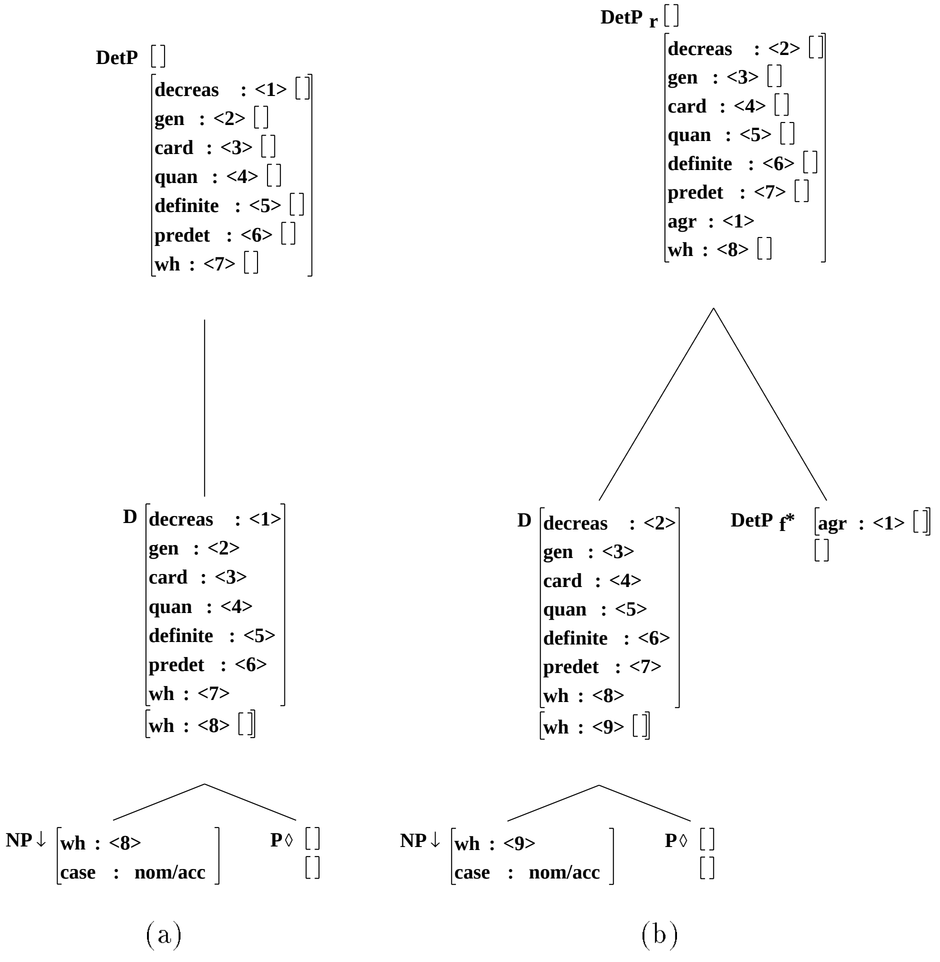


Figure 18.3: Initial: αDXnxP (a) and Auxiliary: βnxPdx (b) Partitive trees

Chapter 19

Modifiers

This chapter covers various types of modifiers: adverbs, prepositions, adjectives, and noun modifiers in noun-noun compounds.[1] These categories optionally modify other lexical items and phrases by adjoining onto them. In their modifier function these items are adjuncts; they are not part of the subcategorization frame of the items they modify. Examples of some of these modifiers are shown in (174)-(176).

(174) $[_{ADV}$ Certainly $_{ADV}]$, the October 13 sell-off didn't settle any stomachs. (WSJ)

(175) Mr. Bakes $[_{ADV}$ previously $_{ADV}]$ had a turn at running Continental. (WSJ)

(176) Most $[_{ADJ}$ foreign $_{ADJ}]$ $[_N$ government $_N]$ $[_N$ bond $_N]$ [prices] rose $[_{PP}$ in light trading $_{PP}]$. (WSJ)

The trees used for the various modifiers are quite similar in form. The modifier anchors the tree and the root and foot nodes of the tree are of the category that the particular anchor modifies. Some modifiers, e.g. prepositions, have arguments that are also included in the tree. The foot node may be to the right or the left of the anchoring modifier (and its arguments) depending on whether that modifier occurs before or after the category it modifies. For example, almost all adjectives appear to the left of the nouns they modify, while prepositions appear to the right when modifying nouns.

19.1 Adjectives

In addition to being modifiers, adjectives in the XTAG English grammar can be also anchor clauses (see Adjective Small Clauses in Chapter 9). There is also one tree family, Intransitive with Adjective (Tnx0Va1), that has an adjective as an argument and is used for sentences such as *Seth felt happy.* In that tree family the adjective substitutes into the tree rather than adjoining as is the case for modifiers.

As modifiers, adjectives anchor the tree shown in Figure 19.1. The features of the N node onto which the βAn tree adjoins are passed through to the top node of the resulting N. The null adjunction marker (NA) on the N foot node imposes right binary branching such that each

[1] Relative clauses are discussed in Chapter 14.

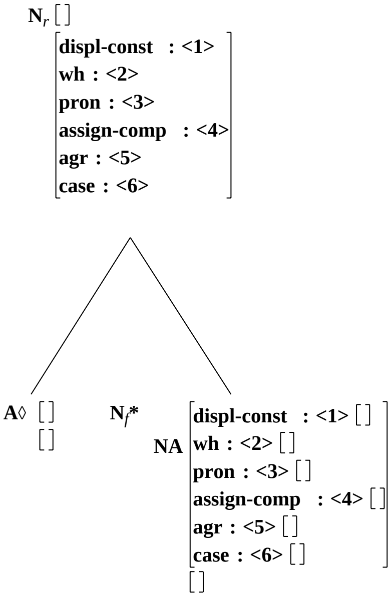


Figure 19.1: Standard Tree for Adjective modifying a Noun: βAn

subsequent adjective must adjoin on top of the leftmost adjective that has already adjoined. Due to the NA constraint, a sequence of adjectives will have only one derivation in the XTAG grammar. The adjective's morphological features such as superlative or comparative are not currently used in the tree. At this point, the treatment of adjectives in the XTAG English grammar does not include selectional or ordering restrictions. Consequently, any adjective can adjoin onto any noun and on top of any other adjective already modifying a noun. All of the modified noun phrases shown in (177)-(180) currently parse with the same structure shown for *colorless green ideas* in Figure 19.2.

(177) big green bugs

(178) big green ideas

(179) colorless green ideas

(180) ∗green big ideas

While (178)-(180) are all semantically anomalous, (180) also suffers from an ordering problem that makes it seem ungrammatical as well.[2] We would argue that the grammar should accept (177)-(179) but not (180). One of the future goals for the grammar is to develop a treatment of adjective ordering similar to that developed by [Hockey and Egedi, 1994] for determiners[3]. An adequate implementation of ordering restrictions for adjectives would rule out (180).

[2] This is, in fact, the point of the famous linguistic example in (179).

[3] See Chapter 18 or [Hockey and Egedi, 1994] for details of the determiner analysis.

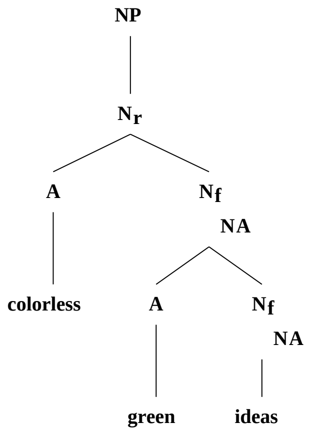


Figure 19.2: Multiple adjectives modifying a noun

Another area in which we plan to have future grammar development is comparatives. Comparatives that involve ellipsis will require a general solution of the problem of representing ellipsis. Simpler comparatives without ellipsis, such as *fewer than nine* in (181), should be amenable to analysis as complex determiners.

(181) Cats actually have fewer than nine lives.

19.2 Noun-Noun Modifiers

Noun-noun compounding in the English XTAG grammar is very similar to adjective-noun modification. The noun modifier tree, shown in Figure 19.3, has essentially the same structure as the adjective modifier tree in Figure 19.1, except for the syntactic category label of the anchor.

Noun compounds have a variety of scope possibilities not available to adjectives, as illustrated by the single bracketing possibility in (182) and the two possibilities for (183). This ambiguity is manifested in the XTAG grammar by the two possible adjunction sites in the noun-noun compound tree itself. Subsequent modifying nouns can adjoin either onto the N_r node or onto the N anchor node of that tree, which results in exactly the two bracketing possibilities shown in (183). This inherent structural ambiguity results in noun-noun compounds regularly having multiple derivations. However, the multiple derivations are not a defect in the grammar because they are necessary to correctly represent the genuine ambiguity of these phrases.

(182) $[_N$ big $[_N$ green design $_N]_N]$

(183) $[_N$ computer $[_N$ furniture design $_N]_N]$
$[_N$ $[_N$ computer furniture $_N]$ design $_N]$

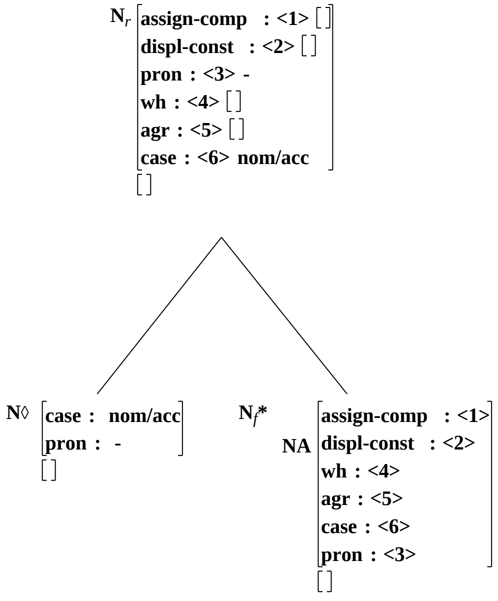


Figure 19.3: Noun-noun compounding tree: βNn

19.3 Prepositions

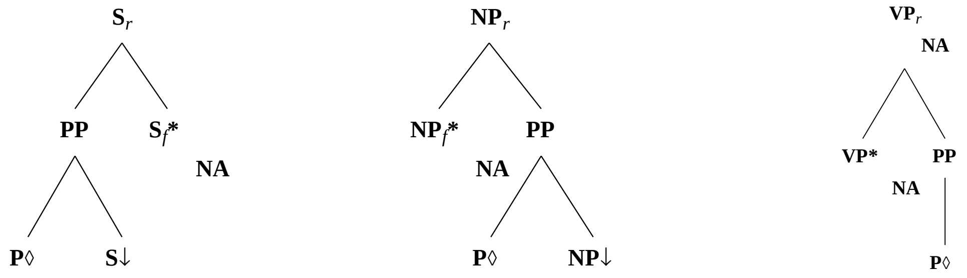
There are three basic types of prepositional phrases, and three places at which they can adjoin. The three types of prepositional phrases are: Preposition with NP Complement, Preposition with Sentential Complement, and Exhaustive Preposition. The three places are to the right of an NP, to the right of a VP, and to the left of an S. Each of the three types of PP can adjoin at each of these three places, for a total of nine PP modifier trees. Table 19.1 gives the tree family names for the various combinations of type and location.

	position and category modified		
Complement type	pre-sentential S modifier	post-NP NP modifier	post-VP VP modifier
S-complement	βPss	βnxPs	βvxPs
NP-complement	βPnxs	βnxPnx	βvxPnx
no complement (exhaustive)	βPs	βnxP	βvxP

Table 19.1: Preposition Anchored Modifiers

The subset of preposition anchored modifier trees in Figure 19.4 illustrates the three locations and the three PP types.

