ON THE GENERATION AND LINEARIZATION OF
MULTI-DOMINANCE STRUCTURES

by

Joan Chen-Main

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

Baltimore, Maryland

September, 2006

© Joan Chen-Main

All Rights Reserved
ABSTRACT

Some common constructions such as wh-questions and coordinated constructions seem to allow lexical elements to play multiple grammatical roles typically associated with distinct positions. For example, in What did Emmy eat?, what is usually assumed to function as the object of eat even though it appears sentence initially rather than in the canonical object position. In Joe bakes and Sam sells cookies, a single noun phrase, cookies, satisfies both verbs’ need for an object. Traditionally, this apparent “multiple-linking” is attributed to co-indexing distinct elements filling multiple positions, only one of which is pronounced. Alternatively, a multiply-linked element can be conceptualized as an element immediately dominated by multiple parent nodes. Under such a multi-dominance approach, trees no longer suffice for representing the immediate dominance relation. Rather, the set of syntactic structures is expanded to include non-tree graphs.

This dissertation examines how such multi-dominance structures are generated and how their terminals are linearized. The generation question is answered by adopting the node-contraction operation, originally introduced into the Tree Adjoining Grammar (TAG) formalism to allow for coordination. This work takes node-contraction to be a general mechanism in the TAG system: node-contraction is involved not only in generating coordination, but also cases traditionally dealt with via movement. The existence of island effects that prohibit movement from certain domains indicates that we must also specify when multi-dominance cannot occur. This work also shows that by placing certain locality restrictions on node contraction at the derivational level, a number of these island effects can be derived.
The linearization question is answered by taking the primitive relations in syntactic structures to be immediate dominance and sister-precedence, an ordering between two siblings, and using a modification of the Non-Tangling Condition on trees to derive ordering among terminals. However, the proposed process for deriving ordering information does not guarantee a linearization of terminals for every graph. A graph may be unlinearizable due to either lack of ordering information or conflicting ordering information. This work also explores formal and linguistic consequences in a multi-dominance system that result from taking linearizability to be a property of well-formed syntactic structures.

Advisor: Dr. Robert Frank
ACKNOWLEDGEMENTS

This dissertation was written in a department that modeled intellectual generosity and critical thinking. I am grateful to the entire Department of Cognitive Science for my time here.

A number of individuals, however, must be singled out for their particularly significant contributions, the first of whom is Bob Frank. In Bob, I found an advisor rich in ideas and a teacher gifted in conveying complex concepts in an accessible manner. It is difficult to conceive how this dissertation could have been possible were it not for his guidance, encouragement, patience, and generous feedback, and the open office door granting access to all these.

I also wish to thank the other linguistically minded faculty, in order of appearance. Bill Badecker’s introductory course first interested me in exploring other courses offered by the department. Géraldine Legendre helped with navigating my undergraduate years, my first taste of linguistics research, and the graduate school application process, and has continued to provide helpful advice. Paul Smolensky provided thought-provoking conversations both in and out of the classroom, especially in his role as advisor for my second qualifying project and a member of my committee. Paul’s obvious concern for the success of all the grad students and his diligence in shaping us to be able to conceive, evaluate, and present theories will continue to influence us long after we leave the department. Finally, during his long service as department chair, Luigi Burzio guided the department through situations ranging from mundane to sticky, all while carrying on an active research program and carefully attending to students’ work.

Acknowledgement is also due to my outside committee members. I am the beneficiary of Aravind Joshi’s labors three-fold, once through his thought-provoking published work, once through his investment in Bob, and once through his kind attention to my own work. Sincere thanks also to Ed Scheinerman for his genuine curiosity in seeking out “math problems in cognitive science” and to Justin Halberda for stepping in on short notice.
A special thank you goes to Barb Fisher for her cheerful contributions to life in the department and her magical ability to axe through red tape.

I am keenly aware that any ability to receive the training offered in graduate school is built on the efforts of the teachers who came before. Again, several must be singled out. From my childhood and time in the public school system, they are, in order of appearance, Wenpeng Chen and Wei-mei Ching, Ellen Sakai, Dave Savage, Susan Wildstrom, and Lynn Gruner. From my time as an undergraduate, they are Dana Boatman and Géraldine Legendre, both of whom pushed me to begin the transition from classroom student to researcher.

For adding happiness to graduate life in the form of soccer games, recitals, barbeques, ice cream breaks, photo discussions, celebrations, good conversation, and shared enthusiasm for good Chinese food, my thanks goes to my classmates and friends in the department. Uyen Le, Tamara Nicol Medina, Niki Tantalou, Manny Vindiola, and especially Gaja Jarosz Snover and Julia Yarmolinskaya were the co-authors of particularly cherished times. I wish also to mention the alumni of the department for their example and advice, the XTAG group at University of Pennsylvania for their contagious curiosity, and Faith Christian Fellowship for consistent reminders of the world outside academia.

Finally, I must acknowledge my family, because their investment in me began before anyone else’s and continues even now. The sacrifices of my grandparents, aunts, and uncles made it possible for my parents, Wenpeng Chen and Wei-mei Ching, to come to America, seeking a different life. Heartfelt thanks to Alice Chen and Jane Wang for making me feel known and loved. Lastly, I am immensely grateful to Joe Main, who knew that love and sacrifice would be two sides of the same coin when we tied our lives together, and I am thankful for my Washington State family.

This dissertation is built on the efforts of many people, yet even so, there is no claim in what follows that I could stake my life on. I do, however, stake my life on the grace of God, and so, this work is dedicated to Him.
### TABLE OF CONTENTS

ABSTRACT ...........................................................................................................................................ii

ACKNOWLEDGEMENTS ...............................................................................................................iv

CHAPTER 1: INTRODUCTION .............................................................................................................1

CHAPTER 2: PRELIMINARIES
2.1 INTRODUCTION ..................................................................................................................11
2.2 PHENOMENA APPEARING FREQUENTLY IN THIS THESIS
   2.2.1 Coordination ..................................................................................................................11
   2.2.2 Movement .....................................................................................................................13
   2.2.3 Islands ........................................................................................................................14
2.3 TREE ADJOINING GRAMMAR BASICS
   2.3.1 Introduction ................................................................................................................17
   2.3.2 Combinatory Operations: substitution and adjoining ..................................................19
   2.3.3 Derivation structures in TAG .......................................................................................21
   2.3.4 Elementary trees ..........................................................................................................25
   2.3.5 Movement in TAG ......................................................................................................27

PART I: GENERATION OF SYNTACTIC GRAPHS

CHAPTER 3: MULTI-DOMINANCE TREE ADJOINING GRAMMAR
3.1 INTRODUCTION ...................................................................................................................30
3.2 GENERATION OF COORDINATE STRUCTURES: SARKAR AND JOSHI (1996)
   3.2.1 The Problem for Classic TAG ..................................................................................30
   3.2.2 The Node Contraction Operation ..............................................................................31
   3.2.3 Conditions on Node Contraction ..............................................................................37
   3.2.4 Derivation Structures ..............................................................................................39
3.3 GENERATION OF QUESTIONS ..........................................................................................40
3.4 GENERATION OF INTERLEAVED CONSTRUCTIONS .......................................................43
3.5 ALLOWING PHRASE Sized TREES ..................................................................................50
3.6 CONCLUSION ......................................................................................................................57
6.2.3.1 Order preservation between sister nodes ........................................125
6.2.3.2 Scandinavian object shift: order preservation in use .........................127
   6.2.3.2.1 Ternary branching ............................................................132
   6.2.3.2.2 Binary branching ..............................................................133

6.2.4 Formal properties of antisymmetric graphs
   6.2.4.1 Restriction to binary branching and self-c-commanding multi-
            dominated nodes ......................................................................134
   6.2.4.2 Necessary conditions for satisfying antisymmetry .........................138
   6.2.4.3 Counterexample to the converse of Theorem ..............................143

6.2.5 Remnant movement and the antisymmetry requirement ......................145
6.2.6 The Mandarin reflexive ziji: remnant movement in use .......................150

6.3 THE TOTALITY REQUIREMENT
   6.3.1 Overview ..................................................................................168
   6.3.2 The need for non-sentential coordination ......................................169
6.3.3 Properties of total graphs ...............................................................173
   6.3.3.1 Insufficient restrictions: binary branching and self-c-commanding
            multi-dominated nodes ..............................................................174
   6.3.3.2 Insufficient restrictions: binary branching and the edge condition
            ...............................................................................................175
   6.3.3.3 Sufficient restrictions: Binary branching, self-c-commanding
            multi-dominated nodes, and the edge condition ..........................178

6.4 CONCLUSION ....................................................................................186
6.5 DIVERSION 1: THE NEED FOR AN ANAPHORIC APPROACH TO GAPPING .......187
6.6 DIVERSION 2: DELETION AS A REPAIR?
   6.6.1 Lasnik’s (2001) Proposal and Linearization Violations ....................193
   6.6.2 Deletion as Antisymmetry Violation Evasion: Pseudogapping ..........194
   6.6.3 Deletion as Totality Violation Evasion: Island Violations ................196
   6.6.4 Remaining Issues .........................................................................198

CHAPTER 7: LINEARIZABLE SYNTACTIC GRAPHS AND GENERATIVE
            COMPLEXITY
7.1 INTRODUCTION ..............................................................................201
7.2 BEYOND CONTEXT-FREE LANGUAGES ...........................................202
7.3 BEYOND FOUR COUNTING DEPENDENCIES ................................205
7.4 SCRAMBLING ..................................................................................210
7.5 CONCLUSION ..................................................................................212

REFERENCES .......................................................................................216

VITA ......................................................................................................223
Some common constructions, such as the coordinated structure in (1) and the *wh*-question in (2), seem to allow lexical elements to play multiple grammatical roles that are usually associated with distinct positions.

(1) \textit{Joe bakes ___ and Sam sells cookies.}

(2) \textit{What did Emmy eat ___?}

In (1), \textit{bakes} and \textit{sells} both take two arguments, one playing the grammatical role of a subject and the other, an object. While two distinct noun phrases satisfy each verb’s need for a subject, a single noun phrase, \textit{cookies}, somehow satisfies both verbs’ need for an object. In (2), \textit{what} appears at the beginning of the phrase, but to maintain the parallelism between the related declarative \textit{Emmy did eat an apple}, \textit{what} is also typically assumed to function as the object of \textit{eat}. To maintain the widely held view that verbs and arguments are in some local configuration, we must posit that some mechanism allows \textit{what} to occupy both its pronounced position and the object position of \textit{eat}.

Under the standard assumption that syntactic structures are trees, it is not possible for a single structural element to occupy multiple structural locations. Consequently, the appearance of one element occupying multiple locations must be reanalyzed. One line of reanalysis involves ellipsis: the multiple positions are occupied by separate copies of the
same element, of which all but one are deleted prior to pronunciation. An analysis of (1) and (2) along these lines is shown in (3) and (4). Alternatively, the multiple positions of a single element may be taken to arise from a movement operation, which removes an element from its original position, leaving behind an empty element or trace, and reattaches it to a new position. This possibility is illustrated in (5) and (6). These treatments seek to maintain that each node may be immediately dominated by at most a single parent node.

(3) Ellipsis analysis of *Joe bakes ___ and Sam sells cookies.*

(4) Ellipsis analysis of *What did Emmy eat ___?*
Movement analysis of *Joe bakes ___ and Sam sells cookies.*

Movement analysis of *What did Emmy eat ___?*

The claim that structures such as (5) and (6) are bona fide trees, however, is not strictly true. Even for the structures in (3) and (4), one might argue that the identity requirement between the parallel elements is also a form of identity among nodes. The co-indexing of nodes is essentially a mechanism for allowing an element to be immediately dominated by more than one parent. The realization that multiply linked syntactic structures do not meet the definition of trees is reflected in the use of the notion
of a *chain* to refer to a moved element and its traces as a single syntactic object (Chomsky 1981) and the characterization of syntactic structures not as trees, but as sets (of sets) (Chomsky 1995). Whereas the phrase structure rules of early generative grammar have clear graph-theoretic analogues, the relationship between these later characterizations and graphs is suppressed. For example, even though it is not obvious how a configuration of chains can be represented using only nodes and edges, the building blocks of graphs, the finished product of a derivation that combines chains can be recast as a composite of nodes and edges. In fact, the possibility that the way in which sets (of sets) are used in syntactic theory may inadvertently reconstruct graph theory is expressed as a concern (Gärtner 1997). These are certainly not arguments that syntactic objects cannot be chains or sets (of sets). They are, however, an expression of doubt that syntactic structure can be or should be disentangled from graph theory in alternate representations.

This work is an intentional return to the notion that syntactic structures should be represented at the most basic level with nodes and edges, and an invitation to import useful ideas and results from the study of graphs. What this proposal leaves behind is the one-parent-per-node restriction that characterizes trees. Instead, nodes are allowed to be immediately dominated by multiple parent nodes. Under such a *multi-dominance* approach, the set of syntactic structures is expanded to include non-tree graphs, and structures such as (7) and (8) are possible analyses of (1) and (2).
Multi-dominance analysis of Joe bakes ____ and Sam sells cookies.

Multi-dominance analysis of What did Emmy eat ____?

Multi-dominance has been explored by a number of researchers, though not everyone necessarily considers their view of syntactic structure to be a graph theoretic view. Peters and Ritchie’s (1981) phrase linking grammar is a variety of multi-dominance syntax in which two types of immediate dominance are possible and a node is allowed to have a parent in both relations. The structures used by Goodall (1987) to analyze coordination allow a single lexical item to be part of multiple conjuncts. Later, Gärtner’s (1997) close examination of the widely followed Minimalist Program
(Chomsky 1995) led to a proposal to replace the operations *Merge*, which combines two syntactic objects and forms a single combined object, and *Move*, which duplicates part of a syntactic object and merges it with the original object, with a single hybrid operation whose application allowed multi-dominated structures. In 2001, both Starke and Chomsky recast Move as a special case of Merge. Starke (2001) argued that Move could be reduced to a special case of Merge applied to non-adjacent nodes, and Chomsky (2001) introduced the terms *External Merge*, which merges nodes that have not been merged before, and *Internal Merge*, which re-merges a node that has previously been merged, resulting in multiply dominated nodes. (For additional work on multi-dominance, also see Moltmann 1992, Wilder 2001, Abels 2004, Citko 2005, among others).