Example sentences using the trees in Figure 19.4 are shown in (184)-(186).



βPss βnxPnx βvxP

Figure 19.4: Selected Prepositional Phrase Modifier trees: βPss, βnxPnx and βvxP

(184) $[_{PP}$ With Clove healthy $_{PP}]$, the veterinarian's bill will be more affordable. (βPss[4])

(185) The frisbee $[_{PP}$ in the brambles $_{PP}]$ was hidden. (βnxPnx)

(186) Clove played frisbee $[_{PP}$ outside $_{PP}]$. (βvxP)

Prepositions that take NP complements assign accusative case to those complements (see section 4.4.3.1 for details). Most prepositions take NP complements. There are just a few prepositions that take sentential complements (see section 8.8).

19.4 Adverbs

In the English XTAG grammar, VP and S-modifying adverbs anchor the auxiliary trees βARBs, βsARB, βvxARB and βARBvx,[5] allowing pre and post modification of S's and VP's. Besides the VP and S-modifying adverbs, the grammar includes adverbs that modify other categories. Examples of adverbs modifying an adjective, an adverb, and a PP are shown in (187)-(192).

- Modifying an adjective

  (187) **extremely** good

  (188) **rather** tall

  (189) rich **enough**

- Modifying an adverb

  (190) oddly **enough**

  (191) **very** well

[4] *Clove healthy* is an adjective small clause

[5] In the naming conventions for the XTAG trees, ARB is used for adverbs. Because the letters in A, Ad, and Adv are all used for other parts of speech (adjective, determiner and verb), ARB was chosen to eliminate ambiguity. Appendix B contains a full explanation of naming conventions.

- Modifying a PP

  (192) **right** through the wall

XTAG has separate trees for each of the modified categories and for pre and post modification where needed. The kind of treatment given to adverbs here is very much in line with the base-generation approach proposed by [Ernst, 1983], which assumes all positions where an adverb can occur to be base-generated, and that the semantics of the adverb specifies a range of possible positions occupied by each adverb. While the relevant semantic features of the adverbs are not currently implemented, implementation of semantic features is scheduled for future work. The trees for adverb anchored modifiers are very similar in form to the adjective anchored modifier trees. Examples of two of the basic adverb modifier trees are shown in Figure 19.5.

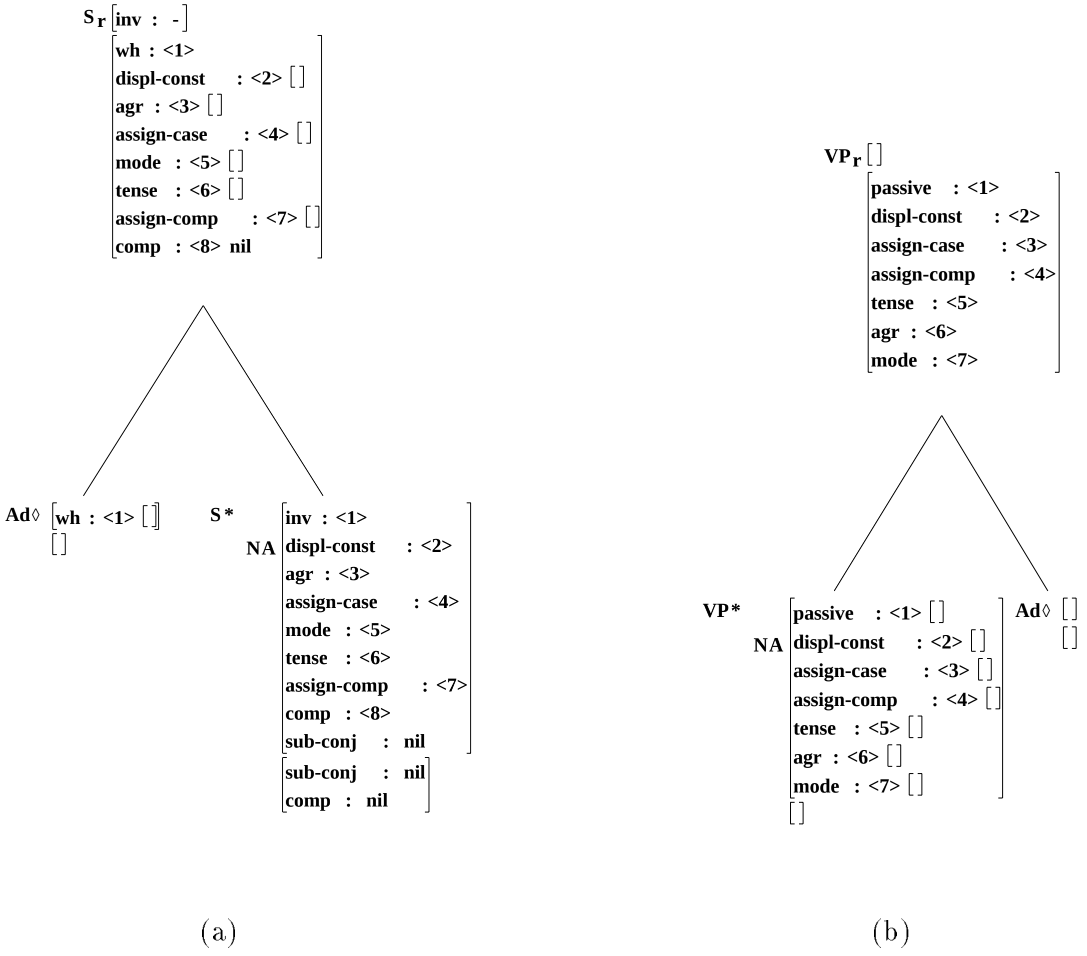


Figure 19.5: Adverb Trees for pre-modification of S: βARBs (a) and post-modification of a VP: βvxARB (b)

Like the adjective anchored trees, these trees also have the NA constraint on the foot node to restrict the number of derivations produced for a sequence of adverbs. Features of the modified category are passed from the foot node to the root node, reflecting correctly that these types of properties are unaffected by the adjunction of an adverb. A summary of the categories modified and the position of adverbs is given in Table 19.2.

Category Modified	Position with respect to item modified	
	Pre	Post
S	βARBs	βsARB
VP	βARBvx	βvxARB
A	βARBa	βaARB
PP	βARBpx	βpxARB
ADV	βARBarb	βarbARB
N	βARBn	

Table 19.2: Simple Adverb Anchored Modifiers

In the English XTAG grammar, no traces are posited for wh-adverbs, in-line with the base-generation approach ([Ernst, 1983]) for various positions of adverbs. Since convincing arguments have been made against traces for adjuncts of other types (e.g. [Baltin, 1989]), and since the reasons for wanting traces do not seem to apply to adjuncts, we make the general assumption in our grammar that adjuncts do not leave traces. Sentence initial wh-adverbs select the same auxiliary tree used for other sentence initial adverbs (βARBs) with the feature **<wh>=+**. Under this treatment, the derived tree for the sentence *How did you fall?* is as in Figure (19.6), with no trace for the adverb.

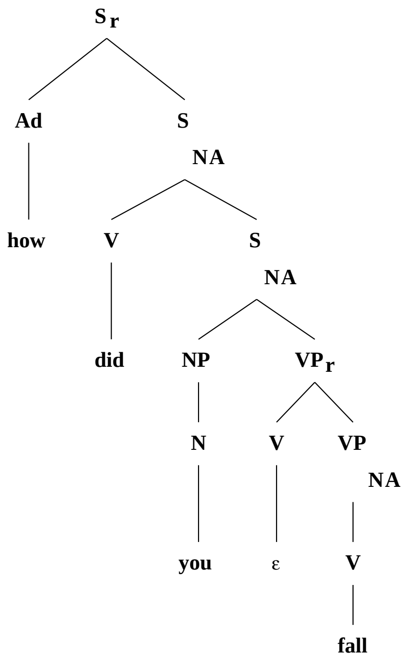


Figure 19.6: Derived tree for *How did you fall?*

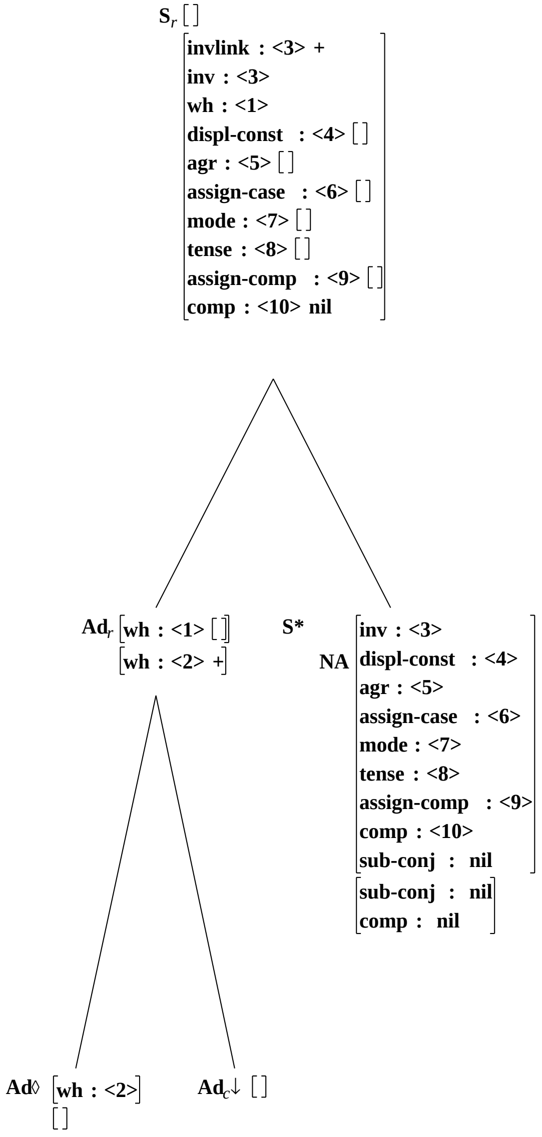


Figure 19.7: Complex adverb phrase modifier: βARBarbs

There is one more adverb modifier tree in the grammar which is not included in Table 19.2. This tree, shown in Figure 19.7, has a complex adverb phrase and is used for wh+ two-adverb phrases that occur sentence initially, such as in sentence (193). Since *how* is the only wh+ adverb, it is the only adverb that can anchor this tree.