In 2006, then, to formalize syntactic structures as graphs is not a particularly radical move. Rather, this work should be seen as an attempt to make explicit and precise the characteristics of syntactic structures, having recognized that syntactic structures permit multi-dominance. Here, I pursue the claim that syntactic structures are directed graphs that meet certain well-formedness conditions, and that these conditions allow some non-tree syntactic structures. Let us refer to the subset of graphs that meet these conditions as *syntactic graphs*.¹

If the multi-dominance approach is to be taken seriously as an alternative, a number of questions must be addressed. What introduces multi-dominance into the system? How are the terminals linearized? What kind of graphs are possible syntactic structures? This thesis examines two major questions. Part One (chapters 3 and 4) addresses how syntactic graphs are generated. Part Two (chapters 5, 6, and 7) address

¹ Terms used in other multi-dominance proposals include: *linked trees* (Peters and Ritchie 1981), *parallel structures* (Goodall 1987) and *Multiconstituent/Multidominance structures* (Gärtner 1997).
how syntactic graphs are pronounced. Other issues, such as the well-formedness
conditions on syntactic graphs, are addressed under the umbrella of generation or
linearization. Each part devotes one chapter to a general mechanism that answers the
question and one chapter to the formal and/or linguistic power of the proposed
mechanism.

As mentioned above, a body of work has already begun to address questions
raised by allowing multi-dominance. Though certainly related to and influenced by these
works, the work here differs in at least two ways. First, many of the authors exploring
multi-dominance have largely been interested in particular linguistic phenomena. Thus,
individual works have tended to focus on either constructions involving coordination or
constructions traditionally taken to involve movement. Coordination and movement \(^2\) are
not typically taken to involve the same mechanism, with their treatment in Combinatory
Categorial Grammar (Steedman 1996) being a notable exception. The first half of chapter
2 introduces the linguistic phenomena that I refer to throughout the course of this work,
and both coordination and movement are included. The first contribution of this work is
that it provides a unified mechanism for generating and linearizing both types of
constructions within a multi-dominance approach.

Second, much of the recent work is developed within the Minimalist Program and
many of the answers are specific to Minimalism. In contrast, the answer given here to the
generation question is couched within the Tree Adjoining Grammar framework (TAG)
(Joshi, Levy, and Takahashi 1975, Joshi 1985). TAG is a mathematically well-defined
formalism, and when applied to a theory of grammar, a number of attractive linguistic

\(^2\) The term ‘movement’ is drawn from transformational theory. There is actually no movement of elements
in CCG. In this paper, ‘movement’ will also be used only as a term to describe the constructions with
syntactic dependencies that were traditionally taken to result from actual movement of an element.
analyses can be stated simply. Additionally, a number of constraints on linguistic phenomena that need to be stipulated in other approaches follow naturally as a result of adopting the restrictions of TAG (Kroch and Joshi 1985). To the degree that the TAG approach aligns with other approaches (see, for example, Frank 2006), the answers in Part One may be translatable.

The TAG formalism provides a general answer to the question of how syntactic structures could be generated: kernel pieces of structure, elementary trees, are combined via two operations, adjoining and substitution, to create larger structures. An introduction to the basics of the TAG framework can be found in the second half of chapter 2. Further, an operation that generates graphs has, in fact, already been introduced into the TAG system to account for coordinate structures: the node contraction operation (Sarkar and Joshi 1996). Chapter 3 reviews the introduction of their node contraction operation and extends their approach to movement phenomena. The extension allows straightforward derivations for a class of constructions which were previously difficult for TAG and is also amenable to additional “factoring out of recursion,” allowing syntactic structures to be characterized as being built up out of smaller pieces of structure than are typically assumed in TAG. The second contribution of this work is a study of the outcome of extending node contraction to a general mechanism.

While node contraction tells us how multi-dominance enters the proposed system, the existence of island effects (Ross 1967), prohibitions against movement from certain constructions, indicates that we must also specify when multi-domination cannot occur. Chapter 4 shows that deriving a number of island effects observed in English can be
accomplished by placing certain locality restrictions on node contraction at the derivational level, which, in TAG, is distinct from the derived phrase structure. As the possibility of making direct reference to the derivational level is useful, the third contribution of this work is an argument in favor of a derivational level.

In Part Two, however, the answer given to the linearization question does not crucially hinge on adopting the TAG framework. Chapter 5 defines syntactic graphs as a formal object and presents a proposal for linearizing the terminals of the derived phrase-structure, which is not where TAG is unique. The phrase structures generated by a TAG system are generally assumed to be comparable to the type of structures assumed by mainstream generative grammarians. The proposal here takes the primitive relations of syntactic structures to be the immediate dominance relation between a parent node and a child node and the local precedence relation between sister nodes that share the same parent node. Global precedence is a relation defined only for terminals, the pronounced lexical items (Chomsky’s (1995) reinterpretation of Kayne (1994)). This ordering among terminals is derived via a modification of the Non-Tangling Condition, a well-formedness condition on trees (Partee et al., 1990). The outcome is a system in which local ordering is preserved in the global ordering. The fourth contribution of this work is a linearization process which depends only on a commitment to a graph theoretic representation of syntactic structure, not a commitment to a particular framework.

However, the proposed process for deriving precedence information does not guarantee a linearization of terminals for every graph. A graph may be unlinearizable due to either the lack of ordering information between some pair of terminals, totality violations, or conflicting orderings between some pair of terminals, antisymmetry.
violations. Following Kayne (1994), I take linearizability to be a property of well-formed syntactic structures. Chapter 6 discusses some of the effects of imposing a linearizability constraint on syntactic graphs. On the formal front, the conditions under which a graph satisfies antisymmetry and the conditions under which a graph satisfies totality are explored. On the linguistic front, the linearization requirement closes the door on some previously available analyses of some constructions (e.g. object \textit{wh}-questions and coordinated subjects) and opens the door for previously unavailable analyses of other phenomena (e.g. binding of Mandarin \textit{zi4ji3}). The order preservation effects observed in Scandinavian object shift are attributed to the linearizability requirement. The end of chapter 6 also includes a digression on a non-multi-dominance analysis of the gapping construction and the possibility of deletion as a repair strategy for linearization violations. As with the linearization process itself, this examination of the linearization requirement is compatible with many approaches to syntax.

Chapter 7 is a coda to Part Two in which we return to generation, but in combination with linearization. The chapter is a brief demonstration of the generative power of the modified TAG system when subject to the linearization requirement.
CHAPTER 2

PRELIMINARIES

2.1 INTRODUCTION

The theory presented here has two characteristics which are true of most theories. First, in order to carve out a manageable problem to address, each theory takes some set of data to be the core cases, cases that the theory straightforwardly provides an account for. Section 2.2 provides an introduction, painted in broad strokes, to several families of empirical data that will play a central role in this thesis, as well as some basic examples of each family. This list does not cover all of the phenomena to be discussed deeply or exhaustively, but additional characterizations and more complex patterns will be introduced as they become relevant. Second, most theories are informed and inspired by existing theories. The proposed framework here is an extension of an existing grammar formalism, Tree Adjoining Grammar (Joshi, Levy, and Takahashi 1975, Joshi 1985). The basics of TAG are presented in section 2.3.

2.2 PHENOMENA APPEARING FREQUENTLY IN THIS THESIS

2.2.1 COORDINATION

In coordinated constructions, two words, phrases, or clauses are connected together to form a larger syntactic unit. Typically, the two conjuncts are constituents, a word or a group of words that function together as a unit. The most straightforward cases of
coordination involve conjoining identical constituents to create a single constituent of the same type.

(1) a. Coordinated subject Determiner Phrases: \([John \ and \ [Mary]] \ ate \ cookies.\]
b. Coordinated object Determine Phrases: \(Joe \ eats \ [[cookies] \ and \ [ice \ cream]].\]
c. Coordinated Verb Phrases: \(Joe \ [[eats \ cookies] \ and \ [drinks \ tea]].\]
d. Coordinated Verbs: \(Joe \ [[bakes] \ and \ [decorates]] \ cookies.\]

However, coordination also can conjoin two apparent non-constituents.

(2) a. \([Joe \ bakes] \ and \ [Sam \ decorates] \ cookies.\]
b. \([Joe \ likes \ beans] \ and \ [Sam, \ rice].\]
c. \([You \ might \ not \ believe \ me], \ but \ [you \ will \ Bob].\]

It is possible, however, that the conjuncts in the constructions in (2) actually are constituents. These constructions have been analyzed as involving coordination of sentences with missing chunks. The analysis for each example is sketched in (3), and we will use the name of the analysis as the name for the phenomenon.

(3) a. Right Node Raising: \([Joe \ bakes \ _,] \ and \ [Sam \ decorates \ _,] \ cookies_.\]
b. Gapping: \([Joe \ likes, \ beans] \ and \ [Sam \ _, \ rice].\]
c. Pseudogapping: \([You \ might \ not \ believe, \ me], \ but \ [you \ will \ _, \ Bob].\]

One of the early approaches to coordination was that all cases of coordination could be derived from sentential coordination, sometimes followed by the application of
some sort of deletion rule. As Gleitman (1965) (and others) has pointed out, this cannot be the correct analysis for all cases of coordination.

(4)  *Ginger and Blue danced together ≠ Ginger danced together and Blue danced together.*

2.2.2 MOVEMENT

The term ‘movement’ describes a syntactic phenomena where elements in a sentence appear in a position that differs from their canonical position. For example, in (5), it seems that the *wh*-phrase *which sashimi* functions as the direct object of *order*, even though the direct object typically appears following the verb. In addition to an appeal to our intuitions, this conclusion is also supported by a paraphrase of (5) in which the *wh*-phrase appears in the canonical object position, the echo question in (6b).

(5)  

(a) [*Which sashimi*]$_i$ is$_j$ Tamara ___$_i$ planning to order ___$_i$?

(6)  

(a) *I think Tamara is planning to order the fugu sashimi!*

(b) *Tamara is planning to order which sashimi?*

This type of movement, where an argument moves to a non-argument position, is called A'-movement. A'-movement contrasts with A-movement, where an argument moves into another position for an argument. In (7), *Susie* is the subject of the verb *like* and the verb *seems*.

(7)  *Susie, seems [___] to like traveling*. 
Note that there is also a second type of movement in (5): the auxiliary verb *is* appears before the subject rather than after the subject as in (6b). *Is* is a head (i.e. a word that determines the syntactic type of the phrase it appears in) and it is typically analyzed as moving into another head position. The subject-aux inversion observed here falls in the family of head movement.

Further, a *wh*-element can be fronted even when it is an argument in a deeply embedded clause. This type of movement has typically been analyzed as involving intermediate landing sites, one in each clause, on the way from the canonical position to the surface position. This type of movement is what is meant by cyclic wh-movement.

(8) \[What, \textit{did} \textit{Julia \_} \textit{think [\_ \_ \textit{Yu-Chin said [\_ \_ \textit{Brian ate \_]}]]}\]?

These different types of movement can co-occur.

(9) Cyclic wh-movement, Head movement, A’-movement, and A- movement:

\[What, \textit{does} \textit{Bekah \_} \textit{think [\_ \_ \textit{Susie \_ seems \_ \_ \_ \textit{to like \_]}]}\]?

2.2.3 ISLANDS

It is well known that syntactic dependencies are subject to certain restrictions, including those that block extraction from certain domains. The syntactic configurations that block these dependencies have been dubbed *islands* (Ross 1967). For example, in a sentence with a relative clause, there is no general ban on movement, as shown by the subj-aux inversion in (10a). However, movement of an element that originates inside the relative clause is prohibited, as shown in (10b).
(10) Relative clause islands
   a.  Did, Karen __i meet [the guy [who wrote “Overjoyed”]]?
   b.  * [What song], did Karen meet [the guy [who had written __i]]?

Complex NPs, embedded questions, adverbial modifiers, and clausal subjects also appear to prohibit the extraction of their elements. The study of island effects led to the hypothesis that the relationship between a moved element and its canonical position appears to be subject to some sort of locality condition.

(11) Complex NP islands
   a.  Did you hear the claim that Sophie wrote “Primary Colors”?  
   b.  * [Which book] did you hear the claim that Sophie wrote __i?

(12) Adverbial modifier islands
   a.  Richard bought a new house after Jane got a raise.
   b.  * Who, did Richard buy a new house [after __i got a raise]?
   c.  Alice fell asleep when she was reading a boring book.
   d.  * What, did Alice fall asleep when she was reading __i?

(13) Wh-islands
   a.  Did, Alice __i ask who you invited to the birthday party?
   b.  *[Which party], did Alice ask [who you had invited to __i]?
   c.  *[To which party], did Alice ask [who you had invited __i]?

(14) Clausal subject islands
   a.  For me to read “Aspects of the Theory of Syntax” would upset Sophie.
   b.  * I wonder [which book], [for me to read __i] would upset Sophie.
Ross (1967) also observed that neither a conjunct nor an element within a conjunct may be extracted. He gives this as the two-part Coordination Structure Constraint (CSC):

(15)  * Extraction of a Conjunct  
   a.  *He will put the chair between [[that table] and [that sofa]]
   b.  *[Which table], will he put the chair between [__i and [that sofa]]?

(16)  * Extraction of an Element from Within a Conjunct
   a.  *Jane [[bought a card] and [wrapped a gift]].
   b.  *The card which, Jane [[bought __i] and [wrapped a gift]] is classy.
   c.  *What, might Jane [[buy __i] and [wrap a gift]]?

In addition to the exceptions that have been noted, Ross (1969) observed that wh-island violations could be tolerated when elements following the fronted wh-element were deleted. Ross called these cases *sluicing*. An example contrasting a post-deletion construction with its non-deletion counterpart is given below.

---

1 A number of exceptions to the second part of the CSC have been noted. One widespread exception is Across-the-Board (ATB) movement, where extraction out of one conjunct is possible if an identical element is simultaneously extracted from the other conjunct.

   a.  *The gift which, [[Jane bought t_i] and [Alice wrapped t_i]] is under the tree.
   b.  *What, did, [[Jane t_j buy t_i] and [Alice t_j wrap t_i]]?

The system that is the object of study in the remaining chapters can be easily modified to account for the parallelism, but there is no mechanism in the proposed system that would tie together the parallelism with subsequent disregard of an island violation. The generalization to which these are exceptions is quite robust, and the analysis to be given covers other island effects as well. That is, the cost of changing the analysis to accommodate this challenging pattern is high, so I note this pattern here, and then set it aside.
Wh-island violation vs. Sluicing

a. * I believe the claim that he bit someone, but they don’t know who, I believe the claim that he bit ___.

b. I believe the claim that he bit someone, but they don’t know who, I believe the claim that he bit ___.

2.3 TREE ADJOINING GRAMMAR BASICS

2.3.1 INTRODUCTION

TAG is a structure-generating formalism introduced by Joshi, Levy, and Takahashi (1975). In this formalism, predefined pieces of structure are combined into larger structures. These predefined pieces of structure are bounded in size, and together, they comprise a finite set called the elementary trees. The two operations used to combine structures are called substitution and adjoining. Each of these operations is illustrated in subsection 2.3.2. TAG itself is a mathematical formalism, not explicitly a theory of natural language, but it can be applied to the description of natural language syntax. As first discussed by Kroch and Joshi (1985), the application of TAG to linguistic analysis has a number of attractive qualities. First, the linguistics community has observed that a number of syntactic relationships appear to be subject to certain locality conditions. (The island effects described in subsection 2.2.3 are an example, though there are other cases as well.) TAG provides a natural vocabulary for delimiting what counts as a local domain: an elementary tree. A central claim of the TAG framework as a description of natural language syntax is that all syntactic dependencies occur within the bounds of an elementary tree. As an outcome of this commitment, a number of locality effects could be derived, rather than stipulated as they had been before. In addition, the manner in
which structures can be combined allowed iterated movement (of the kind we saw above in (8)) while maintaining the distinctness of what we (might reasonably) think of as local domains. This is the result of factoring out recursive structure so that what appears to be a non-local dependency can actually be recast as a local dependency. Furthermore, there is a possible correspondence between the formal languages that can be generated by a TAG and natural languages. It has been shown that natural languages fall outside the class of languages that can be generated by context-free grammars (Culy 1985, Shieber 1985), but the next class of languages in the Chomsky hierarchy, the languages generated by context-sensitive grammars, is clearly too unrestricted to be a formal characterization of the class of natural languages. TAG grammars are only slightly more expressive than context-free grammars, enough to capture the patterns reported by Culy (1985) and Shieber (1985) yet still restrictive enough to be a candidate for the appropriate characterization of the formal power of natural language.

One last note is in order. I have assumed the elementary trees to be the type of phrase structure familiar in Government Binding or Minimalist approaches to syntax, as these are the representations I am most familiar with. However, because the TAG formalism itself says nothing about the origins of the elementary trees, the trees need not be the types of structure I have assumed here. Other theories, such as Head-driven phrase structure grammar (Pollard and Sag 1994) and Lexical functional grammar (Kaplan and Bresnan 1982), posit that the building blocks of syntax look very different. The TAG formalism can be applied to these alternatives as well.
2.3.2 Combinatory Operations: Substitution and Adjoining

The Substitution operation takes the root of one tree and identifies it with a node along a second tree’s periphery. Typically, the nodes along a tree’s periphery are referred to as that tree’s frontier.

(18) Schematic of substitution

An example of how Substitution can be used linguistically is illustrated below in the derivation of a complex sentence in which a verb takes a clausal complement.

Substitution combines the trees in (19) to derive the structure in (20). A solid head arrow is used here to show that Tree 2 is substituted into Tree 1 at node CP.

(19) Tree 1

(20) Tree 2
In contrast to Substitution, which can only add structure at a frontier node, the Adjoining operation can add structure at any node. Adjoining involves a special class of trees called *auxiliary trees*. Auxiliary trees are recursive: they have some node along their frontier that is labeled identically to the label of its root node. This node is called the *foot node*. Given an auxiliary tree whose root and foot are labeled X, we can think of the Adjoining operation as taking a node labeled X in another tree and inserting the auxiliary tree at that node, expanding what used to be node X into a recursive piece of structure. When Adjoining combines two trees, one of which is recursive on X, the root of the recursive tree is identified with a node X in the other tree and any structure that originally appeared below node X is now attached below the foot X.
An example of how Adjoining can be used linguistically is illustrated below in the derivation of a temporally modified clause. Adjoining combines the trees in (22) to derive the structure in (23). An open head arrow is used here to show that Tree 4 is adjoined into Tree 3 at node VP. In the case of this derivation, the operation of adjoining is similar to that of Chomsky-adjunction as it serves to introduce a modifier. However, in subsequent sections, we will see that adjoining has more general applications.

(22) Tree 3

(23) Tree derived from adjoining Tree 4 into Tree 3 at VP

2.3.3 Derivation Structures in TAG

A TAG derivation involves the combination of elementary trees via adjoining or substitution. The sequence of derivational steps can be represented as a derivation structure (Vijay-Shanker 1987), a tree whose nodes represent elementary trees and whose
edges represent combination via adjoining or substitution. The edge that connects a mother elementary tree A to a daughter elementary tree B indicates that B has either substituted or adjoined into A. Each edge is labeled with the locus of combination. The combination of Tree 3 and Tree 4 as shown in (22) can be represented with the derivation structure below.

(24) Derivation structure indicating Tree 4 combines with Tree 3 at VP

Sam bought a ring

| VP

after Heidi returned to Bellingham

A slightly more complex example, the derivation of *We think that Sam must like Heidi, because Sam bought a ring after Heidi returned to Bellingham* and the corresponding derivation structure are given below.

(25)
A well-formed derivation obeys the requirement that every pair of elementary trees in a mother-daughter relationship in the derivation structure must be possible independently of the rest of the derivation structure. This reflects the way in which the combinatorial operations work. For example, we may not combine two non-recursive trees to create a recursive derived tree that subsequently adjoins into a third tree. Consider an example from Frank (2002; p77) of the type of derivation this condition prohibits. It is allowed for the \textit{it-was-certain}-tree to substitute into the \textit{seemed}-tree in (27). Since the \textit{seemed}-tree has a I’ root and the \textit{it-was-certain}-tree has a I’ frontier node, the resulting structure appears to be recursive. However, this derived structure is banned from acting as an auxiliary tree; it cannot adjoin into another tree. In the derivation structure that follows, the combination of the mother \textit{Eleanor to know the answer} with its daughter \textit{seemed} is dependent on the bottom half of the derivation structure. The recursive structure required for adjoining is only achieved if \textit{it was certain} has substituted into \textit{seemed}. 

\begin{center}
\begin{tikzpicture}
  \node (root) {because};
  \node[below left] (ip1) {IP} at (root.south west) {We think};
  \node[below right] (ip2) {IP} at (root.south east) {Sam bought a ring};
  \node[below] (cp) {CP} at (ip1.south) {that Sam must like Heidi};
  \node[below] (vp) {VP} at (ip2.south) {after Heidi returned to Bellingham};
\end{tikzpicture}
\end{center}
(27) Disallowed derivation for *Eleanor seemed it was certain to know the answer.

(28) Ill-formed derivation structure for *Eleanor seemed it was certain to know the answer.

In TAG, the derivation structure cannot be read off of the derived phrase marker. This is a point of departure from alternative frameworks. In the Minimalist Program, the derivational history is transparently read off the derived phrase marker, and derivational structure and phrase structure are essentially interchangeable. Other frameworks, such as Head-Driven Phrase Structure Grammar and Lexical Functional Grammar, approach phrase structure from a non-derivational, constraint based view, and therefore lack an analogous notion of derivation.
2.3.4 ELEMENTARY TREES

Above, we saw that the TAG operations can constrain the possible derived phrase markers. Since whether or not an operation may apply depends on the form of the two pieces of structure to be combined, other constraints on the phrase markers may also arise as a result of properties of the elementary trees. In the introduction to this section, I alluded to the TAG hypothesis that syntactic dependencies occur within the bounds of an elementary tree.

One of the dependencies that is generally taken to be elementary tree-local is the relationship between a verbal head and its thematic arguments. Thus, it is typically assumed that elementary trees include slots for any arguments of the tree’s lexical head. This is part (a) of the theta criterion for TAG, which is given in Frank (1992, 2002). Not only so, but every substitution node in an elementary tree must be assigned a theta role. This is stated in part (b).

(29) Theta-Criterion (TAG version)
   a. If H is the lexical head of elementary tree T, H assigns all of its theta-roles within T.
   b. If A is a frontier nonterminal node of elementary tree T, A must be assigned a theta-role in T.

The dependency between a moved element and its base position is another dependency taken to be within an elementary tree. Since head movement is a case where the head of a phrase moves into another phrase, it appears that elementary trees must be larger than phrases. Based on movement of a verb out of its verb phrase (VP) and into an inflectional phrase (IP) (observed in languages such as French and Early Modern English)
and movement of an auxiliary verb out of its inflectional phrase and into a higher complementizer phrase (CP) (as in the subject-auxiliary inversion we saw earlier), elementary trees should be able to be as large as CP. Movement of a \textit{wh}-object, presumably inside the VP, to the front of a question, presumably into the CP, further supports this idea. Grimshaw (1991) argues that a VP, the projection\(^2\) of a lexical verb, and its IP, the projection of a functional head associated with the verb, behaves as a syntactic unit. Likewise, when a CP is present, a VP, IP, and CP behave as a syntactic unit as well. Grimshaw (1991) suggests that an IP and a CP are both extended projections of V and that DP and PP are both extended projections of N. It turns out that a number of grammatical processes in addition to the two examples here are localized within extended projections. This leads Frank (1992, 2002) to contend that an elementary tree can be as large as an extended projection. Below, we give Frank’s Condition on Elementary Tree Minimality.

\begin{enumerate}
\item \textbf{Condition on Elementary Tree Minimality (CETM)}
\begin{quote}
The syntactic heads in an elementary tree and their projections must form an extended projection of a single lexical head (in the sense of Grimshaw (1991)).
\end{quote}
\end{enumerate}

Taking the TAG Theta-Criterion and the CETM together, we can characterize elementary trees in TAG as extended projections that include argument slots for any arguments of the tree’s lexical head.

\footnote{\textsuperscript{2} A projection is a constituent which is an expansion of a head word. e.g. VP is an expansion of V.}
2.3.5 Movement in TAG

In TAG, there are two approaches to movement. Movement in which an element moves to a position that differs in type from its original position (e.g. *wh*-movement) is done by movement in the traditional transformational sense with traces, but within the bounds of an elementary tree, as shown in (31).

(31) Traditional treatment of *wh*-movement and head movement

![Tree Diagram](image)

In contrast, movement phenomena in which an element moves from its original position to a position of the same type (e.g. raising, cyclic *wh*-movement) does not actually involve any moving of elements. This type of movement is accomplished via the adjoining operation. Below, in (32), auxiliary Tree 6 adjoins into Tree 5 to yield the embedded clause in *Tyler knows which books Bekah had thought Jack might like*.

This latter approach to apparent movement is similar in spirit to the multi-domination approach to movement. Here, a single element manages to play more than one role in the course of the derivation. In the extension of TAG explored in this thesis, this type of movement-via-stretching remains unchanged. The former approach, however,
is replaced in the extension so that movement from a base position to a position of another type is also only apparent.

(32) Adjoining treatment of cyclic wh-movement

a. Tree 5

b. Tree 6

c. Derived tree
These few pages on TAG cannot adequately convey all the advantages that the framework has/might have/does not have. It is my hope, however, that the reader is left with a basic understanding of the mechanics and an appreciation for the TAG approach to syntax as a launching point for the system considered in this thesis. This section has ended with a demonstration of the well developed TAG approach to movement and a promissory note that multi-dominance will replace elementary tree internal movement. However, as the previous section indicates, coordination phenomena play a central role in this thesis, but I have not said anything yet about coordination in TAG. This is the topic we turn to in the next chapter.
CHAPTER 3
MULTI-DOMINANCE TREE ADJOINING GRAMMAR

3.1 INTRODUCTION
This chapter addresses the question of how multi-dominance structures are generated. In
the original Tree Adjoining Grammar (TAG) formalism (Joshi, Levy, and Takahashi
1975), which we outlined in section 2.3, two operations, adjoining and substitution, apply
to elementary trees to derive a larger structure. To allow for coordination in TAG, Sarkar
and Joshi (1996) introduce a third operation, node-contraction, which takes two nodes of
like categories and collapses them into a single node. The proposal pursued here is that,
instead of being restricted to coordination, node-contraction is a general mechanism in
the TAG system; it is node-contraction that introduces multi-dominance into the system.
Below, we first recap Sarkar and Joshi’s original proposal, then see how extending their
proposal generates cases traditionally dealt with via displacement/movement, including
cases that are difficult for traditional TAG. Finally, we consider how the addition of
node contraction to the formalism bears on our conception of the size of elementary trees.

3.2 GENERATION OF COORDINATE STRUCTURES: SARKAR AND JOSHI (1996)

3.2.1 THE PROBLEM FOR CLASSIC TAG
Because the relationship between a verb and its arguments is taken to be an elementary
tree internal dependency, coordination presents a problem. Consider the example below:
(1)  *Joe eats cookies and drinks tea.*

*Joe* is an argument of both *eats* and *drinks*. If we maintain the Theta-Criterion for TAG (given in Chapter 2 (29)), then each verb must assign all of its theta roles within the elementary tree that it heads. This means that *Joe* must be an argument of both the elementary tree headed by *eats* and the elementary tree headed by *drinks*. Consequently, coordination requires either (i) some means for two elementary trees to share arguments or (ii) a single elementary tree with two verbs. The latter, however, has the disadvantage of violating part of the CETM. Recall that to rein in the size of elementary trees (and, as a consequence, the notion of local domain), I have committed to capping the size of an elementary tree to an extended projection of a single lexical head. We cannot posit an elementary tree with two verbs without abandoning the notion that a tree contains a single lexical head.