(193) How quickly did Srini fix the problem?

Chapter 20

Auxiliaries

Although there has been some debate about the lexical category of auxiliaries, the English XTAG grammar follows [McCawley, 1988], [Haegeman, 1991], and others in classifying auxiliaries as verbs. The category of verbs can therefore be divided into two sets, main or lexical verbs, and auxiliary verbs, which can cooccur in a verbal sequence. Only the highest verb in a verbal sequence is marked for tense and agreement regardless of whether it is a main or auxiliary verb. Some auxiliaries (*be*, *do*, and *have*) share with main verbs the property of having overt morphological marking for tense and agreement, while the modal auxiliaries do not. However, all auxiliary verbs differ from main verbs in several crucial ways.

- Multiple auxiliaries can occur in a single sentence, while a matrix sentence may have at most one main verb.
- Auxiliary verbs cannot occur as the sole verb in the sentence, but must be followed by a main verb.
- All auxiliaries precede the main verb in verbal sequences.
- Auxiliaries do not subcategorize for any arguments.
- Auxiliaries impose requirements on the morphological form of the verbs that immediately follow them.
- Only auxiliary verbs invert in questions (with the sole exception in American English of main verb *be*[1]).
- An auxiliary verb must precede sentential negation (e.g. $*$*John not goes*).
- Auxiliaries can form contractions with subjects and negation (e.g. *he'll*, *won't*).

The restrictions that an auxiliary verb imposes on the succeeding verb limits the sequence of verbs that can occur. In English, sequences of up to five verbs are allowed, as in sentence (194).

[1] Some dialects, particularly British English, can also invert main verb *have* in yes/no questions (e.g. *Have you any Grey Poupon?*). This is usually attributed to the influence of auxiliary *have*, coupled with the historic fact that English once allowed this movement for all verbs.

(194) The music should have been being played [for the president].

The required ordering of verb forms when all five verbs are present is:

modal base perfective progressive passive

The rightmost verb is the main verb of the sentence. While a main verb subcategorizes for the arguments that appear in the sentence, the auxiliary verbs select the particular morphological forms of the verb to follow each of them. The auxiliaries included in the English XTAG grammar are listed in Table 20.1 by type. The third column of Table 20.1 lists the verb forms that are required to follow each type of auxiliary verb.

TYPE	LEX ITEMS	SELECTS FOR
modals	*can, could, may, might, will,* *would, ought, shall, should*	base form[2] (e.g. *will go, might come*)
perfective	*have*	past participle (e.g. *has gone*)
progressive	*be*	gerund (e.g. *is going, was coming*)
passive	*be*	past participle (e.g. *was helped by Jane*)
do support	*do*	base form (e.g. *did go, does come*)
infinitive to	*to*	base form (e.g. *to go, to come*)

Table 20.1: Auxiliary Verb Properties

20.1 Non-inverted sentences

This section and the sections that follow describe how the English XTAG grammar accounts for properties of the auxiliary system described above.

In our grammar, auxiliary trees are added to the main verb tree by adjunction. Figure 20.1 shows the adjunction tree for non-inverted sentences.[3]

The restrictions outlined in column 3 of Table 20.1 are implemented through the features <**mode**>, <**perfective**>, <**progressive**> and <**passive**>. The syntactic lexicon entries for the auxiliaries gives values for these features on the foot node (VP*) in Figure 20.1. Since the top features of the foot node must eventually unify with the bottom features of the node it adjoins onto for the sentence to be valid, this enforces the restrictions made by the auxiliary

[2] There are American dialects, particularly in the South, which allow double modals such as *might could* and *might should*. These constructions are not allowed in the XTAG English grammar.

[3] We saw this tree briefly in section 4.4.3.2, but with most of its features missing. The full tree is presented here.

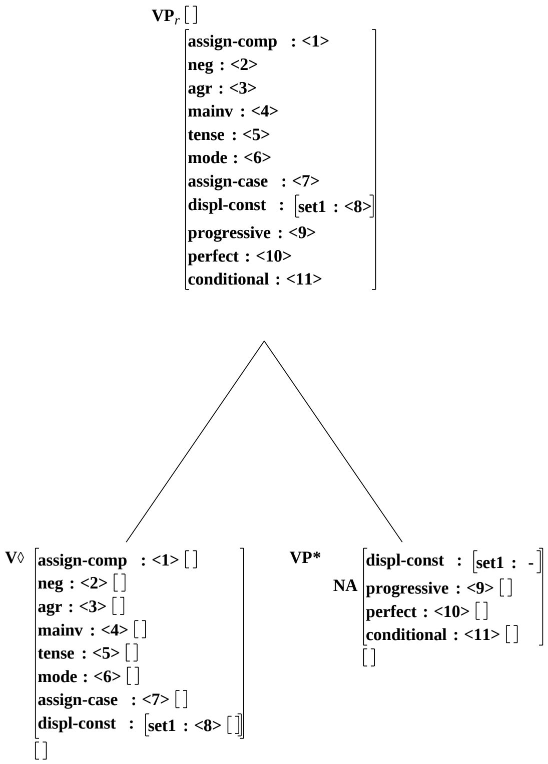


Figure 20.1: Auxiliary verb tree for non-inverted sentences: βVvx

node. In addition to these feature values, each auxiliary also gives values to the anchoring node (V$\diamond$), to be passed up the tree to the root VP (VP$_r$) node; there they will become the new features for the top VP node of the sentential tree. Another auxiliary may now adjoin on top of it, and so forth. These feature values thereby ensure the proper auxiliary sequencing. Figure 20.2 shows the auxiliary trees anchored by the four auxiliary verbs in sentence (194). Figure 20.3 shows the final tree created for this sentence.

The general English restriction that matrix clauses must have tense (or be imperatives) is enforced by requiring the top S-node of a sentence to have **<mode>=ind/imp** (indicative or imperative). Since only the indicative sentences have tense, non-tensed clauses are restricted to occurring in embedded environments.

20.2 Inverted Sentences

In inverted sentences, the two trees shown in Figure 20.4 adjoin to an S tree anchored by a main verb. The tree in Figure 20.4(a) is anchored by the auxiliary verb and adjoins to the S node, while the tree in Figure 20.4(b) is anchored by an empty element and adjoins at the VP node. Figure 20.5 shows these trees (anchored by *will*) adjoined to the declarative transitive tree[4] (anchored by main verb *buy*).

[4] The declarative transitive tree was seen in section 6.2.