3.2.2 THE NODE CONTRACTION OPERATION

To allow for shared arguments, Sarkar and Joshi (1996) introduce a novel operation that augments TAG’s means of combination: node-contraction. Node-contraction takes two nodes of like categories that are both marked for contraction and collapses them into a single node. The edge between the single node and each of its mother nodes is maintained, yielding a multi-dominated node. Thus, it is node contraction that introduces multi-dominance into the augmented TAG system.
To see how node contraction works, let us consider the example of VP coordination. First, each lexical item anchors an elementary tree, as shown in (2).\(^1\) The two subject NP nodes are marked for contraction, noted here with a circle.

(2) Elementary trees involved in *Joe eats cookies and drinks tea*.

Note that I have adopted an asymmetric structure for coordinated structures as argued for by Munn (1993).\(^2\) *And* is taken to head a Boolean phrase, to take two XP arguments, and to anchor the elementary tree it belongs to.\(^3\) Let us label the root of the Boolean elementary tree as BP:XP, where X is the category of the two arguments taken. We will assume some transfer of properties allows this and that the combinatory operations of TAG do not distinguish between XP and BP:XP.

The VP coordination *and* tree, second tree from the left in (2), takes two VPs as arguments. The *eats* tree and *drinks* tree both substitute into the *and* tree to yield (3):

---

\(^1\) Sarkar and Joshi (1996) propose that elementary trees do not have nodes marked for contraction and that another operation, *build contraction*, creates trees with contraction nodes by freely marking some nodes for contraction. Whether elementary trees include contraction nodes or not will not change our discussion, so for ease of explication, let us assume specification as a contraction node is information included in elementary trees.

\(^2\) Munn (1993) actually contends that the boolean phrase is adjoined to the first conjunct. However, the structure adopted here is in line with Munn’s primary claim: that an asymmetrical structure for coordinate structures is more accurate than assuming a flat structure.

\(^3\) This is a slight departure from Sarkar and Joshi (1996), who propose a coordination schema that would result in some structures with no root. Adopting an asymmetric structure allows us to avoid modifying the assumption that each syntactic structure must have a root, and keep multidominance as the only modification under exploration. In chapter 5, we will also find that adopting Munn’s structure has positive consequences in our linearization process, which relies on the notion of sister precedence.
(3) Trees after some composition but before node contraction

The identification of the two NP nodes yields (4). The resulting node in subject position is now immediately dominated by two nodes, VP1 and VP2, rather than just one.

(4) Trees after node contraction

Finally, the NP’s substitute in, yielding (5).
A few final notes on the node contraction operation are in order. First, though not explicitly stated by Sarkar and Joshi, I assume that nodes marked for contraction must contract. A well-formed final structure may not have any nodes that remain marked for contraction. Second, I leave open the possibility that a node marked for contraction may be identified with more than one node. That is, even though a node that has contracted has no need to contract again, perhaps the system ought to allow the node to continue contracting with other nodes. Under this view, the nodes marked for contraction maintain a status through the derivation as a “contracting node.” This has the positive outcome of allowing iterated conjunction. Consider, for example, the sentence Joe eats cookies and drinks tea.

---

This contrasts with an approach where a node that has satisfied its need to contract is no longer available for contraction. Under such an approach, node contraction would be very similar to feature checking, where the need to contract is a feature that must be checked. Once the feature is checked, further movement is not possible.

This also contrasts with a second approach to iterated conjunction in which a node may be double/triple/quadruple/etc marked for contraction. For example, we would need an eats-tree where the subject DP was marked for a single contraction to derive Joe eats cookies and drinks tea, and we would need an eats-tree where the subject DP was marked for double contraction to derive Joe eats cookies and drinks tea and studies bridges. If a whether or not a node is marked for contraction is considered part of the information carried by an elementary tree (e.g. a system without the build contraction operation described in footnote 1), then because iterated conjunction is unbounded, this multiple-contraction-marking approach would call for an infinite number of elementary trees.
cookies and drinks tea and studies bridges.\(^6\) If the same subject is shared by eats, drinks, and studies, then the final structure should look like the one below:

(6) Proposed structure for Joe eats cookies and drinks tea and studies bridges.

To derive the structure in (6), it appears that even after contraction of two of the NP nodes marked for contraction, the resulting multiply dominated node must be available for subsequent contraction with the remaining NP node marked for contraction.

(7) Trees after a single node contraction

\(^6\) Though prescriptive grammars for English dictate that only the and between the final two conjuncts is pronounced, a pronounced and between each of the conjuncts is not uncommon in spoken English. To eliminate the non-final ands, we could posit that additional conjuncts may be adjoined to BP (the line taken by Munn (1993)) or that a silent and exists.
If the multiply dominated node has lost the ability to contract with another node, no node will be available to contract with the remaining NP node marked for contraction.

While Sarkar and Joshi (1996) introduce node contraction specifically to accommodate coordination, this work considers node-contraction to be a general mechanism in the TAG system. Let us call this system Multi-dominance-TAG (MD-TAG). We will also use the term *syntactic graph* interchangeably with the term multi-dominance structure. I will continue to assume that elementary trees are indeed trees, and not graphs. Within this dissertation, this decision is not explored as a serious claim about the nature of natural language syntax. The decision about the nature of elementary trees is more for the sake of isolating for study the effect of making one small change to the original TAG system, i.e. expanding the set of operations that combine nodes in elementary trees. This seems to be the right place to make a modification, because the TAG formalism itself is about how structures can be combined; TAG is silent on the issue of the origins of the elementary structures. Because the mechanism responsible for constructing elementary trees is outside of the system under consideration, I prefer to leave their status as trees in classic TAG untouched in MD-TAG. This also has the consequence of attributing the source of multi-dominance wholly to a structure combining operation. This does have the effect of allowing multi-dominance only in derived structures, but note that it does not rule out contraction between nodes within a single elementary tree. Node contraction may apply to two nodes within a single elementary tree after that tree has combined with a second elementary tree. Thus, in the

---

An alternative view is that node contraction augments the traditional operations rather than having status as a third operation distinct from adjoining and substitution. I.e. adjoining in MD-TAG would be one operation that has the effect of traditional adjoining followed by node contraction, and substitution in MD-TAG would likewise have the effect of traditional substitution followed by node contraction.
proposed system, the answer to how multidominance enters is the same as that given by Sarkar and Joshi: The node-contraction operation applies to syntactic structures and generates a new syntactic structure in which a node may have multiple parents.

3.2.3 CONDITIONS ON NODE CONTRACTION

Having augmented the set of operations, a new question arises: Under what conditions is the new operation allowed to apply? Sarkar and Joshi (1996) identify two conditions on node-contraction:

1. For two nodes to contract, they must both be marked for contraction, and
2. For two nodes to contract, they must be of like category.

Here, I set forth a few additional conditions. First, node contraction does not apply to lexical items (8). (Or, alternatively, lexical items cannot be marked for contraction.) This has the effect of saying that a lexical item’s preterminal node is always that lexical item’s sole parent.

(8) Non-lexical Condition on Node Contraction:
Lexical items cannot be contracted. The node contraction operation contracts only nodes that are not lexical items.

i.e. if (X is a lexical item),
then there do not exist nodes Y and Z such that
((Y is a parent of X) and
(Z is a parent of X) and
(Y \neq Z))
Second, the node contraction operation cannot contract two nodes when each of those nodes carries phonological content (9). What “having phonological content” means is defined formally in Chapter 5, (18). For now, it is sufficient to think of the condition as requiring at least one of the two nodes that are contracted to be a substitution node or a foot node of an auxiliary tree since a substitution node does not dominate any lexical items prior to substitution and the foot of an auxiliary tree does not dominate any lexical items prior to adjoining. One effect of this condition in conjunction with the non-lexical condition is the prohibition against preterminals that dominate more than one lexical item. Together, these conditions ensure a one-to-one relation between lexical items and each item’s preterminal parent.

(9)  

Phonological Content Condition on Node Contraction:

The node contraction operation contracts only pairs of nodes (X, Y) where at most one member of the pair has phonological content.

i.e. if ((X has phonological content) and (Y has phonological content), and (X ≠ Y))

then there does not exist a node Z such that ((Z is a parent of X) and (Z is a parent of Y))

Third, the node contraction operation cannot contract nodes in a dominance relation. This is a condition that is shared with the traditional c-command condition on movement (if a constituent must move to a c-commanding position, then the constituent must not move to a dominating position), though it does not require the nodes to be in a c-
command relation. This has the desirable effect of keeping dominance asymmetric and of keeping the resulting graphs acyclic.

(10) Non-dominance Condition on Node Contraction:
The node contraction operation does not contract pairs of nodes \((X, Y)\) where \(X\) and \(Y\) are in a dominance relation.

\[
i.e. \quad \text{if } ((X \text{ dominates } Y) \text{ or } (Y \text{ dominates } X)),
\]
\[
\text{then there does not exist a node } Z \text{ such that }
\]
\[
((Z \text{ is a parent of } X) \text{ and }
\]
\[
(Z \text{ is a parent of } Y))
\]

3.2.4 DERIVATION STRUCTURES

In subsection 2.3.3, derivation structures were introduced as representation of sequences of derivational steps. Since the MD-TAG system has been augmented with a new operation, let us add to Vijay-Shanker’s (1987) conception of derivation structures a new kind of edge corresponding to node contraction, marked with a dotted line. We will also typically use all the frontier nodes of an elementary tree as its label in a derivation structure. That is, instead of labeling the node representing the conjunction tree as “and,” we will label it “VP and VP.” This also allows us to denote nodes marked for contraction, indicated with a circle. Trees that substitute into a contracted node will be shown as substituting into each of the trees that host one of the contraction nodes. The derivation structure for the combination described above in (2) through (5) is given below.
3.3 GENERATION OF QUESTIONS

In the introduction to classic TAG in chapter 2, two approaches to movement were described, one for A-movement and one for A'-movement. Recall that A-movement is accomplished via “stretching,” as a result of adjoining and does not actually involve moving an element. MD-TAG inherits this elegant account of A-movement from traditional TAG. A'-movement in classic TAG is accomplished via a mainstream notion of movement, but within an elementary tree. When A'-movement is outside of a single clause, elementary tree internal movement is coupled with the same “stretching” used in A-movement. The proposal pursued here is that no element is actually moved in A'-movement either. Rather, node contraction replaces elementary tree internal movement. That is, there is only a single copy of the “moved” element and it is immediately dominated by multiple nodes.

In English, wh-questions share two characteristics: a fronted wh-element and subject-auxiliary inversion. Indeed, it has been proposed by Ginzburg and Sag (2000) that these are the key grammatical properties which give rise to the interrogative force typically associated with questions. That is, interrogative force arises as the result of
either a *wh*-operator or subject-auxiliary inversion. Because subject-auxiliary inversion can occur independently from *wh*-fronting and *wh*-fronting can occur independently from subject-auxiliary inversion, let us suppose each of these components corresponds to a particular elementary tree.

(12) a. *Wh*-question tree 
    b. subj-aux inversion tree

The *wh*-question tree for English in (12a) is a non-recursive structure with a DP non-terminal node that is marked for contraction. It is this piece of structure that houses the fronted *wh*-element. This tree is involved in generating matrix and embedded *wh*-questions. In contrast to (12a), the structure in (12b) is recursive. We will see in the next chapter that positing the former tree to be non-recursive and the latter tree to be recursive will assist us in stating additional constraints on node contraction that yield island effects. The tree in (12b) includes an I non-terminal node marked for contraction, which accomplishes auxiliary inversion. This tree is involved in matrix *wh*-questions and *yes/no*-questions. Though it is often thought that I moves into a C position, I have labeled the head as an I since the element that moves into the head is always an auxiliary and never a canonical complementizer such as *that* or *for* in English. As I and C have both been widely assumed to belong to a V’s extended projection, both are verb-like categories in some sense and perhaps the distinction between the two is blurred in question formation.
The subj-aux inversion tree in (12b) can be adjoined into the did-eat tree in the usual TAG manner. A CP tree with an I contraction node and a DP contraction node is substituted into the CP node of (12a), also in the usual TAG manner. Following substitution and adjoining, the DP nodes marked for contraction contract and the T nodes marked for contraction contract. Each contraction in this example is between two elementary trees that are “adjacent” in the derivation. (Later, in chapter 4, the notion of two trees being “close enough” in the derivation to allow contraction of nodes across trees is developed more fully.)

(13) Did-eat IP tree substitutes into the wh-question tree
    subj-aux inversion tree adjoins into the did-eat IP tree

Finally, by substituting in the appropriate lexical items into the DP nodes, we are left with a wh-question, as in (14).
3.4 GENERATION OF INTERLEAVED CONSTRUCTIONS

Above, we saw that a traditional TAG system has two types of dislocation: elementary tree internal movement (in the traditional transformation sense) and interposing a tree between two elements of another tree via adjoining. Constructions such as the one in (15) involve both types of dislocation: the movement of does has been analyzed as elementary tree internal movement, and the movement of the subject has been analyzed as adjoining.

(15)  *Does Sam t₁ seem t₁ to like pizza?*

These constructions have been difficult for traditional TAG, because they require that adjoining and elementary tree internal movement interact in a manner that allows interleaving of elements of elementary trees. To fully appreciate the puzzle, let us
backtrack and first consider the TAG analyses of (16) and (17) in which each type of movement appears separately.

(16)  \textit{Does}_i \textit{Sam}_t \textit{like} \textit{pizza}?

(17)  \textit{Sam}_j \textit{does seem} \textit{t}_j \textit{to like} \textit{pizza}.

Because of the assumption that movement is elementary tree local, the final position and the original position of \textit{does} must belong to the same elementary tree. These positions are standardly taken to be the C head of CP and the I head of IP. Since the CETM requires that the tree anchored by \textit{like} be the extended projection of \textit{like}, both of these positions may be elementary tree internal. The classic TAG analysis for the sentence in (16) is given in (18). (The derivations in (18) and (19) are slightly condensed; they do not show the substitution of the DP elementary trees as a separate step.)

(18)  Traditional TAG analysis of \textit{Does}_i \textit{Sam}_t \textit{like} \textit{pizza}?

The sentence in (17) involves the matrix verb \textit{seem}, which falls into the class of verbs called \textit{raising verbs}. Like the class of verbs called \textit{control verbs}, raising verbs take
an infinitival sentential complement. Unlike control verbs, however, raising verbs do not appear to impose their own restrictions on their subjects. Instead, the range of possible subjects is determined by the embedded predicate. Consider the contrasts between the sentences in (19) involving *seem* and those in (20) involving the control verb *hope*. For example, when the embedded predicate allows an expletive *there, a weather it, or an idiom chunk as a subject, matrix *seems* also permits the same subject. Verbs like *hope*, however, do not take such subjects.

(19) a. There seems to be a problem.
    b. It seems to be hailing outside.
    c. Curiosity seems to have killed the cat.

(20) a. * There hopes to be a problem.
    b. * It hopes to be hailing outside.
    c. * Curiosity hopes to have killed the cat.

Syntacticians concluded that the difference between the two classes of verbs is a reflection of a difference in whether or not the verb assigns a thematic role to its subject. Control verbs assign their subjects a theta role, while raising verbs do not. The subject of a raising verb receives its theta role from the embedded verb. In chapter 2, we made a commitment to a theta-theory for TAG which requires an argument receiving a theta role to be in the same elementary tree as the verb that assigns the theta-role. However, the requirement is a TAG version of an older idea from transformational grammar that noun phrases originate in the position where they receive a theta role. The transformational account of raising, then, is that the subject of a raising verb begins as the subject of the
embedded infinitival clause and, later in the derivation, moves into the subject position of the raising verb.

The TAG formalism does not allow the dependencies that result from movement to take place across clauses. Instead, the TAG derivation of raising proposed by Kroch and Joshi (1985) and Frank (1992) involves no actual movement, only apparent movement. In (17), for example, there is only a single instance of the subject Sam. As theta-role assignment comes from like and theta-role assignment is taken to be elementary tree local, the substitution site of Sam is posited to be in the elementary tree anchored by like. This is the tree shown on the left in (21). Because an elementary tree that contained seem and its auxiliary does in addition to like and to would violate the CETM, seem and does must appear in a separate tree, the tree on the right in (21). Note that because seem does not assign a theta-role to its subject, there is no subject position in this tree. The insertion of the does seem tree stretches the distance between the subject and the verb that selects it (like) in the derived tree. This account straightforwardly captures both why the range of subjects is determined by the embedded verb in sentences with matrix raising verbs and why the subject surfaces as the subject of the matrix raising verb.

(21) Derivation of Sam does seem to like pizza.

![Diagram of sentence structure]
Let us return now to example (15) *Does Sam seem to like pizza?* The CETM requires that *does* appear in an elementary tree along with a verb, so we assume that *does* and *seem* belong to the same elementary tree. Further, the assumption that movement is elementary tree-local requires that *does* and the position that *does* extracts out of ought to belong to the same elementary tree. The *does seem* tree is presumably adjoined into the *Sam to like pizza* tree. Just above, in our example of raising, the tree headed by the raising verb adjoins between *Sam* and *to*. The puzzle in the interrogative is how the final position of *does* comes to be higher than *Sam* while the final position of *seem* (and presumably the trace of *does*) is simultaneously lower than *Sam*. The puzzle is not an isolated case. As Kulick (2000) discusses, constructions such as clitic climbing, long passives, and long scrambling case from Romance and Germanic languages also appear to involve interleaving of elements from two elementary trees. To address these types of constructions, a number of extensions to TAG have been proposed (e.g. Rambow 1994, Bleam 2000, Kulick 2000).

Augmenting the traditional combinatorial operations in TAG with node contraction straightforwardly allows the derivation of interleaved constructions. The derivation of (15) is given below.

(22) Trees involved in the derivation of *Does Sam seem to like pizza?*
The *does seem* tree adjoins into the *to like* tree in the usual fashion.

(23) Trees after *does-seem* tree adjoins into the *to-like* tree

When the CP substitutes into the subj-aux inversion tree, the I nodes contract.

(24) Trees after the *to-like* tree substitutes into the subj-aux inversion tree
Once the DP’s substitute in, the derivation yields the structure below:

(25) Final structure for *Does Sam seem to like pizza?*

![Diagram](image)

Of course, a modification that allows a previously undervivable construction leads to a decrease in restrictiveness. The goal, however, is to limit the increase in power to avoid overgenerating. A major concern associated with adopting some of the previous suggestions for dealing with interleaved cases is the sharp increase in permissiveness. One of the attractive aspects of the node-contraction approach is that the increase in power does appear to be relatively small. In the final chapter, we see that MD-TAG can generate languages with six counting-dependencies, but perhaps no more. (Languages with five counting-dependencies are already outside the class of classic TAG languages.) However, in the next chapter, we see that the derivation of island effects, an attractive outcome of classic TAG, can be retained in MD-TAG.
3.5 ALLOWING PHRASE SIZED TREES

In our elementary trees thus far, we have maintained the Theta-Criterion and the CETM. Hegarty (1993), however, suggests that elementary trees may be even smaller, that any syntactic phrase may be an elementary tree. He argues that this “factors out” recursive structure that was present in the elementary trees obeying CETM. Certainly, if we assume that our structures conform to X'-theory, a verb’s most extended projection includes at least three instances of recursive structure. In Hegarty’s system, the verb-like categories V, I, and some Cs are crucially considered the same categories by the combinatory operations. They are instead distinguished by their feature content.8 This allows phrase-sized trees to be combined using the TAG machinery, as depicted in figure (26). Thus, the large structures that we have been assuming as elementary trees can actually be derived from simpler structures rather than stipulated. Note, however, that under Hegarty’s view, some trees do not obey the CETM; as with the did-tree in (26), it need not be the case that a lexical (vs. functional) predicate anchors each elementary tree.

(26) Deriving extended projections with Hegarty (1993) style trees

---

8 Hegarty points out that Grimshaw’s (1991) suggestion that I is categorically nondistinct from V is exploited in different ways by Frank’s conception of elementary trees and Hegarty’s conception of elementary trees.
Under Hegarty’s approach, what was once allowed as elementary tree internal movement from phrase to phrase now appears to be movement between elementary trees. For example, the classic TAG framework easily accommodated the view that all subjects originate in the specifier position within VP and are subsequently raised to the specifier position of IP.

(27) Elementary tree internal subject raising

In (26), the derivation treats subject raising from spec-VP to spec-IP just like subject raising involving raising verbs. Hegarty takes care to point out, however, that if we wished to make a commitment to two distinct specifier positions, one associated with the lexical verb and the other with the auxiliary, his system can accommodate this by positing that TAG adjoining may combine sets of auxiliary trees rather than singleton trees. The set which introduces an element and its trace includes one tree that includes the trace and another tree consisting of a single node⁹ that substitutes into the final position. This single-node-auxiliary tree and the “base position” trace are allowed to be co-indexed, as they comprise a single set and enter the derivation together, and both members of the set must adjoin into the same initial tree. The dotted line encodes the

⁹ That is, an auxiliary tree where the root happens to be the identical to the foot node.
requirement that the single-node-auxiliary tree must c-command the V'-auxiliary tree upon adjunction of the tree set into another tree (set).

(28) Subject raising in Hegarty (1993)

\[ \text{The use of sets of trees as the input to the combinatory operations will not, however, be able to accommodate what was previously elementary tree internal head-to-head movement. We take V to I movement in French and Early Modern English as our examples. In the presence of an auxiliary, a French adverb like } \textit{souvent} \text{ optionally appears between the auxiliary and the verb or after the auxiliary and verb. (The difference in placement yields a difference in adverbial scope.) Such an adverb does not appear preceding the auxiliary.} \]

   Pierre often has eaten my breakfast.

b. Pierre a souvent mangé mon petit déjeuner.
   Pierre has often eaten my breakfast.

c. Pierre a mangé souvent mon petit déjeuner.
   Pierre has eaten often my breakfast.

‘Pierre often has eaten my breakfast.’
In the absence of an auxiliary, however, the order Adv V is prohibited. This suggests that verbs in French can and sometimes must move to a higher position. (In this work, we have not expanded the IP to show all the functional projections argued for by Pollock 1989. For our purposes, it is sufficient to use IP as a shorthand for all these functional projections and to show the verb moving into I, even though example (29c) suggests that there exist distinct head positions for a and for the final position of mangé.)

    Pierre often eats my breakfast.
b. Pierre mange souvent mon petit déjeuner.
    Pierre eats often my breakfast.
    ‘Pierre often eats my breakfast.’

In Early Modern English, not appears between an auxiliary and the verb. Not can be plausibly analyzed as a VP-adverb, an adverb occupying some position internally within VP.

(31) a. Thou shalt not die for lack of a dinner (Orlando, As You Like It, II.vi)
b. I will not hear thy vain excuse (Duke, Two Gentlemen of Verona, III.i)

In the absence of an auxiliary, however, the verb appears before not.

(32) a. I care not for her (Thurio, Two Gentlemen of Verona, V.iv)
b. I know not where to hide my head. (Trinculo, The Tempest, II.ii)
Unlike the case of subject raising, verb raising involves positions that are posited to anchor elementary trees. To account for the verb raising, Hegarty posits that in elementary trees headed by tensed French verbs (verbs in a form that will not take an auxiliary), there exists a V'-hiccup. Elementary tree internal movement takes place between the two V heads. The difference between contemporary Standard English, which does not exhibit V to I raising, and French (and Early Modern English) is attributed to the absence of such a hiccup in English elementary trees.

(33) Hegarty’s V to I raising tree and VP-adverb tree

The incorporation of node-contraction, however, allows us to adopt small, phrase sized trees while keeping the assumption that the TAG operations combine singleton trees and avoiding having to posit a V'-hiccup. The difference between English and French is attributed instead to a ban on V heads being marked for contraction in English. We see below how Hegarty’s proposal that trees are phrase-sized and node contraction interact to allow spec-VP to spec-IP raising and head movement.
This conception of functional projections as auxiliary trees fits well with approaches to language acquisition that posit that acquisition is partially a structure acquisition process (e.g. Vainikka 1993/4, Clahsen et al. 1994, Legendre et al. 2002). These approaches to first language acquisition take the observation that functional elements are acquired after lexical elements to indicate that the structure housing functional elements is acquired after lexical projections. That is, the child begins with less structure than adults rather than beginning with some Master Tree that includes all the functional projections associated with a lexical projection. Suppose that acquisition is an elementary tree acquiring process. If elementary trees must be headed by a lexical
item\textsuperscript{10}, then structure and lexical entry would necessarily be acquired together. If elementary trees may be the size of extended projections, then a single elementary tree may, for example, include both an auxiliary and a verb. It is not necessarily the case that auxiliaries and verbs should be acquired separately. On the other hand, if auxiliaries and verbs must head separate trees, then it is unexpected that auxiliaries and verbs would be acquired simultaneously.

I leave open the question of whether we should assume that elementary trees are phrasal in size. Head movement in extended-projection-sized elementary trees could be handled by allowing nodes within an elementary tree to contract as well. The main idea here is to show that allowing node contraction also allows us to posit smaller elementary trees. In subsequent discussion, I will continue to use traditional, extended projection sized trees. For discussion primarily referring to the strings of terminals generated or to the phrase structure level, i.e. our formal language and linearization sections, the reader who wishes to consider this shorthand for a derivation adjoining several phrase sized trees may do so without concern that results hinge crucially on adopting one or the other conception of elementary tree size. A change in the derivational structure leaves results referring to the phrase structure level unchanged. For proposals referring to the derivational level, it is more difficult to assess whether or not a result will hold. The results presented here do hold, and we will argue this to be so in the relevant sections.

\textsuperscript{10} In this context, I use “lexical item” to mean what we typically think of as a single word or entry in the lexicon. Here, I am not using “lexical” to contrast with “functional.”
3.6 CONCLUSION

This chapter has proposed Multi-dominance TAG (MD-TAG), a system which includes the node contraction operation in addition to the adjoining and substitution operations of classic TAG. In MD-TAG, all elements are assumed to be generated in their canonical position of interpretation, but may be marked for contraction. To accommodate the new operation, Vijay-Shanker’s (1987) representation of the derivation steps as a derivation structure can be enriched with a new kind of edge corresponding to contraction. The system not only inherits the account of generation of coordinate structures given in Sarkar and Joshi (1996), but provides accounts for cases traditionally dealt with via movement as well. Questions are generated by combining elementary trees anchored by verbs with elementary trees which correspond to a particular grammatical property of questions. Cases that involve both adjoining and “movement” that posed difficulties for classic TAG, receive an account as well. This is possible because there is no longer a base position and surface position of a moved element that must be localized to the same elementary tree. Because of this, the argument that V to I and I to C movement implies extended-projection-sized elementary trees loses some of its persuasiveness, and we revisited Hegarty’s (1993) proposal that elementary trees are phrase sized.
4.1 INTRODUCTION

In the previous chapter, the notion of syntactic displacement was replaced with the notion of multi-dominated nodes. The existence of island effects, such as those illustrated in chapter 2, show that syntactic displacement/multi-dominance is subject to certain restrictions. If we take these effects to arise as the result of the grammar, then we are obliged to account for these restrictions. Often, the proposed restrictions involve some notion of locality. The Subjacency theory, for example, stipulates that certain nodes are language-particular bounding nodes and that elements must not be displaced across more than one bounding node. In the classic TAG framework, stipulated constraints are replaced by more principled restrictions on structural combination and possible elementary trees (Kroch 1989, Frank 2002). Though classic TAG yields a principled account of island effects, this approach suffers from being overly restrictive (Kulick 2000). We have seen an example of this already in chapter 3, section 3.4: in addition to ruling out extraction from islands, the classic TAG framework also rules out interleaved constructions. The proposal I will pursue here is that islandhood is the result of requiring locality at the derivational level. Section 4.2 states this locality condition and provides examples of allowed and disallowed configurations. We see that the interleaved construction is allowed. The middle sections, 4.3 to 4.8, are devoted to showing how the proposed locality restriction works to derive a number of island effects in English:
relative clause islands, complex NP islands, adverbial modifier islands, the Coordinate Structure Constraint, wh-islands, and subject islands.