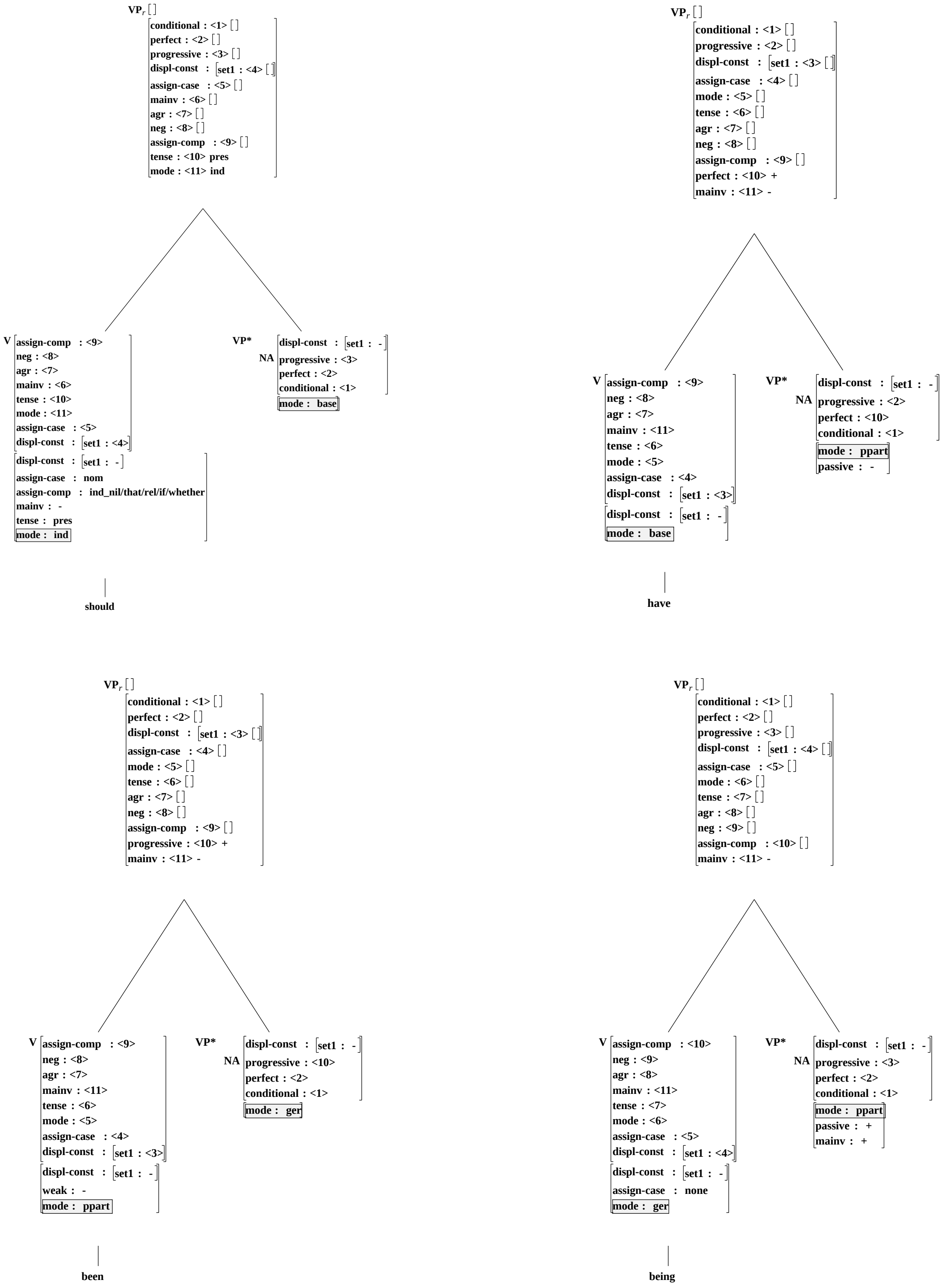


Figure 20.2: Auxiliary trees for *The music should have been being played.*

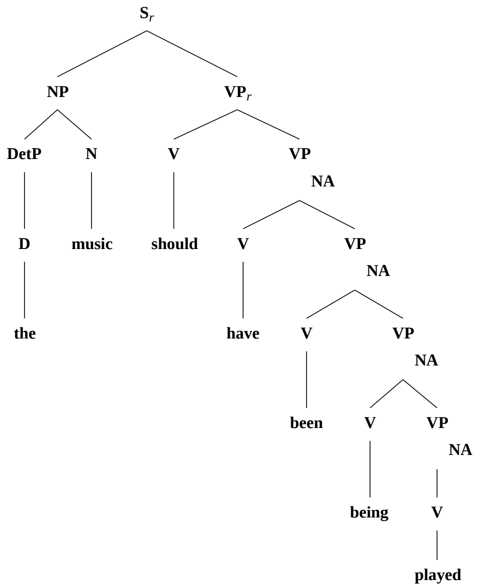
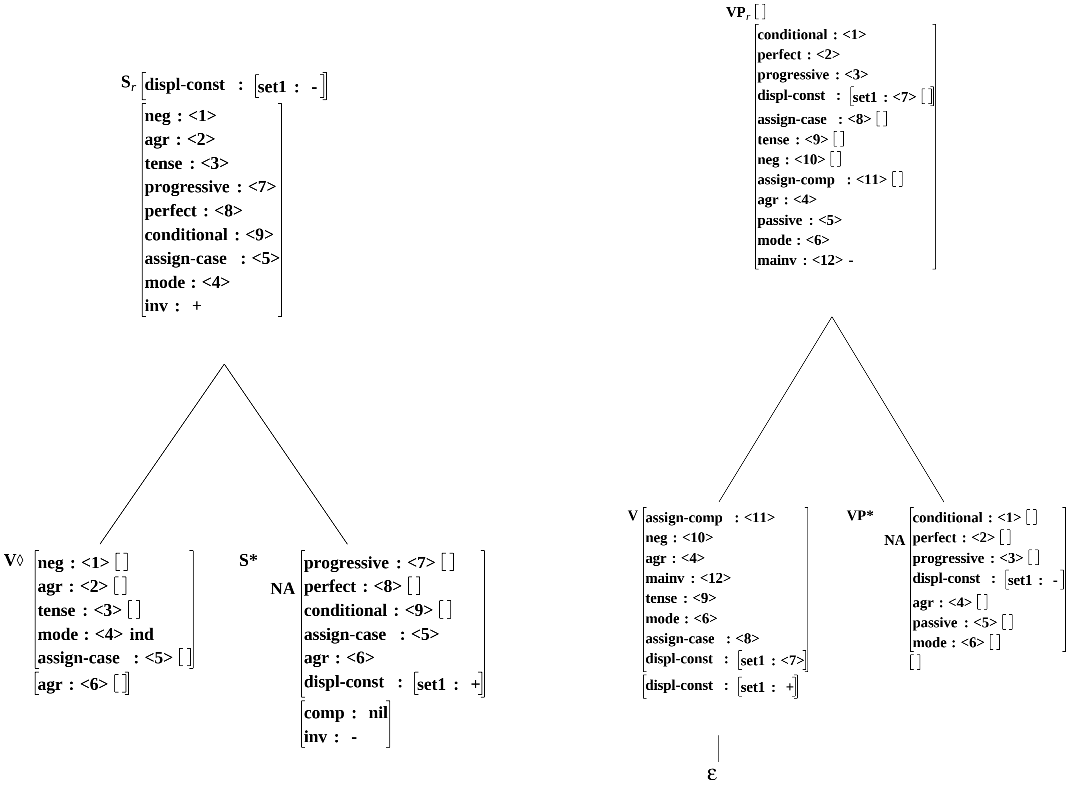


Figure 20.3: *The music should have been being played.*



(a) (b)

Figure 20.4: Trees for auxiliary verb inversion: βVs (a) and βVvx (b)

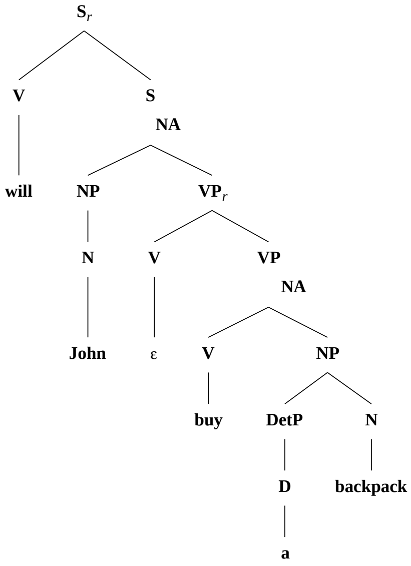


Figure 20.5: *Will John buy a backpack?*

The feature <**displ-const**> ensures that both of the trees in Figure 20.4 must adjoin to an elementary tree whenever one of them does. For more discussion on this mechanism, which simulates tree local multi-component adjunction, see [Hockey and Srinivas, 1993]. The tree in Figure 20.4(b), anchored by ϵ, represents the originating position of the inverted auxiliary. Its adjunction blocks the <**assign-case**> values of the VP it dominates from being co-indexed with the <**case**> value of the subject. Since <**assign-case**> values from the VP are blocked, the <**case**> value of the subject can only be co-indexed with the <**assign-case**> value of the inverted auxiliary (Figure 20.4(a)). Consequently, the inverted auxiliary functions as the case-assigner for the subject in these inverted structures. This is in contrast to the situation in uninverted structures where the anchor of the highest (leftmost) VP assigns case to the subject (see section 4.4.3.2 for more on case assignment). The XTAG analysis is similar to GB accounts where the inverted auxiliary plus the ϵ-anchored tree are taken as representing I to C movement.

20.3 Do-Support

It is well-known that English requires a mechanism called 'do-support' for negated sentences and for inverted yes-no questions without auxiliaries.

(195) John does not want a car.

(196) $*$John not wants a car.

(197) John will not want a car.

(198) Do you want to leave home?

(199) $*$Want you to leave home?

(200) Will you want to leave home?

20.3.1 In negated sentences

The GB analysis of do-support in negated sentences hinges on the separation of the INFL and VP nodes (see [Chomsky, 1965], [Jackendoff, 1972] and [Chomsky, 1986]). The claim is that the presence of the negative morpheme blocks the main verb from getting tense from the INFL node, thereby forcing the addition of a verbal lexeme to carry the inflectional elements. If an auxiliary verb is present, then it carries tense, but if not, periphrastic or 'dummy', *do* is required. This seems to indicate that *do* and other auxiliary verbs would not cooccur, and indeed this is the case (see sentences (201)-(202)). Auxiliary *do* is allowed in English when no negative morpheme is present, but this usage is marked as emphatic. Emphatic *do* is also not allowed to cooccur with auxiliary verbs (sentences (203)-(206)).

(201) ∗We will have do bought a sleeping bag.

(202) ∗We do will have bought a sleeping bag.

(203) You **do** have a backpack, don't you?

(204) I **do** want to go!

(205) ∗You **do** can have a backpack, don't you?

(206) ∗I **did** have had a backpack!

In the XTAG grammar, *do* is prevented from cooccurring with other auxiliary verbs by a requirement that it adjoin only onto main verbs. It has indicative mode, so no other auxiliaries can adjoin above it.[5] The lexical item *not* is only allowed to adjoin onto a non-indicative (and therefore non-tensed) verb. Since all matrix clauses must be indicative (or imperative), a negated sentence will fail unless an auxiliary verb, either *do* or another auxiliary, adjoins somewhere above the negative morpheme, *not*. In addition to forcing adjunction of an auxiliary, this analysis of *not* allows it freedom to move around in the auxiliaries, as seen in the sentences (207)-(210).

(207) John will have had a backpack.

(208) ∗John not will have had a backpack.

(209) John will not have had a backpack.

(210) John will have not had a backpack.

[5] Earlier, we said that indicative mode carries tense with it. Since only the topmost auxiliary carries the tense, any subsequent verbs must **not** have indicative mode.

20.3.2 In inverted yes/no questions

In inverted yes/no questions, *do* is required if there is no auxiliary verb to invert, as seen in sentences (198)-(200), replicated here as (211)-(213).

(211) Do you want to leave home?

(212) ∗Want you to leave home?

(213) Will you want to leave home?

(214) ∗Do you will want to leave home?

In English, unlike other Germanic languages, the main verb cannot move to the beginning of a clause, with the exception of main verb *be*.[6] In a GB account of inverted yes/no questions, the tense feature is said to be in C^0 at the front of the sentence. Since main verbs cannot move, they cannot pick up the tense feature, and do-support is again required if there is no auxiliary verb to perform the role. Sentence (214) shows that *do* does not interact with other auxiliary verbs, even when in the inverted position.

In XTAG, trees anchored by a main verb that lacks tense are required to have an auxiliary verb adjoin onto them, whether at the VP node to form a declarative sentence, or at the S node to form an inverted question. *Do* selects the inverted auxiliary trees given in Figure 20.4, just as other auxiliaries do, so it is available to adjoin onto a tree at the S node to form a yes/no question. The mechanism described in section 20.3.1 prohibits *do* from cooccurring with other auxiliary verbs, even in the inverted position.

20.4 Infinitives

The infinitive *to* is considered an auxiliary verb in the XTAG system, and selects the auxiliary tree in Figure 20.1. *To*, like *do*, does not interact with the other auxiliary verbs, adjoining only to main verb base forms, and carrying infinitive mode. It is used in embedded clauses, both with and without a complementizer, as in sentences (215)-(217). Since it cannot be inverted, it simply does not select the trees in Figure 20.4.

(215) John wants to have a backpack.

(216) John wants Mary to have a backpack.

(217) John wants for Mary to have a backpack.

The usage of infinitival *to* interacts closely with the distribution of null subjects (PRO), and is described in more detail in section 8.5.

[6] The inversion of main verb *have* in British English was previously noted.