4.2 DERIVATIONAL LOCALITY

Because this chapter refers to locality at the derivational level, the use of derivation structures will facilitate clear explication. Recall that, as discussed above in chapter 2 (subsection 2.3.3), derivation structures in traditional TAG are trees whose nodes represent elementary trees and whose edges represent combination via adjoining or substitution, with edge labels indicating the locus of combination. In chapter 3 (subsection 3.2.4), dotted edges that correspond to node contraction were added and the derivation structures were no longer trees. In addition, a tree substituting/adjoining into a multiply dominated node was connected with an edge to both elementary trees to which the multiply dominated node belongs.

The proposed locality condition on contractions is given below:

(1) Locality Condition on Node Contraction (in English):
If a node in elementary tree A contracts with a node in elementary tree B in a derivation C, then one of the following relationships holds in the derivation structure for C
a. A is the mother of B,
b. A is the daughter of B, or
c. A is the sister of B.

Extending the mother-daughter metaphor used to describe structural relationship, elementary trees may only share nodes with immediate family members in the derivation
structure. The two permitted configurations are depicted schematically in (2). Each schematic should be taken to be part of a larger derivation structure.

(2) Schematics of permitted configurations

a. Contraction between mother and daughter

\[
\begin{align*}
&\ldots \\
&\text{XP} \ Y \ ZP \\
&\text{(Tree A)} \\
&\ldots \\
&\text{XP} \ Z \ WP \\
&\text{(Tree B)} \\
&\ldots
\end{align*}
\]

b. Contraction between sisters

\[
\begin{align*}
&\ldots \\
&\text{VP} \ X \ WP \\
&\text{(Tree B)} \\
&\ldots \\
&\text{VP} \ Z \ TP \\
&\text{(Tree C)} \\
&\ldots \\
&\ldots
\end{align*}
\]

Examples of banned configurations are depicted in (3). In the first case, tree C has either substituted or adjoined into tree B, and during the course of the derivation, tree B subsequently combines with tree A. The locality condition disallows contraction between nodes in tree A and nodes in tree C as tree A and C are in neither a mother-daughter relationship nor a sister-relationship. Nodes in tree A may only contract with nodes in tree B or nodes in its parent tree (not shown here). In the second case, tree D has combined with tree B, which is a sister of tree C. Nodes in tree C may contract with nodes in tree A or nodes in tree B. Nodes in tree D may contract with nodes in tree B or nodes in its daughter trees (not shown here). However, contraction between nodes in tree C and nodes in tree D is not permitted.
Recall that the elementary trees responsible for question formation in chapter 3 were pieces of the CP, pieces that were not separate elementary trees in traditional TAG. To accomplish *wh*-movement, for example, an elementary tree headed by a verb substitutes into a *wh*-question tree to form a derived phrase structure. This is illustrated in (3), a repetition of (11) and (12) in chapter 3. In traditional TAG, this derived structure would have been taken to be an elementary structure, not a derived one. The multiply linked elements involved in question formation would be linked within an elementary tree, satisfying the traditional TAG requirement that movement be elementary tree internal. Here, it is parts (a) and (b) of the condition in (1) that give rise to a similar range of possible multiple linking. This part of the locality condition allows a *wh*-
argument in the tree headed by a verb to be multiply-dominated by a node in the \textit{wh}-question tree.

(4) Derivation and derived phrase structure for \textit{What did Emmy eat?}

(5) Derivation-local node-contraction in a \textit{wh}-question: contraction between mother and daughter

(1a) and (1b) will not, however, allow the contraction of nodes belonging to a tree that substitutes or adjoins into the \textit{did-eat} tree. We shall see in the sections below that what (1a) and (1b) allow as a local domain for contraction is not very different from an extended projection.
This claim may raise the question of how long-distance dependencies, such as the dependency between the object position of *ate* and the fronted position of *what* in *What did Theo think Susie said Emmy ate?*, can be permitted. The traditional non-TAG analysis is that cyclic *wh*-movement from one CP to another has taken place. Recall, however, that in the classic TAG account of cyclic *wh*-movement, there is only a single movement, the elementary tree internal A'-movement that fronts the *wh*-element of the most embedded clause. The “cyclic” part of the displacement is accomplished by inserting structure between the *wh*-element and the rest of the most embedded clause. (See subsection 2.3.5.) MD-TAG accomplishes the fronting of the *wh*-element via node contraction and inherits the stretching account of the “cyclic” movement from classic TAG. The fronting involves contraction between mother and daughter trees in the derivation structure and the stretching involves no contraction at all.

(6) Derivation of *What did Theo think Susie said Emmy ate?*,

[Diagram of derivation of the sentence]
Part (c) of the condition in (1) is similar to a locality condition used in multi-component TAG (MC-TAG) (Weir, 1988). In MC-TAG, the combinatory operations operate over sets of trees rather than trees themselves. (Derivation steps that appear to combine two trees are taken to actually combine two sets which each contain only a single tree.) Instead of restricting multiple linking to occur within the domain of a single elementary tree, the multiple linking must occur within one of the sets of elementary trees. In tree-local multi-component TAG, all members of a set of elementary trees must combine into the same elementary tree.¹ Recast as a configuration in a derivation structure, this amounts to requiring that all members of a set of elementary trees are sisters. In MD-TAG, the operations combine singleton trees, but there is a relationship between (1c) and the tree-local requirement of MC-TAG. Imagine that each member of a

¹ In set-local multi-component TAG, all members of a derived set of trees must combine with trees belonging to the same elementary set.
MC-TAG set of elementary trees was instead a MD-TAG singleton tree with contraction nodes instead of co-indexed nodes. If the requirement that all members combine into the same tree is maintained, this would yield only node contraction between sister trees in the derivation structure. In MD-TAG, subject-aux inversion is the result of node contraction between sister trees in the derivation. This allows the derivation of the interleaved constructions which were problematic for classic TAG. The derivation in (8) is described in more detail in chapter 3, examples (20)-(23).

(8) Derivation of *Does Sam seem to like pizza?*

(9) Derivation-local node-contraction in interleaved constructions:
contraction between sisters
Above, in chapter 2, section 3.5, we claimed that positing phrase-sized trees would not change the results of the sections that followed. Consider how following Hegarty (1993) in assuming that extended projections are derived rather than elementary trees revises the derivation structures. Auxiliary trees anchored by functional elements will adjoin into non-auxiliary initial trees anchored by lexical elements. These functional elements should not take arguments. Thus, a tree with positions for arguments will still be in the same relationship with question trees as before. Other cases of contracted nodes between mother and daughter trees under the CETM, such as subject-auxiliary inversion, would instead correspond to cases of contracted nodes between sister trees. Consider, for instance the derivation structure for *What did Emmy eat?*. (Recall that in Hegarty’s proposal, the combinatory operations do not distinguish between Vs, Is, and some Cs. Thus, though the *eat* tree has a VP root, the Hegarty system allows it to be substituted into the CP node of the *wh*-question tree. Similarly, the *subject-auxiliary inversion* tree may adjoin into the *eat* tree at its VP root, even though the root and foot of the *subject-auxiliary inversion* tree are CPs.)

(10) Derivation for *What did Emmy eat?* under Hegarty’s phrase-sized elementary tree system
Thus far, we have considered examples where the locality condition allows node contraction in the derivation of grammatical constructions. Let us now consider the node contractions that are prohibited. Imposing the locality condition in (1) correctly predicts several island effects in English.

4.3 RELATIVE CLAUSES

The example below illustrates the unacceptability of extraction from inside a relative clause.

(12) *Which book did Alice meet the guy who had written?*

In determining the elementary trees that would be involved in the derivation of (12), I will assume a maximum of one lexical head per elementary tree, a condition satisfied by every tree abiding by the CETM. As before, I assume the use of Hegarty style sized trees will not yield a different outcome than using elementary trees that include functional projections. Here, I employ the latter, elementary trees headed by a
single lexical head and its associated functional projections. Using larger trees appears to be the safer approach, just in case my assumption that the use of extended projection trees or Hegarty style trees makes no difference in the size of the local domain for node contraction is wrong. Larger trees are likely to result in larger amounts of structure that are considered local, which provides the most likely environment for node contraction to succeed in satisfying the condition in (1). Additionally, the derivation will be easier to follow with fewer trees.

Deriving the sentence in (12), then, would involve the elementary trees below. Note that this set includes a relative clause tree for creating relative clauses, the tree second from the left in the top row. The relative clause tree is essentially the *wh*-question tree with a recursive piece of structure added to allow it to modify a DP. We shall see that the derivation fails because of a prohibition against contraction between the two trees that are boxed.

(13) Elementary trees that would be involved in deriving (12)
In (14), the combination of all trees is shown. The *had-written* tree substitutes into the relative clause tree. The relative clause tree adjoins into the object DP, the *the-guy* tree. This *the-guy* tree substitutes into the *did-meet* tree. The subj-aux tree adjoins into the *did-meet* tree at its CP root. The *did-meet* tree substitutes into the *wh*-question tree. Node contraction takes place between nodes in the subj-aux tree and the *did-meet* tree, between the rel-clause tree and the *had-written* tree, and between the *wh*-question tree and the *had-written* tree. Finally, the DP trees substitute into the argument slots, the trees headed by *wh*-elements substituting into multiply-dominated DP nodes.

(14) Combination of trees in (13)

The corresponding derivation structure is given in (15). To achieve the desired displacement of the object of *written*, a pair of DP nodes, one from the *had-written* tree inside the relative clause and one from the *wh*-question tree, must contract. Such a contraction would violate the locality constraint on contraction. Nodes in the clause of the relative clause may contract only with a node in its parent tree, the relative clause tree.
Nodes in the wh-question tree can contract only with nodes in its daughter tree, the matrix clause. Extraction from a relative clause would involve contraction between great-great-grandma and great-great-grandchild.

Failed derivation for *Which book did Alice meet the guy who had written?*

The derivation above assumes that the relative clause adjoins into the object DP, as is standard in the TAG literature. Suppose, however, that the relative clause tree is adjoined to the matrix clause instead. That is, the *had-written*-tree substitutes into the relative clause tree. The relative clause tree adjoins into the object DP of the *did-meet* tree. The subj-aux tree also adjoins into the *did-meet* tree, but at its CP root. Finally, the *did-meet* tree substitutes into the *wh*-question tree. This alternative minimizes the distance between the relative clause itself and the *wh*-question tree, making it the more generous of the two possibilities. However, the decrease in distance is not sufficient to
achieve derivational locality. Extraction from a relative clause would involve contraction between great-grandma and great-grandchild.

(16) Alternative combination of trees in (13)

(17) Second failed derivation for *Which book did Alice meet the guy who had written?
Interestingly, even if the set of elementary trees were to contain a recursive \textit{wh}-tree instead of our non-recursive \textit{wh}-tree, the extraction would still be banned by the locality condition. If a recursive \textit{wh}-tree adjoined into the matrix clause, the contraction would be between aunt and niece, which still would not meet the requirements of (1).

4.4 EXTRACTION FROM COMPLEX-NPS

Complex NPs also exhibit island effects.

\begin{equation}
* \text{Which book did you hear the claim that Sophie wrote?}
\end{equation}

Under the same assumptions given in section 4.3, the elementary trees that would be involved in deriving (18) are given below, with the structure for the DP \textit{the claim} more fully articulated than other representations of DPs here. This shows the locus of the sentential argument to the noun \textit{claim} (from which extraction takes place). The combination of these trees is given in (20). As above, we can derive the island effect from the non-local nature of contraction between the boxed trees.

\begin{equation}
\text{Trees that would be involved in deriving (18).}
\end{equation}
In the derivation structure below, it is shown that extraction from a complex NP would involve unpermitted node contraction between two trees in a great-grandmother-great-grandchild relationship. Nodes in the *wh*-question tree can contract only with nodes in its daughter tree, the matrix clause. Nodes in the complement clause of N may contract only with a node in the DP tree it dominates or a node in the DP tree that it is dominated by.

(21) Failed derivation for *Which book did you hear the claim that Sophie wrote?*
4.5 ADVERBIAL MODIFIER ISLANDS

Adverbia l modifiers do not allow their elements to be extracted.2

(22)  * Whodid Jane buy a new house after t came a raise?

Again, we can attribute the island effect to the non-local derivational relationship between the boxed trees and the barring of node contraction between the two.

(23) Trees that would be involved in deriving (22).

\begin{itemize}
  \item[(i)] a. Sophie is going to the movies with Courtney.
  \item[b. Who, is Sophie going to the movies with t?]
  \item[(ii)] a. Sami and Bethany will hold the banquet in room 17, the Astor Ballroom.
  \item[b. [Which room], will Sami and Bethany hold the banquet in t?]
\end{itemize}

In this system, we are forced to analyze these as sentential modifiers.

---

2 There are apparent counter-examples, such as the b examples below:

(i) a. Sophie is going to the movies with Courtney.
  b. Who, is Sophie going to the movies with t?

(ii) a. Sami and Bethany will hold the banquet in room 17, the Astor Ballroom.
  b. [Which room], will Sami and Bethany hold the banquet in t?
The derivation tree that corresponds to the derivation in (24) shows that the question tree may only contract with nodes in its sole daughter tree, the matrix clause. Nodes in the modifying clause may only contract with nodes in its mother tree, the auxiliary tree anchored by the preposition *after*. As above, the extraction needed for generating (22) involves an unpermitted contraction between trees in a great-grandmother-great-grandchild relationship.
Let us briefly compare the case of extraction out of adverbial adjuncts to that of moving an auxiliary associated with a raising verb. The combinatorial machinery does not distinguish between the structures anchored by adverbial modifiers and the structures anchored by raising verbs. Both are VP recursive auxiliary trees which are adjoined into an elementary tree anchored by a verb. The subject-auxiliary inversion in the interleaved cases, such as *What does Sam seem to like?*, depicted above in (3b), indicates that auxiliary trees anchored by raising verbs are not domains barring extraction. There are two crucial differences between the interleaved case and the adverbial modification case, exemplified above in (17). First, whereas the extraction takes place out of the auxiliary tree itself in the interleaved case, in the adverbial modification case, an additional piece of structure must substitute into the auxiliary tree anchored by the adverb, and the extraction must take place out of this piece of structure. This is the source of one extra generation gap to be bridged. Second, in the interleaved case, the node in the auxiliary
tree contracts with a node in the *subj-aux* inversion tree, which is itself adjoined into the verb-anchored elementary tree. In the adverbial modification case, an argument contracts with a node in the *wh*-question tree, which we have posited to be non-recursive. That is, in the interleaved case, the *subj-aux* inversion tree is in the daughter of the verb-anchored tree while in the adverbial modification case, the *wh*-question tree is the mother of the verb-anchored tree. This is the second source of additional derivational distance.