Chapter 21

Conjunction

21.1 Introduction

The XTAG system can handle sentences with conjunction of two constituents of the same syntactic category. There are eight syntactic categories that can conjoin, and in each case an auxiliary tree is used to implement the conjunction. These eight categories can be considered as four different cases, as described in the following sections. In all cases the two constituents are required to be of the same syntactic category, but there may also be some additional constraints, as described below.

21.2 Adjective, Adverb, Preposition and PP Conjunction

Each of these four categories has an auxiliary tree that is used for conjunction of two constituents of that category. The auxiliary tree adjoins into the left-hand-side component, and the right-hand-side component substitutes into the auxiliary tree.

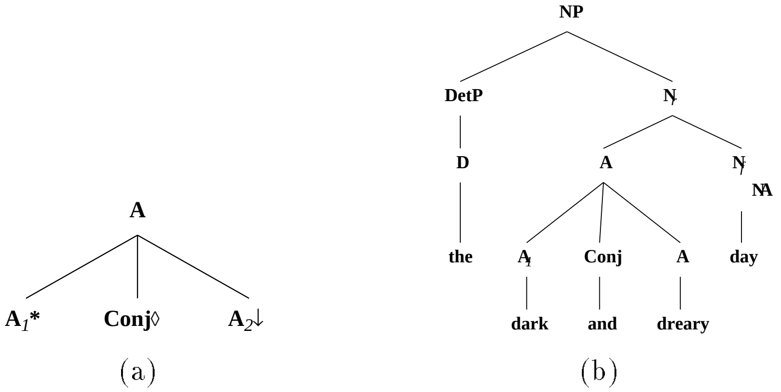


Figure 21.1: Tree for adjective conjunction: βA1conjA2 and a resulting parse tree

Figure 21.1(a) shows the auxiliary tree for adjective conjunction, and is used, for example, in the derivation of the parse tree for the noun phrase *the dark and dreary day*, as shown in Figure 21.1(b). The auxiliary tree adjoins onto the node for the left adjective, and the right adjective substitutes into the right hand side node of the auxiliary tree. The analysis for adverb, preposition and PP conjunction is exactly the same and there is a corresponding auxiliary tree for each of these that is identical to that of Figure 21.1(a) except, of course, for the node labels.

21.3 Noun Phrase and Noun Conjunction

The tree for NP conjunction, shown in Figure 21.2, has the same basic analysis as in the previous section except that the <**wh**> and <**case**> features are used to force the two noun phrases to have the same <**wh**> and <**case**> values. This allows, for example, *he and she wrote the book together* while disallowing ∗*he and her wrote the book together.* The <**agr**> feature of the top node sets the resulting NP to have plural number. The tree for N conjunction is identical to that for the NP tree except for the node labels.

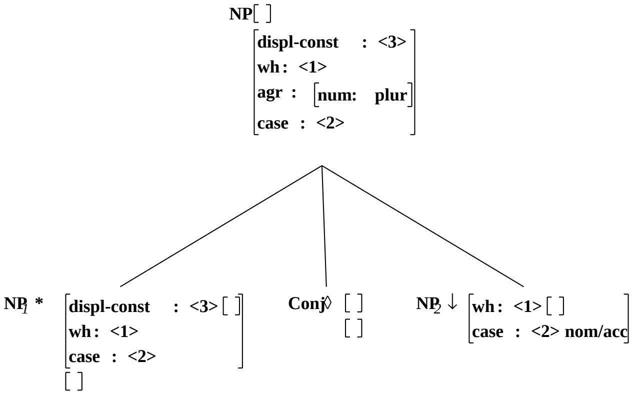


Figure 21.2: Tree for NP conjunction: βNP1conjNP2

21.4 Sentential Conjunction

The tree for sentential conjunction, shown in Figure 21.3, is based on the same analysis as the conjunctions in the previous two sections, with a slight difference in features. The <**mode**> feature[1] is used to constrain the two sentences being conjoined to have the same mode so that *the day is dark and the phone never rang* is acceptable, but ∗*the day dark and the phone never rang* is not. The <**assign-comp**> feature[2] feature is used to allow conjunction of infinitival sentences, such as *to read and to sleep is a good life.*

21.5 Determiner Conjunction

The tree for determiner conjunction, shown in Figure 21.4, is unlike the other conjunction trees in that the foot node is on the right. This is because determiner phrases generally build to the

[1] See section 8.3 for an explanation of the <**mode**> feature.

[2] See section 8.5 for an explanation of the <**assign-comp**> feature.

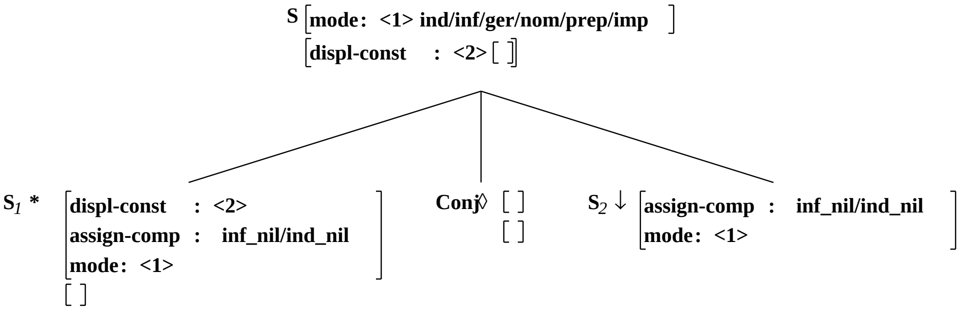

S
mode: <1> ind/inf/ger/nom/prep/imp
displ-const : <2> [ ]
S1 *
displ-const : <2>
assign-comp : inf_nil/ind_nil
mode: <1>
[ ]
Conj◊ [ ]
[ ]
S2 ↓
assign-comp : inf_nil/ind_nil
mode: <1>

Figure 21.3: Tree for sentential conjunction: βS1conjS2

left. For the same reason, all the various feature values are taken from the left determiner, and the only requirement is that the <**wh**> feature is the same, while the other features, such as <**card**>, are unconstrained. For example, *which and what* and *all but one* are both acceptable determiner conjunctions, but $*$*which and all* is not.

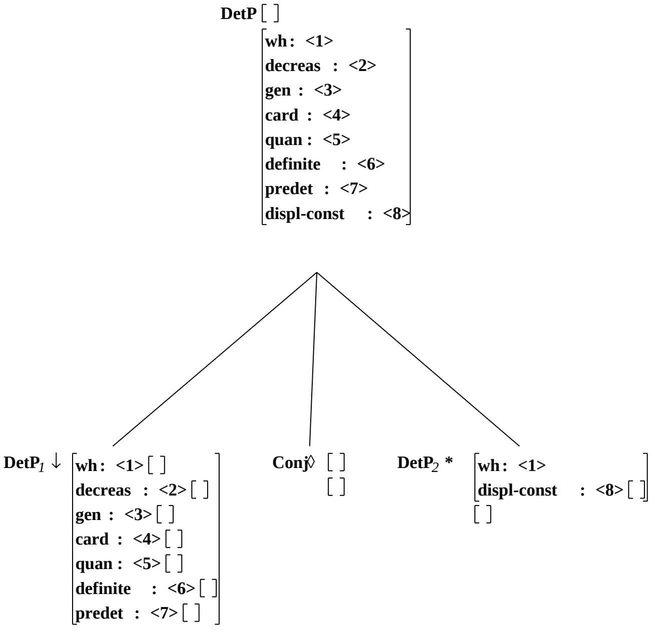


Figure 21.4: Tree for determiner conjunction: βDX1conjDX2

21.6 Other Conjunctions

The conjunction analysis described in the previous sections is designed to handle only the most straightforward cases of conjunction. Three types of conjunction that are not handled are:

- **Incomplete Constituents:** Although the sentence *John likes and Bill hates bananas* is a simple case of sentential conjunction, it cannot be handled by the current XTAG grammar. Since *likes* anchors a tree that needs both a subject noun phrase and an object noun phrase to be substituted in, the latter sentence would need have an unfilled substitution node after *John likes* for the sentence to parse.
- **Verb Phrase Conjunction:** Since verbs anchor a tree with a root node of type S and not VP, there is no straightforward way to implement verb phrase conjunction. For example, in the sentence *John eats cookies and drinks beer*, there is no point in the derivation at which *eats cookies* and *drinks beer* are available as separate trees ready to be conjoined. They are both only subtrees in their respective S trees. This could also be considered as a case of incomplete constituents, since *drinks beer* is missing a noun phrase.
- **Gapping:** Sentences such as *John likes apples and Bill pears* are also not handled by the previous analysis. These could also be considered as a case of incomplete constituents.

One grammar formalism that is capable of handling these types of conjunction is Combinatory Categorial Grammar (CCG) ([Steedman, 1990]) which relies on a nonstandard notion of a constituent in order to accomplish this. Proposals have been made (e.g. [Joshi and Schabes, 1991]), inspired by the CCG approach, to handle these problematic cases in the FB-LTAG formalism. Unlike the CCG analysis, however, the traditional notion of constituents and phrase structure is maintained. Such proposals are as of yet unimplemented.

Part V

Appendices

Appendix A

Future Work

A.1 Adjective ordering

At this point, the treatment of adjectives in the XTAG English grammar does not include selectional or ordering restrictions.[1] Consequently, any adjective can adjoin onto any noun and on top of any other adjective already modifying a noun. All of the modified noun phrases shown in (218)-(221) currently parse.

(218) big green bugs

(219) big green ideas

(220) colorless green ideas

(221) *green big ideas

While (219)-(221) are all semantically anomalous, (221) also suffers from an ordering problem that makes it seem ungrammatical as well. Since the XTAG grammar focuses on syntactic constructions, it should accept (218)-(220) but not (221). Both the auxiliary and determiner ordering systems are structured on the idea that certain types of lexical items (specified by features) can adjoin onto some types of lexical items, but not others. We believe that an analysis of adjectival ordering would follow the same type of mechanism.

A.2 Determiner Adverbs

There are some apparent adverbs that interact with the NP and determiner system ([Quirk *et al.*, 1985]), although there is some debate in the literature as to whether these should be classified as determiners or adverbs.[2]

(222) **Hardly** any attempt was made at restitution.

(223) **Only** Albert would say such a thing.

[1] This section is a repeat of information found in section 19.1.

[2] This section is from [Hockey and Egedi, 1994].

(224) **Almost** all the people had left by 5pm.