4.6 COORDINATE STRUCTURE CONSTRAINT

The coordinate structure constraint is a two part constraint that bans extraction of elements from within a conjunct and against extraction of a conjunct itself.

(26) a. *What₁ will Jane [[buy t₁] and [might wrap a gift]]?*
b. *[[What], did Alice cook [ t₁ and [pasta]]]?*

Again, our contraction locality constraint rules such extractions out.

4.6.1 CSC PART 1: NO EXTRACTION OF AN ELEMENT FROM WITHIN A CONJUNCT

Consider the elementary trees and derivation tree that correspond to *What₁ will Jane buy t₁ and might wrap a gift?*

---

3 Cases of ATB movement are exceptions. I do not have an account for why extraction of parallel elements makes this island permeable.

(i) *What₁ did Jane buy t₁ and Alice wrap t₁?*
(27) Trees that would be involved in deriving (26a).

(28) Combination of trees in (27)

Because each conjunct must substitute into the coordination tree which must then substitute into the wh-question tree, the conjuncts and the wh-question are in a grandmother-granddaughter relationship. Contraction of nodes between these two trees would not be considered local.
(29) Failed derivation for *What, will Jane buy, and might wrap a gift

4.6.2. CSC PART 2: NO EXTRACTION OF A CONJUNCT

Consider the combination of elementary trees and the derivation structure that correspond to *What, did Alice cook, and pasta?

(30) Trees that would be involved in deriving (26b).
The derivation tree below shows clearly that extraction of a conjunct would require contraction between nodes in a grandmother tree and granddaughter tree. Such a contraction fails to satisfy the contraction locality constraint.

Failed derivation for *What, did Alice cook t, and pasta?
For both cases of the CSC, the coordination tree can be considered the island creating tree.

The analysis of coordination adopted here involves an elementary tree headed by a conjunction which takes two conjuncts as arguments. As the conjuncts must substitute into the coordination tree, the coordination tree is each conjunct’s parent and nodes belonging to the conjunct may not contract with nodes in higher trees without violating the contraction locality constraint.

4.7 WH-ISLANDS

As example (29) illustrates, extraction out of a wh-phrase is not possible in English.

(33)  *What book did Mark ask whom you had given?*

If we would like to maintain the account adopted thus far for cyclic wh-movement, the elementary tree anchored by what becomes the matrix verb ought to adjoin into the verb of its embedded clause. Deriving this sentence in such a manner would involve the elementary trees pictured below. As above, the trees that are boxed will be the locus of the problem.
(34) Elementary trees needed for deriving (33)

There is actually no derivation that uses exactly this set of trees. Note the two trees in the upper left box. Each is needed for displacing the *wh*-DPs. Both are rooted in a QCP node. However, there is no substitution site that is a QCP node, so these trees cannot be incorporated into the derived tree via substitution. At best, one of the QCPs may be the root of the entire derived tree while the other QCP (and any tree it has combined with) remains disconnected. No valid derivation can use both.

Suppose, however, that the elementary tree anchored by the matrix verb was not recursive and did house a substitution node for a QCP. That is, suppose that *ask* takes a QCP complement.
(35) Alternative set of trees for deriving (33)

These trees in (35) can now be combined as shown in (36) below.

(36) Combination of trees in (35)
Although the trees may now be combined, the contraction of the DP nodes into which the *what book* tree substitutes spans too many generations in the derivation to satisfy the condition in (1).

(37) Derivation structure corresponding to tree combination in (36)

This account for wh-islands crucially depends on the non-recursive status of the wh-tree and the absence of another wh-tree with two DP contraction nodes. Suppose, however, there existed a natural language just like English except that either the adjoining operation or the set of node labels did not distinguish between QCP and CP. The set of elementary trees would then include a recursive wh-tree. Here, let us label the nodes as though there is no QCP label, so as to emphasize its recursiveness.
Recursive wh-tree in Pseudo-English I

In this language, Pseudo-English I, the tree in (38) could adjoin into the root node of non-interrogative elementary trees. The presence of such a tree in the set of elementary elements would make a derivation possible. In this derivation, the recursive wh-trees and the lower clause are sisters in the derivation tree. Under the locality condition in (1), contraction between the two is allowed.

Derivation for *What book did Mark ask whom you had given?* in Pseudo-English I
Extraction out of \( \text{wh} \)-islands is billed here as possible in Pseudo-English, but such extractions are in fact permitted in actual languages, such as Romanian (Comorovski 1986) and Bulgarian (Rudin 1988). Thus, one possible account of this cross-linguistic difference is that the set of elementary trees in languages which allow extraction out of \( \text{wh} \)-islands differ from the set of English elementary trees with respect to characteristics of their \( \text{wh} \)-question trees.

In the Pseudo-English example above, the \( \text{wh} \)-elements are arguments in the same clause, but in languages that allow extraction out of \( \text{wh} \)-islands, this is not a necessary condition for extraction out of \( \text{wh} \)-islands. The following Romanian example below shows extraction of elements from different clauses. (42) shows the derivation for the English gloss in Pseudo-English. Because the recursive \( \text{wh} \)-question tree is an auxiliary tree, it is always possible for the \( \text{wh} \)-question tree to be a daughter of an elementary tree anchored by a verb in the derivation structure.
(41) Pentru care clauză vrei să află cine îi nu a decis încă ce va vota încă?
for which paragraph you want to learn who not has decided yet what will vote
“For which paragraph do you want to learn who has not decided yet what he will vote?” (Comorovski 1986)

(42) Derivation for *For which paragraph do you want to learn who has not decided yet what he will vote?* in Pseudo-English

The use of the recursive tree in (38) is not the only possible question-tree that will result in a derivation that accomplishes extraction out of *wh*-islands using local node contraction. Suppose, for example, that a single tree included two nodes marked for node contraction, as in (43). A derivation very similar to that in (40) would be possible. The difference between the two is that in (44), five elementary trees combine into the *had-given*-tree whereas in (40), six elementary trees do so.

(43) Recursive *wh*-tree with two contraction nodes, Pseudo-English II
Derivation structure for What book did Mark ask whom you had given? in Pseudo-English II

As in classic TAG, there is nothing in the machinery that rules out extraction from wh-islands. Neither does the contraction locality condition rule out such extractions on its own. Rather, the island effect arises from the contraction locality condition given the available elementary trees. Since the set of elementary trees differs from language to language and the islandhood of wh-phrases also differs from language to language, the outcome is a desirable one.

4.8 EXTRACTION FROM SUBJECT NPS

It is also not possible to extract an element out of a subject (in English).

* I wonder which book for me to read would upset Sophie.

Deriving this sentence would involve the elementary trees pictured below. As above, the problematic trees are boxed. As (47) and (48) show, the CP tree for DP to read DP substitutes into the would upset tree, which substitutes into the wh-question tree.
The elementary tree serving as the subject is the granddaughter of the wh-question tree.
Thus, contraction of their nodes would violate the contraction locality constraint.

(46) Trees that would be involved in deriving (45).

(47) Combination of trees in (44)
Failed derivation for *I wonder which book for me to read would upset Sophie.

It has been noted that the islandhood of subjects also varies across languages. For example, extraction of *-elements out of subjects is permitted in Japanese, Turkish, German, Hindi, Russian, and Hungarian (Cinque 1978, Stepanov 2001). Note that if the recursive *-question tree, introduced in (31), is an elementary tree in a language, then it may adjoin into the root of the embedded clause. Such a configuration would make the *-question tree the sister of the subject, and contraction of nodes between the two trees would be local. As Pseudo-English allows both extraction out of subjects and extraction out of *-phrases, a natural language that allows both extractions is predicted to be possible. The languages reported above to allow extraction out of subjects, however, do not have (only) English type movement in that overt movement is optional, partial, or banned in these languages. This suggests that the distribution of nodes marked for contraction among the elementary trees of these languages differs in a significant way from that of English. Above, we argued that *-islands are the combined result of the
members of the elementary tree set and the locality condition. We take this to be the case with subject islandhood as well. However, because it is not necessarily the case that the same type of elementary question-tree is involved in extraction from both \textit{wh}-islands and subject islands, the set of languages allowing extraction from subjects is not predicted to be a subset of the languages allowing extraction from \textit{wh}-phrases. (Recall that either the Pseudo-English I tree in (38) or the Pseudo-English II tree in (43) is sufficient to accomplish extraction out of \textit{wh}-islands.)

Note further, however, that while the non-English elementary tree given in section 4.7 allows for extractions that are disallowed in English, it will not allow for extraction out of relative clauses or complex NPs. The contractions required for those extractions involve trees which are even more “distant relatives” than the trees involved in \textit{wh}- and subject islands.

4.9 CONCLUSION
This chapter shows how imposing a locality condition on a syntactic operation at the derivational level, namely node contraction, gives rise to strong island effects. Specifically, the system restricts contraction to nodes belonging to trees in a mother-daughter or sister relationship in the derivation tree, and a number of island effects in English are correctly predicted. As extraction of elements from relative clauses, complex NPs, adverbial modifiers, and conjuncts would all involve node-contraction between trees in a non-local relationship, these extractions are prohibited. Other islands, such as \textit{wh}-islands and subject NPs, whose status varies from language to language, result from the
combined effect of the derivational locality condition and the elementary trees available in the language’s lexicon.

This division parallels the partitioning in classic TAG between island effects that arise as a direct consequence of the system and island effects that require reference to the language specific lexicon. The parallel results from counting a mother-daughter relationship in the derivation structure as local. Recall that in classic TAG, elementary trees may be as large as extended-projections. Thus, the phrase structure that comprises both the wh-question tree (mother) and the CP that substitutes into the question tree (daughter) is typically thought of as a single elementary tree. At the same time, the proposed locality condition also permits the generation of previously problematic interleaved constructions. This is made possible by allowing contraction across trees in a sister relationship in the derivation tree.

The fruitfulness of an approach that makes direct reference to the derivational level constitutes an argument in favor of a derivational level that is distinct from a phrase structural level.
5.1 INTRODUCTION

In the preceding chapters, I have adopted the view that when a lexical element appears to play multiple grammatical roles typically associated with distinct positions, that element is dominated by multiple parent positions. Under such a multi-dominance approach, the set of syntactic structures is expanded to include non-tree graphs. The question then arises of how a syntactic graph ought to be pronounced. Something must inform us, for example, that the object cookies in Joe bakes and Sam decorates cookies is always pronounced after the second verb and never immediately after the first, even though it is an element in both conjuncts.

In the tree-based approaches to such multiply-linked elements, a tree-based approach to linearization is, of course, sufficient. Typically, the formal definition of trees includes a precedence relation that orders all nodes not in a dominance relation, and the ordering of the terminal nodes is a subset of this precedence relation (Partee et al. 1990). It has been shown, however, that a precedence relation on nodes is not necessary for ordering the terminal nodes for certain classes of trees. For some classes, a linearization of terminals can be derived using dominance (Kayne 1994) or c-command (Frank and Vijay-Shanker 2001) as the primitive relation between nodes.

In a multi-dominance approach, the structures no longer conform to the condition on trees that guaranteed a linearization of the terminals, the non-tangling condition
(Partee et al. 1990, p440). The previous proposals for how linearization is accomplished in tree structures does not uniquely determine a linearization of the terminals (e.g. as discussed in Wilder 2001). This chapter shows, however, that it is not a necessary move to stipulate a precedence relation over the terminals of a graph. Instead, the proposal here assumes that the primitive relations in syntactic structures are immediate dominance and sister-precedence, an ordering between two sibling nodes, and that the information needed by the phonology to ultimately produce a string of words can be extracted from these primitive relations. The precedence of terminals is determined via a modification of the non-tangling condition on trees. Section 5.2 begins with a definition of syntactic graphs and moves to a presentation and discussion of the primitive and derived relations between nodes, including the precedence of terminals in subsection 5.2.3. Section 5.3 provides examples of successful linearizations yielded by the proposal.

5.2 NODE RELATIONS IN SYNTACTIC GRAPHS

5.2.1 SYNTACTIC GRAPHS

In Part I, we did not distinguish between syntactic graphs and phrase structures that can be generated by a MD-TAG system. Here, we consider the set of multi-dominance graphs on its own, independent of the system used to generate it. As we shall see, the graphs that can be generated by some MD-TAG (subject to the theta criterion) are a

---

1 A brief note on notational conventions may be helpful:

- Bold letters are used to denote sets.
- Italicized lower case is used to denote terminal nodes.
- Non-italicized, non-bold letters are used to denote nodes in general. Usually, these will be roman letters, but Greek letters are sometimes used to distinguish the nodes of particular relevance to a discussion.
- Script capitals are used to denote relations. For any relation $R$, $R^+$ denotes the transitive closure of $R$ and $R^*$ denotes the reflexive transitive closure of $R$. 

---

94
subset of the set of formal objects defined in (1). The proposal for how a linearization of terminals is accomplished does require access to the primitive relations in a syntactic graph, but it does not make use of information from the derivation. Because of its independence from how the final phrase structure was generated, the proposal does not crucially rely on a specific generation system.²

(1) Definition of syntactic graph
A syntactic graph is a five-tuple \(<N, Q, ID, SP, L>\), where
\[ N \] is a finite set, the set of nodes,  
\[ Q \] is a finite set, the set of labels,  
\[ ID \] is an irreflexive, intransitive, asymmetric relation in \( N \times N \), the immediate dominance relation  
\[ SP \] is an irreflexive, intransitive, asymmetric relation in \( N \times N \), the sister precedence relation  
\[ L \] is a function from \( N \) into \( Q \), the labeling function,
and such that the following conditions hold:

a. Single Root Condition
\[ \exists X \in N \text{ such that } \forall Y \in N, (X, Y) \in ID^* \]

b. Non-Overlapping Condition
\[ \forall X, Y \in N, \]
\[ i. \text{ if } (X, Y) \in ID, \text{ then } (X, Y) \notin SP, \text{ and } \]
\[ ii. \text{ if } (X, Y) \in SP, \text{ then } (X, Y) \notin ID. \]

c. Acyclicity Condition
\[ \forall X, Y \in N, \]
\[ \text{if } (X, Y) \in ID^+, \text{ then } (Y, X) \notin ID^+. \]

² The work here shares key similarities to Gärtner (1997), but as Gärtner’s proposal depends on mechanisms specific to the Minimalist Program, this is a point of departure.
Syntactic trees are special cases of syntactic graphs. They are subject to an additional condition that prohibits multi-dominance.