Adverbs that modify NP's or determiners have restrictions on what types of NP's or determiners they can modify. They divide into three classes based on the pattern of these restrictions. The adverbs *especially*, *even*, *just* and *only* form a class that can modify any NP that is <**wh**>=–, including proper nouns. A second class, consisting of adverbs such as *hardly*, *merely* and *simply*, modifies NP's with determiners that are <**definite**>=– and <**const**>=+, or that are <**gen**>=+. This second class of adverbs can also modify NP's with *the* as a determiner, but they do not modify NP's without determiners. The third class, exemplified by *almost*, *approximately* and *relatively*, modifies the determiner itself. These adverbs are restricted to modifying determiners with the <**card**>=+ feature, as well as *all*, *double* and *half*. The distinction between adverbs that modify NP's and ones that modify determiners can be seen in the NP's in (225) and (226).

(225) [Just][half the people]

(226) [Approximately half][the people]

A.3 More work on Determiners

In addition to the analysis described in Chapter 18, there remains work to be done to complete the analysis of determiner constructions in English.[3] Although constructions such as determiner coordination are easily handled if overgeneration is allowed, blocking sequences such as *one and some* while allowing sequences such as *five or ten* still remains to be worked out. There are still a handful of determiners that are not currently handled by our system. We do not have an analysis to handle *most*, *such*, *certain*, *other* and *own*[4]. In addition, there is a set of lexical items that we consider adjectives (*enough*, *less*, *more* and *much*) that have the property that they cannot cooccur with determiners. We feel that a complete analysis of determiners should be able to account for this phenomenon, as well.

A.4 Comparatives

Also included in our future grammar development plans are comparatives. Comparatives that involve ellipsis would require a general solution of the problem of representing ellipsis, but simpler comparatives without ellipsis, such as *fewer than nine* in (227), should be amenable to analysis as complex determiners, perhaps with trees similar in construction to the partitive and genitive NP trees.

(227) Cats have **fewer than** nine lives.

[3] This section is from [Hockey and Egedi, 1994].

[4] The behavior of *own* is sufficiently unlike other determiners that it most likely needs a tree of its own, adjoining onto the right-hand side of genitive determiners.

A.5 Time NP's

Although in general NP's cannot simply adjoin onto sentences, there is a class of NP's, called Time NP's, that can. These NP's behave essentially like PP's, and the XTAG analysis for this is fairly simple, requiring only the creation of proper NP auxiliary trees. Only slightly more difficult is the identification of all possible anchors of these trees. A **<time>=+** feature will be used to ensure that only certain nouns can select the time NP auxiliary trees.

(228) I went to Kentucky last month/*big cat.

(229) This morning/*Big cat, we practiced juggling four balls.

A.6 *-ing* adjectives

An analysis has already been provided for *-ed* adjectives (as in sentence (230)), which are restricted to the Transitive Verb family.[5] A similar analysis needs to take place for the *-ing* adjectives. This type of adjective, however, does not seem to be as restricted as the *-ed* adjectives, since verbs in other tree families seem to exhibit this alternation as well (e.g. sentences (231) and (232)).

(230) The murdered man was a doctoral student at UPenn.

(231) The man died.

(232) The dying man pleaded for his life.

A.7 Punctuation

We are currently developing an analysis of comma coordination, to cover sentences such as (233) and (234).

(233) The brilliant, funny and timeless comedian had his 99th birthday today.

(234) NP's, PP's and VP's are all adjunction sites.

Beyond this, we intend to add an analysis of other types of punctuation. We believe that there are cases where the punctuation will serve as a guide to the correct parse (e.g. comma following topicalized element), thus reducing ambiguity.

A.8 PRO control

Within the FB-LTAG formalism, PRO-control is an interesting problem because of the intrinsic non-local nature of control.[6] The controller NP and the controlled PRO are always in different clauses. In this sense, Control is even more non-local than Binding.

In the literature on Control, two types are often distinguished: obligatory control, as in sentences (235) and (236), and optional control, as in sentence (237).

[5] This analysis may need to be extended to the Transitive Verb particle family as well.

[6] This section is taken from [Bhatt, 1994].

(235) Jan$_i$ promised Maria [PRO$_i$ to go].

(236) Jan persuaded Maria$_i$ [PRO$_i$ to go].

(237) [PRO$_{arb}$ to dance] is important.

An analysis for obligatory control has been worked out, although it has yet to be implemented. The NP anchored by PRO will have the feature <**control**>=+. The <**control**> feature is also introduced in trees that can take sentential arguments. Depending on the verb, the control propagation paths in the auxiliary trees are different. In the case of subject control (as in sentence (235)), the subject NP and the foot node are constrained to have the same control features, while for object control (e.g. sentence (236)), the object NP and the foot node are constrained to have the same control features.

Work has also been done on an XTAG analysis for optional control, but this has not been fully worked out yet.

A.9 Verb selectional restrictions

Although we explicitly do not want to model semantics in the XTAG grammar, there is some work along the syntax/semantics interface that would help reduce syntactic ambiguity and thus decrease the number of semantically anomalous parses. In particular, verb selectional restrictions, particularly for PP arguments and adjuncts, would be quite useful. With the exception of the required *to* in the Ditransitive with PP Shift tree family (Tnx0Vnx1Pnx2), any preposition is allowed in the tree families that have prepositions as their arguments. In addition, there are no restrictions as to which prepositions are allowed to adjoin onto a given verb. The sentences in (238)-(240) are all currently accepted by the XTAG grammar. Their violations are stronger than would be expected from purely semantic violations, however, and the presence of verb selectional restrictions on PP's would keep these sentences from being accepted.

(238) #Survivors walked of the street.

(239) #The man about the earthquake survived.

(240) #The president arranged on a meeting.

A.10 Thematic Roles

Elementary trees in TAGs capture several notions of locality, with the most primary of these being locality of θ-role assignment. Each elementary tree has associated with it the θ-roles assigned by the anchor of that elementary tree. In the current XTAG system, while the notion of locality of θ-role assignment within an elementary tree has been implicit, the θ-roles assigned by a head have not been explicitly represented in the elementary tree. Incorporating θ-role information will make the elementary trees more informative and will enable efficient pruning of spurious derivations when embedded into a specific context. In the case of a Synchronous TAG, θ-roles can also be used to automatically establish links between two elementary trees, one in the object language and one in the target language.

A.11 Idioms

An analysis of idioms has already been worked out ([Abeille and Schabes, 1989]), and one idiom tree family is contained in the English XTAG grammar (Transitive Idioms; section 6.3). What remains to be done is a wide-ranging cataloging of the many English idioms. The list of idioms must then be divided into appropriate tree families, based on the construction of the idiom, the elements that are frozen and must therefore anchor the trees, and the alternations that the idiom can undergo (e.g. passive, wh-movement, etc). This work has not been done.

Appendix B

Tree Naming conventions

The various trees within the XTAG grammar are named more or less according to the following tree naming conventions. Although these naming conventions are generally followed, there are occasional trees that do not strictly follow these conventions.

B.1 Tree Families

Tree families are named according to the basic declarative tree structure in the tree family (see section B.2), but with a T as the first character instead of an α or β.

B.2 Trees within tree families

Each tree begins with either an α (alpha) or a β (beta) symbol, indicating whether it is an initial or auxiliary tree, respectively. Following an α or a β the name may additionally contain one of:

I	imperative
E	ergative
N0,1,2	relative clause{position}
G	NP gerund
D	Determiner gerund
pW0,1,2	wh-PP extraction{position}
W0,1,2	wh-NP extraction{position}

Numbers are assigned according to the position of the argument in the declarative tree, as follows:

0	subject position
1	first argument (e.g. direct object)
2	second argument (e.g. indirect object)

The body of the name consists of a string of the following components, which corresponds to the leaves of the tree. The anchor(s) of the trees is(are) indicated by capitalizing the part of speech corresponding to the anchor.

s	sentence
a	adjective
arb	adverb
be	*be*
x	phrasal category
d	determiner
v	verb
lv	light verb
conj	conjunction
comp	complementizer
it	*it*
n	noun
p	preposition
pl	particle
by	*by*
neg	negation

As an example, the transitive declarative tree consists of a subject NP, followed by a verb (which is the anchor), followed by the object NP. This translates into αnx0Vnx1. If the subject NP had been extracted, then the tree would be αW0nx0Vnx1. A passive tree with the *by* phrase in the same tree family would be αnx1Vbynx0. Note that even though the object NP has moved to the subject position, it retains the object encoding (nx1).

B.3 Assorted Initial Trees

Trees that are not part of the tree families are generally gathered into several files for convenience. The various initial trees are located in `lex.trees`. All the trees in this file should begin with an α, indicating that they are initial trees. This is followed by the root category which follows the naming conventions in the previous section (e.g. n for noun, x for phrasal category). The root category is in all capital letters. After the root category, the node leaves are named, beginning from the left, with the anchor of the tree also being capitalized. As an example, the αNXdxN tree is rooted by an NP node (NX), with a determiner phrase subnode (dx), and anchored by a noun (N). This tree is shown in Figure B.1.

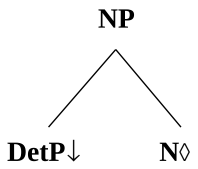


Figure B.1: NP with determiner tree: αNXdxN

B.4 Assorted Auxiliary Trees

Most auxiliary trees are contained in `modifiers.trees`, although a couple of other files also contain auxiliary trees. The auxiliary trees follow a slightly different naming convention from the initial trees. Since the root and foot nodes must be the same for the auxiliary trees, the root nodes are not explicitly mentioned in the names of auxiliary trees. The trees are named according to the leaf nodes, starting from the left, and capitalizing the anchor node. All auxiliary trees begin with a β, of course. For example, βARBs, indicates a tree anchored by an adverb (ARB), that adjoins onto the left of an S node (Note that S must be the foot node, and therefore also the root node).

Appendix C

Features

Table C.1 contains a comprehensive list of the features in the XTAG grammar and their possible values.

Feature	Value
<agr 3rdsing>	+, −
<agr num>	plur,sing
<agr pers>	1,2,3
<assign-case>	nom,acc,none
<assign-comp>	that,whether,if,for,rel,inf_nil,ind_nil
<card>	+, −
<case>	nom,acc,none
<comp>	that,whether,if,for,rel,inf_nil,ind_nil
<const>	+, −
<conditional>	+, −
<decreas>	+, −
<definite>	+, −
<displ-const>	+, −
<extracted>	+, −
<gen>	+, −
<inv>	+, −
<invlink>	
<mainv>	+, −
<mode>	base,ger,ind,inf,imp,nom,ppart,prep,sbjunt
<neg>	+, −
<passive>	+, −
<perfect>	+, −
<pred>	+, −
<progressive>	+, −
<pron>	+, −
<quan>	+, −
<sub-conj>	ind1,ind2,ind3,inf1,inf2,ger,nil
<tense>	pres,past
<trace>	no value, indexing only
<trans>	+, −
<wh>	+, −

Table C.1: List of features and their possible values

Appendix D

Evaluation and Results

In this appendix, we show that the XTAG grammar, which uses only corpus-independent statistical information, can match the performance of induced probabilistic grammars as well as other rule-based grammars. Given the fact that the grammar has not been fine-tuned to the test data, we predict still better results if it were tailored to the domain of the test data.

XTAG has recently been used to parse Wall Street Journal[1], IBM manual, and ATIS corpora as a means of evaluating the coverage and correctness of XTAG parses. For this evaluation, a sentence is considered to have parsed if the correct parse is among the parses generated by XTAG. Verifying the presence of the correct parse among the generated parses is done manually at present by random sampling. Preliminary results without the use of parse ranking are shown in Table D.1. It is worth emphasizing that the XTAG grammar is truly wide-coverage and has not been fine-tuned to any particular genre, unlike many other grammars.

Corpus	# of Sentences	% Parsed	Av. # of Parses/Sent
WSJ	18,730	41.22 %	7.46
IBM Manual	2040	75.42%	6.12
ATIS	524	88.35%	6.0

Table D.1: Performance of XTAG on various corpora

Performance on the WSJ corpus is lower relative to IBM and ATIS due to the wide-variety of syntactic constructions used. Even grammars induced on the partially bracketed WSJ corpus have fairly low performance (e.g. 57.1% sentence accuracy for [Schabes *et al.*, 1993]).

D.1 Comparison with IBM

A more detailed experiment to measure the crossing bracket accuracy of the XTAG-parsed IBM-manual sentences has been performed. In this experiment, XTAG parses of 1100 IBM-manual

[1] Sentences of length ≤ 15 words.

sentences have been ranked using certain heuristics. The ranked parses have been compared[2] against the bracketing given in the Lancaster Treebank of IBM-manual sentences[3]. Table D.2 shows the results of XTAG obtained in this experiment, which used the highest ranked parse for each system. It also shows the results of the latest IBM statistical grammar ([Jelinek *et al.*, 1994]) on the same genre of sentences. Only the highest-ranked parse of both systems was used for this evaluation. Crossing Brackets is the percentage of sentences with no pairs of brackets crossing the Treebank bracketing (i.e. ((a b) c) has a crossing bracket measure of one if compared to (a (b c))). Recall is the ratio of the number of constituents in the XTAG parse to the number of constituents in the corresponding Treebank sentence. Precision is the ratio of the number of correct constituents to the total number of constituents in the XTAG parse.

System	# of sentences	Crossing Bracket Accuracy	Recall	Precision
XTAG	1100	81.29%	82.34%	55.37%
IBM Statistical grammar	1100	86.20%	86.00%	85.00%

Table D.2: Performance of XTAG on IBM-manual sentences

As can be seen from Table D.2, the precision figure for the XTAG system is considerably lower than that for IBM. For the purposes of comparative evaluation against other systems, we had to use the same crossing-brackets metric though we believe that the crossing-brackets measure is inadequate for evaluating a grammar like XTAG. There are two reasons for the inadequacy. First, the parse generated by XTAG is much richer in its representation of the internal structure of certain phrases than those present in manually created treebanks (e.g. IBM: [$_N$ your personal computer], XTAG: [$_{NP}$ [$_G$ your] [$_N$ [$_N$ personal] [$_N$ computer]]]). This is reflected in the number of constituents per sentence, shown in the last column of Table D.3.[4]

System	Sent. Length	# of sent	Av. # of words/sent	Av. # of Constituents/sent
XTAG	1-10	654	7.45	22.03
	1-15	978	9.13	30.56
IBM Stat. Grammar	1-10	447	7.50	4.60
	1-15	883	10.30	6.40

Table D.3: Constituents in XTAG parse and IBM parse

A second reason for considering the crossing bracket measure inadequate for evaluating XTAG is that the primary structure in XTAG is the derivation tree from which the bracketed tree is derived. Two identical bracketings for a sentence can have completely different derivation trees (e.g. *kick the bucket* as an idiom vs. a compositional use). A more direct measure of the performance of XTAG would evaluate the derivation structure, which captures the dependencies between words.

[2] We used the parseval program written by Phil Harison (phil@atc.boeing.com).

[3] The Treebank was obtained through Salim Roukos (roukos@watson.ibm.com) at IBM.

[4] We are aware of the fact that increasing the number of constituents also increases the recall percentage. However we believe that this a legitimate gain.

D.2 Comparison with Alvey

We also compared XTAG to the Alvey Natural Language Tools (ANLT) Grammar, and found that the two performed comparably. We parsed the set of LDOCE Noun Phrases presented in Appendix B of the technical report ([Carroll, 1993]) using XTAG. Table D.4 summarizes the results of this experiment. A total of 143 noun phrases were parsed. The NPs which did not have a correct parse in the top three derivations were considered failures for either system. The maximum and average number of derivations columns show the highest and the average number of derivations produced for the NPs that have a correct derivation in the top three. We show the performance of XTAG both with and without the tagger since the performance of the POS tagger is significantly degraded on the NPs because the NPs are usually shorter than the sentences on which it was trained. It would be interesting to see if the two systems performed similarly on a wider range of data.

System	# of NPs	# parsed	% parsed	Maximum derivations	Average derivations
ANLT Parser	143	127	88.81%	32	4.57
XTAG Parser with POS tagger	143	93	65.03%	28	3.45
XTAG Parser without POS tagger	143	120	83.91%	28	4.14

Table D.4: Comparison of XTAG and ANLT Parser

D.3 Comparison with CLARE

We have also compared the performance of XTAG against that of the CLARE-2 system ([Alshawi *et al.*, 1992]) on the ATIS corpus. Table D.5 shows the performance results. The percentage parsed column for both systems represents the percentage of sentences that produced any parse. It must be noted that the performance result shown for CLARE-2 is without any tuning of the grammar for the ATIS domain. The performance of CLARE-3, a later version of the CLARE system, is estimated to be 10% higher than that of the CLARE-2 system.[5]

System	Mean length	% parsed
CLARE-2	6.53	68.50%
XTAG	7.62	88.35%

Table D.5: Performance of CLARE-2 and XTAG on the ATIS corpus

In an attempt to compare the performance of the two systems on a wider range of sentences (from similar genres), we provide in Table D.6 the performance of CLARE-2 on LOB corpus and the performance of XTAG on the WSJ corpus. The performance was measured on sentences of up to 10 words for both systems.

[5] When CLARE-3 is tuned to the ATIS domain, performance increases to 90%. However XTAG has not been tuned to the ATIS domain.

System	Corpus	Mean length	% parsed
CLARE-2	LOB	5.95	53.40%
XTAG	WSJ	6.00	55.58%

Table D.6: Performance of CLARE-2 and XTAG on LOB and WSJ corpus respectively

It can be seen from the above comparisons that XTAG system performs comparably to CLARE on texts with limited and varied range of sentence phenomena.

Bibliography

[Abeille and Schabes, 1989] Anne Abeille and Yves Schabes. Parsing Idioms in Lexicalized TAGs. In *Proceedings of EACL '89*, pages 161–65, 1989.

[Abeillé *et al.*, 1990] Anne Abeillé, Kathleen Bishop, Sharon Cote, and Yves Schabes. *A Lexicalized Tree Adjoining Grammar for English.* Technical report, University of Pennsylvania, 1990.

[Abeillé, 1990] Anne Abeillé. French and English Determiners: Interaction of Morphology, Syntax, and Semantics in Lexicalized Tree Adjoining Grammars. In *TAG+ Workshop - Abstracts*, University of Pennsylvania, 1990.

[Akmajian, 1970] A. Akmajian. On deriving cleft sentences from pseudo-cleft sentences. *Linguistic Inquiry*, 1:149–168, 1970.

[Alshawi *et al.*, 1992] Hiyan Alshawi, David Carter, Richard Crouch, Steve Pullman, Manny Rayner, and Arnold Smith. *CLARE – A Contextual Reasoning and Cooperative Response Framework for the Core Language Engine.* SRI International, Cambridge, England, 1992.

[Ball, 1991] Catherine N. Ball. *The historical development of the it-cleft.* PhD thesis, University of Pennsylvania, 1991.

[Baltin, 1989] Mark Baltin. Heads and projections. In Mark Baltin and Anthony S. Kroch, editors, *Alternative conceptions of phrase structure*, pages 1–16. Chicago, Illinois, 1989.

[Bhatt, 1994] Rajesh Bhatt. Pro-control in tags. Unpublished paper, University of Pennsylvania, 1994.

[Browning, 1987] Marguerite Browning. *Null Operator Constructions.* PhD thesis, MIT, 1987.

[Burzio, 1986] Luigi Burzio. *Italian syntax. A Government-Binding approach.* Studies in natural language and linguistic theory. Reidel, Dordrecht, 1986.

[Carroll, 1993] John Carroll. *Practical Unification-based Parsing of Natural Language.* University of Cambridge, Computer Laboratory, Cambridge, England, 1993.

[Chomsky and Lasnik, 1993] Noam Chomsky and Howard Lasnik. Untitled. 1993.

[Chomsky, 1965] Noam Chomsky. *Aspects of the Theory of Syntax.* MIT Press, Cambridge, Massachusetts, 1965.

[Chomsky, 1970] Noam Chomsky. Remarks on Nominalization. In *Readings in English Transformational Grammar*, pages 184–221. Ginn and Company, Waltham, Massachusetts, 1970.

[Chomsky, 1986] Noam Chomsky. *Barriers.* MIT Press, Cambridge, Massachusetts, 1986.

[Chomsky, 1992] Noam Chomsky. A Minimalist Approach to Linguistic Theory. *MIT Working Papers in Linguistics*, Occasional Papers in Linguistics No. 1, 1992.

[Church, 1988] Kenneth Ward Church. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In *2nd Applied Natural Language Processing Conference*, Austin, Texas, 1988.

[Cinque, 1990] G. Cinque. *Types of A-bar-Dependencies.* MIT Press, Cambridge, Massachusetts; London, England, 1990.

[Cowie and Mackin, 1975] A. P. Cowie and R. Mackin, editors. *Oxford Dictionary of Current Idiomatic English*, volume 1. Oxford University Press, London, England, 1975.

[Delahunty, 1984] Gerald P. Delahunty. The Analysis of English Cleft Sentences. *Linguistic Analysis*, 13(2):63–113, 1984.

[Delin, 1989] Judy L. Delin. *Cleft Constructions in Discourse.* PhD thesis, University of Edinburgh, 1989.

[Doran *et al.*, 1994] Christy Doran, Dania Egedi, Beth Ann Hockey, B. Srinivas, and Martin Zaidel. XTAG System - A Wide Coverage Grammar for English. In *Proceedings of the* 17^{th} *International Conference on Computational Linguistics (COLING '94)*, Kyoto, Japan, August 1994.

[Egedi and Martin, 1994] Dania Egedi and Patrick Martin. A Freely Available Syntactic Lexicon for English. In *Proceedings of the International Workshop on Sharable Natual Language Resources*, Nara, Japan, August 1994.

[Emonds, 1970] J. Emonds. *Root and structure-preserving transformations.* PhD thesis, Massachusetts Institute of Technology, 1970.

[Ernst, 1983] T. Ernst. More on Adverbs and Stressed Auxiliaries. *Linguistic Inquiry 13*, pages 542–48, 1983.

[Gazdar *et al.*, 1985] G. Gazdar, E. Klein, G. Pullum, and I. Sag. *Generalized Phrase Structure Grammar.* Harvard University Press, Cambridge, Massachusetts, 1985.

[Grimshaw, 1990] Jane Grimshaw. *Argument Structure.* MIT Press, Cambridge, Massachusetts, 1990.

[Haegeman, 1991] Liliane Haegeman. *Introduction to Government and Binding Theory.* Blackwell Publishers, Cambridge, Massachusetts, Oxford, U.K., 1991.

[Hale and Keyser, 1986] Ken Hale and Samuel Jay Keyser. Some transitivity alternations in English. Technical Report #7, MITCCS, July 1986.

[Hale and Keyser, 1987] Ken Hale and Samuel Jay Keyser. A view from the middle. Technical Report #10, MITCCS, January 1987.

[Hanks, 1979] Patrick Hanks, editor. *Collins Dictionary of the English Language*. Collins, London, England, 1979.

[Heycock and Kroch, 1993] Caroline Heycock and Anthony S. Kroch. Verb movement and the status of subjects: implications for the theory of licensing. *GAGL*, 36:75–102, 1993.

[Heycock, 1991] Caroline Heycock. Progress Report: The English Copula in a LTAG. Internal XTAG project report, University of Pennsylvania, Summer 1991.

[Hockey and Egedi, 1994] Beth Ann Hockey and Dania Egedi. A Syntactic Account of Determiner Ordering using Lexicalized TAGs. In *Proceedings of the 3rd TAG+ Conference*, Paris, France, 1994.

[Hockey and Srinivas, 1993] Beth Ann Hockey and B. Srinivas. Feature-Based TAG in Place of Multi-component Adjunction: Computational Implications. In *Proceedings of the Natural Language Processing Pacific Rim Symposium (NLPRS)*, Fukuoka, Japan, December 1993.

[Hockey, 1994] Beth Ann Hockey. Echo questions, intonation and focus. In *Proceedings of the Interdisciplinary Conference on Focus and Natural Language Processing in Celebration of the 10th anniversary of the Journal of Semantics*, Eschwege, Germany, 1994.

[Hornby, 1974] A. S. Hornby, editor. *Oxford Advanced Learner's Dictionary of Current English*. Oxford University Press, London, England, third edition, 1974.

[Iatridou, 1991] Sabine Iatridou. *Topics in Conditionals*. PhD thesis, MIT, 1991.

[Jackendoff, 1972] R. Jackendoff. *Semantic Interpretation in Generative Grammar*. MIT Press, Cambridge, Massachusetts, 1972.

[Jackendoff, 1990] R. Jackendoff. On Larson's Analysis of the Double Object Construction. *Linguistic Inquiry*, 21:427–456, 1990.

[Jelinek *et al.*, 1994] F. Jelinek, J. Lafferty, D. Magerman, R. Mercer, A. Ratnaparkhi, and S. Roukos. Decision Tree Parsing using a Hidden Derivation Model. *ARPA Workshop on Human Language Technology*, pages 260–265, 1994.

[Joshi and Schabes, 1991] Aravind K. Joshi and Yves Schabes. Fixed and Flexible Phrase Structure: Coordination in Tree Adjoining Grammars. In *DARPA Workshop on Spoken Language Systems*, February 1991.

[Joshi *et al.*, 1975] Aravind K. Joshi, L. Levy, and M. Takahashi. Tree Adjunct Grammars. *Journal of Computer and System Sciences*, 1975.

[Joshi, 1985] Aravind K. Joshi. Tree Adjoining Grammars: How much context Sensitivity is required to provide a reasonable structural description. In D. Dowty, I. Karttunen, and A. Zwicky, editors, *Natural Language Parsing*, pages 206–250. Cambridge University Press, Cambridge, U.K., 1985.

[Kaplan and Bresnan, 1983] Ronald Kaplan and Joan Bresnan. Lexical-functional Grammar: A Formal System for Grammatical Representation. In J. Bresnan, editor, *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge, Massachusetts, 1983.

[Karp *et al.*, 1992] Daniel Karp, Yves Schabes, Martin Zaidel, and Dania Egedi. A Freely Available Wide Coverage Morphological Analyzer for English. In *Proceedings of the* 15^{th} *International Conference on Computational Linguistics (COLING '92)*, Nantes, France, August 1992.

[Kennan and Stavi, 1986] E. L. Kennan and J. Stavi. A Semantic Characterization of Natural Language Determiners. *Linguistics and Philosophy*, 9, August 1986.

[Knowles, 1986] John Knowles. The Cleft Sentence: A Base-Generated Perspective. *Lingua: International Review of General Linguistics*, 69(4):295–317, August 1986.

[Kroch and Joshi, 1985] Anthony S. Kroch and Aravind K. Joshi. The Linguistic Relevance of Tree Adjoining Grammars. Technical Report MS-CIS-85-16, Department of Computer and Information Science, University of Pennsylvania, 1985.

[Kroch and Joshi, 1987] Anthony S. Kroch and Aravind K. Joshi. Analyzing Extraposition in a Tree Adjoining Grammar. In G. Huck and A. Ojeda, editors, *Discontinuous Constituents, Syntax and Semantics*, volume 20. Academic Press, 1987.

[Lapointe, 1980] S. Lapointe. A lexical analysis of the English auxiliary verb system. In T. Hoekstra et al, editor, *Lexical Grammar*, pages 215–254. Foris, Dordrecht, 1980.

[Larson, 1988] R. Larson. On the Double Object construction. *Linguistic Inquiry*, 19:335–391, 1988.

[Larson, 1990] R. Larson. Double Objects Revisited: Reply to Jackendoff. *Linguistic Inquiry*, 21:589–612, 1990.

[Lasnik and Saito, 1984] H. Lasnik and M. Saito. On the Nature of Proper Government. *Linguistic Inquiry*, 15:235–289, 1984.

[Lasnik and Uriagereka, 1988] H. Lasnik and J. Uriagereka. *A Course in GB Syntax.* MIT Press, Cambridge, Massachusetts, 1988.

[Liberman, 1989] Mark Liberman. Tex on tap: the ACL Data Collection Initiative. In *Proceedings of the DARPA Workshop on Speech and Natural Language Processing*. Morgan Kaufman, 1989.

[McCawley, 1988] James D. McCawley. *The Syntactic Phenomena of English.* The University of Chicago Press, Chicago, Illinois, 1988.

[Moro, 1990] A. Moro. *There* as a Raised Predicate. Paper presented at GLOW, 1990.

[Napoli, 1988] Donna Jo Napoli. Ergative Verbs in English. *Journal of the Linguistic Society of America (Language)*, 64:1(1):130–142, March 1988.

[Obernauer, 1984] H. Obernauer. On the Identification of Empty Categories. *Linguistic Review 4*, 2:153–202, 1984.

[Partee *et al.*, 1990] Barbara Partee, Alice ter Meulen, and Robert E. Wall. *Mathematical Methods in Linguistics.* Kluwer Academic Publishers, 1990.

[Pollard and Sag, 1987] Carl Pollard and Ivan A. Sag. *Information-Based Syntax and Semantics. Vol 1: Fundamentals.* CSLI, 1987.

[Pollock, 1989] J-Y. Pollock. Verb Movement, UG, and the Structure of IP. *Linguistic Inquiry*, 20.3:365–424, 1989.

[Quirk *et al.*, 1985] Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. *A Comprehensive Grammar of the English Language.* Longman, London, 1985.

[Rizzi, 1990] Luigi Rizzi. *Relativized Minimality.* MIT Press, Cambridge, Massachusetts; London, England, 1990.

[Rosenbaum, 1967] Peter S. Rosenbaum. *The grammar of English predicate complement constructions.* MIT press, Cambridge, Massachusetts, 1967.

[Sag *et al.*, 1985] I. Sag, G. Gazdar, T. Wasow, and S. Weisler. Coordination and How to distinguish categories. *Natural Language and Linguistic Theory*, 3:117–171, 1985.

[Schabes *et al.*, 1988] Yves Schabes, Anne Abeillé, and Aravind K. Joshi. Parsing strategies with 'lexicalized' grammars: Application to Tree Adjoining Grammars. In *Proceedings of the* 12*th* *International Conference on Computational Linguistics (COLING'88)*, Budapest, Hungary, August 1988.

[Schabes *et al.*, 1993] Y. Schabes, M. Roth, and R. Osborne. Parsing the Wall Street Journal with the Inside-Outside Algorithm. In *Proceedings of the European ACL*, 1993.

[Schabes, 1990] Yves Schabes. *Mathematical and Computational Aspects of Lexicalized Grammars.* PhD thesis, Computer Science Department, University of Pennsylvania, 1990.

[Seuss, 1971] Dr. Seuss. *The Lorax.* Random House, New York, New York, 1971.

[Soong and Huang, 1990] Frank K. Soong and Eng-Fong Huang. Fast Tree-Trellis Search for Finding the N-Best Sentence Hypothesis in Continuous Speech Recognition. *Journal of Acoustic Society, AM.*, May 1990.

[Sornicola, 1988] R. Sornicola. IT-clefts and WH-clefts: two awkward sentence types. *Journal of Linguistics*, 24:343–379, 1988.

[Steedman, 1990] Mark Steedman. Gapping as Constituent Coordination. *Linguistics and Philosophy*, 13:207–263, 1990.

[Vijay-Shanker and Joshi, 1991] K. Vijay-Shanker and Aravind K. Joshi. Unification Based Tree Adjoining Grammars. In J. Wedekind, editor, *Unification-based Grammars.* MIT Press, Cambridge, Massachusetts, 1991.

[Watanabe, 1993] Akira Watanabe. The Notion of Finite Clauses in AGR-Based Case Theory. *MIT Working Papers in Linguistics*, 18:281–296, 1993.