(2) Single Parent Condition for syntactic trees
\[ \forall X, Y, Z \in N, \]
\[ \text{if } (X \neq Y) \]
\[ \text{then } \neg (((X, Z) \in ID) \text{ and } ((Y, Z) \in ID)) \]

This conception of syntactic graphs takes immediate dominance and sister precedence to be the primitive node relations.\(^3\) For example, we will suppose that the tree in (3a) is a graphical representation of the immediate dominance relation (ID) and sister precedence relation (SP) in (3b). Note that (VP, fly) is not in the immediate dominance relation, nor is (This, pig) in the sister precedence relation.

(3) a. 
\[ \text{ID} = \{ (IP, DP), (IP, I'), (DP, D), (DP, N), (I', I), (I', VP), (D, This), (N, pig), (I, will), (VP, V) \} \]

b. 
\[ S\mathcal{P} = \{ (DP, I'), (D, N), (I, VP) \} \]

\(^3\) Frank (p.c.) notes that these are the two primitive relations that Chomsky argues we get “for free” from an operation argued to be a “virtual conceptual necessity,” the Merge operation. Merge is proposed to be one of the elementary steps in a minimalist grammar. It takes a pair of syntactic objects \( \alpha \) and \( \beta \) and forms a new syntactic object \( \{ \gamma, \{ \alpha, \beta \} \} \). If we think of \( \gamma \) as the root of the new syntactic object, immediate dominance is the relation of membership (\( \gamma \) immediately dominates \( \alpha \) and \( \gamma \) immediately dominates \( \beta \)), and sisterhood is the relation of being merged together (\( \alpha \) and \( \beta \) are sisters). (See Chomsky (2004) for discussion.)
Let $X \text{ID} Y$ denote $(X, Y) \in \mathcal{ID}$, i.e. “$X$ immediately dominates $Y$.”

Let $X \text{SP} Y$ denote $(X, Y) \in \mathcal{SP}$, i.e. “$X$ sister precedes $Y$.”

The single root condition is the same as that given for trees. The non-overlapping condition prohibits pairs of nodes from being in both the immediate dominance relation and the sister precedence relation. It is intended to be the syntactic graph analogue of the exclusivity condition for trees (Partee et al 1990, p440). Unlike the exclusivity condition, however, it allows pairs of nodes to belong to neither relation. The acyclicity condition specifies that the transitive closure of immediate dominance is asymmetric, thus preventing cycles in the graphs.

Also, as with syntactic trees, it is apparent from the graphs we have seen in previous chapters that distinct nodes can have identical labels attached to them. For example, the graph given for a transitive verb and its two arguments will have two nodes labeled DP. We want to be able to have two nodes with the same label without positing the two nodes to be the same formal object, so we distinguish between a node and its label. That is, in the graph below, DP, VP, pancakes, etc. are the labels for the nodes, not the nodes themselves. The labeling function maps nodes to their labels: categories in the case of the non-terminals and lexical items in the case of terminals. The labeling function is included in the formal definition, but this distinction between a node and its label is suppressed in subsequent discussion for simplicity. That is, a statement such as “Joe precedes pancakes” is really shorthand for stating “The node labeled Joe precedes the node labeled pancakes.” To distinguish between two nodes with the same label, I will make use of subscripts or enumerate the labels that appear more than once. For example,
if VP1 and VP2 are used as node labels, this is not meant to reflect a difference in category between the nodes bearing the labels.

(4)  *Joe likes pancakes.*

![Tree diagram](image)

5.2.2 DERIVED RELATIONS: DOMINANCE AND TRADITIONAL PRECEDENCE

The relation that in tree structures is the dominance relation can be derived by taking the reflexive transitive closure of immediate dominance. Returning to our example graph in (3a), the dominance relation is given in (5). Note that, in contrast to the immediate dominance relation, the dominance relation does include the pair (VP, *fly*).

(5)  Derived dominance relation for graph (3a)

\[
D = \overline{ID^*} = \overline{ID} \cup \{ (IP, IP), (DP, DP), (I', I'), (D, D), (N, N), (I, I), (VP, VP), (This, This), (pig, pig), (will, will), (V, V), (fly, fly), (IP, D), (IP, N), (IP, I), (IP, VP), (IP, This), (IP, pig), (IP, will), (IP, V), (IP, fly), (DP, This), (DP, pig), (I', will), (I', V), (I', fly), (VP, fly) \}
\]

Let \(X \overline{D} Y\) denote \((X, Y) \in D\), i.e. “\(X\) dominates \(Y\).”
The irreflexive, asymmetric, transitive relation we traditionally encode by precedence, denoted here as $\mathcal{P}$, is given in a four part definition in (6). First, (6a) tells us that all members of the sister precedence relation are also members of the traditional precedence relation. Second, the children of sisters inherit the precedence relation of their parents via cousin-precedence$^4$ given in (6b), a modification of the non-tangling condition given in (7). Third, we take the transitive closure of the relation that results from the union of the pairs that satisfy the membership requirements of (6a) and (6b). Fourth, no other elements stand in the precedence relation other than those licensed by (6a), (6b), and (6c). One way to think of the definition for precedence is as a process that reads the immediate dominance and sister precedence relations among nodes and returns orderings between nodes.

$^4$ The name cousin-precedence is meant to emphasize that inheritance is inherited by children of sisters. See Gärtner (1997) for the closely related Precedence Inheritance Condition (p. 120), which does not make use of sister precedence, and $\varphi$-Precedence (p. 179), which makes use of both sister precedence and the weak/strong feature distinction of the Minimalist Program.
(6) $\mathcal{P}$ (traditional conception of precedence)$^5$

a. If $\alpha$ sister-precedes $\beta$,
then $\alpha$ precedes $\beta$.

b. Cousin-precedence:
If $\alpha$ sister-precedes $\beta$, and
$\alpha$ dominates $X$, and
$\beta$ dominates $Y$,
then $X$ precedes $Y$.

c. If $X$ precedes $Y$ and
$Y$ precedes $Z$,
then $X$ precedes $Z$.

d. Nothing else is in $\mathcal{P}$.

(7) Non-tangling condition (Partee et al 1990):
If $\alpha$ precedes $\beta$, then all nodes dominated by $\alpha$ precede all nodes dominated by $\beta$.

For our example graph (3a), this derives the traditional precedence relation below.

(8) Derived traditional precedence relation for graph (2a)

$$\mathcal{P} = S\mathcal{P} \cup \{(DP, I), (DP, will), (DP, VP), (DP, V), (DP, fly), (D, I'), (D, I), (D, will), (D, VP), (D, V), (D, fly), (N, I'), (N, I), (N, will), (N, VP), (N, V), (N, fly), (This, I'), (This, I), (This, will), (This, VP), (This, V), (This, fly), (pig, I'), (pig, I), (pig, will), (pig, VP), (pig, V), (pig, fly), (D, pig), (This, N), (This, pig), (I, V), (I, fly), (will, VP), (will, V), (will, fly) \}$$

$^5$ The closure conditions on $\mathcal{P}$ that yield the same relation are:

a) $\forall \alpha, \beta$, if $(\alpha, \beta) \in S\mathcal{P}$, then $(\alpha, \beta) \in \mathcal{P}$.

b) $\forall \alpha, \beta, X, Y$, if $(\alpha, \beta) \in S\mathcal{P}$, $\alpha \not\mathcal{D} X$, and $\beta \not\mathcal{D} Y$, then $(X, Y) \in \mathcal{P}$.

c) $\forall X, Y, Z$, if $(X, Y) \in \mathcal{P}$ and $(Y, Z) \in \mathcal{P}$, then $(X, Z) \in \mathcal{P}$. 
Let $X \mathcal{P} Y$ denote $(X, Y) \in \mathcal{P}$, i.e. “$X$ traditionally precedes $Y$.”

$\mathcal{P}$ yields the linearization of terminals we expect: *This pig will fly.* The definition for traditional precedence yields the same results that we expect for all trees.

### 5.2.3 Derived Relation: Precedence of Terminals

Our definition of traditional precedence will not, however, succeed in yielding a unique ordering of the terminals in structures involving multi-domination. To see why, consider the structure in (9), an example of subject-aux inversion in yes-no questions.

\[(9)\]

The daughters of the root, namely I and IP, are in the sister precedence relation: I sister precedes IP. Because *will* is dominated by I and *pig* is dominated by IP, the cousin precedence condition tells us that *will* precedes *pig*. Notice now the sister precedence relation between the daughters of IP: DP sister precedes I'. Note that *will* is dominated by I'. Because *pig* is dominated by DP and *will* is dominated by I', the cousin precedence condition tells us that *pig* precedes *will*. We see that $\mathcal{P}$ is not guaranteed to be
asymmetric. This is a problem. If precedence is to provide an ordering for pronunciation, it cannot be the case that both will precedes pig and pig precedes will.

Assuming that structures like the one given in (9) are correct, we see that we must modify how ordering is inherited in cases of multidominance. First, let us only concern ourselves with linearizing the terminals, as these are the only elements whose ordering is necessary for the purposes of pronunciation. That is, we will take the domain and co-domain of the ordering relation, which we denote with the symbol », to be the terminals. From this point forward, what we mean by precedence is the relation », not the relation \( P \).

The set of terminals is defined in (10).

\[ (10) \quad \text{Definition of Terminals (T)} \]
\[ T = \{ X \mid X \in N, \text{ and } \neg \exists Y \in N \text{ such that } (X, Y) \in ID \} \]

Second, we will modify cousin-precedence by incorporating the notion of full-dominance (Wilder 2001). The definition of full-dominance and path\(^7\) are given in (11) and (12).

\[ (11) \quad \text{Full-dominance } \mathcal{FD} \text{ (Wilder 2001):} \]
\[ \alpha \text{ fully dominates } \gamma \text{ iff every path from } \gamma \text{ to the root of the sentence includes } \alpha. \]

---

6 Wilder uses full-dominance to modify Kayne’s (1994) Linear Correspondence Axiom, which makes use of the c-command relation but not the sister-precedence relation.

7 More precisely, I mean a simple directed path. A simple path is a path that does not repeat any nodes. A directed path is a path in which the direction of all the edges is the same as the direction of the steps. (Sipser 1997).
Definition of a path

A path from X to Y is a sequence of nodes \( (N_1, N_2, \ldots, N_k) \) such that \( X = N_1, Y = N_k \) for all distinct \( i, j \), \( N_i \neq N_j \), and for all \( 1 < i \leq k \), \( N_i \neq N_{i-1} \).

To illustrate the notion of full-dominance, let us return to the graph given in (9). In this graph, IP fully dominates This, pig, and fly, but not will. Will has two paths to the root, one which includes IP: (will, I, I', IP, CP), and one which does not include IP: (will, I, CP). The chart in (13) summarizes the terminals fully dominated by each node in the graph given in (9).

Full-dominance in (9)

<table>
<thead>
<tr>
<th>CP</th>
<th>FD</th>
<th>will, This, pig, fly</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>FD</td>
<td>will</td>
</tr>
<tr>
<td>IP</td>
<td>FD</td>
<td>This, pig, fly</td>
</tr>
<tr>
<td>DP</td>
<td>FD</td>
<td>This, pig</td>
</tr>
<tr>
<td>I'</td>
<td>FD</td>
<td>fly</td>
</tr>
<tr>
<td>D</td>
<td>FD</td>
<td>This</td>
</tr>
<tr>
<td>N</td>
<td>FD</td>
<td>pig</td>
</tr>
<tr>
<td>VP</td>
<td>FD</td>
<td>fly</td>
</tr>
<tr>
<td>V</td>
<td>FD</td>
<td>fly</td>
</tr>
</tbody>
</table>

Finally, we give the definition of ». The italics indicate where (14) differs from the earlier precedence definition given in (6).
(14)  » (precedence of terminals)
   a. If $\alpha$ and $\beta$ are terminals and $\alpha$ sister-precedes $\beta$,
      then $\alpha$ precedes $\beta$
   b. Full-dominance Cousin-precedence:
      If $\alpha$ sister-precedes $\beta$, and
      $\alpha$ fully dominates $X$, and
      $\beta$ fully dominates $Y$,
      then $X$ precedes $Y$.
   c. If $X$ precedes $Y$ and
      $Y$ precedes $Z$,
      then $X$ precedes $Z$.
   d. Nothing else is in ».

   The definition in (14) does not make it easy to see what we can conclude from
knowing that $x \, \, \, \, \, \, \, \, y$. We now provide a more compact statement of (14) intended to make
clear what can be assumed after being given the information that $x \, \, \, \, \, \, \, \, y$.

(15) Definition of $x \, \, \, \, \, \, \, \, y$
    $x \, \, \, \, \, \, \, \, y$ in graph $G$ iff
    $\exists X, Z \in N$ such that
    $\mathcal{FD} X$, $\mathcal{FD} Z$, $\mathcal{PD} X$, and
    $(z = y)$ or $(z \, \, \, \, \, \, \, \, y)$

In the case where $z = y$, $x$ precedes $y$ directly, i.e. without involving transitivity. Because
syntactic graphs are finite objects, the number of transitive steps relating two terminals is
also finite. This means that for each pair of terminals $(x, y)$ in the precedence relation,
there is a finite sequence of nodes that establishes the intermediate orderings whose
transitive closure yields \(x\) precedes \(y\). This is stated as the corollary below.

(16) Corollary 1

\[ x \rightarrow y \text{ in graph } G \iff \exists \text{ a sequence of nodes } \{X, N_1, N_1', N_2, N_2', \ldots N_i, N_i', N_{i+1}, N_{i+1}', \ldots N_n, N_n', Y\} \]

such that

\[
\begin{align*}
X & \mathcal{FD} x, \quad X \mathcal{SP} N_1, \quad N_1 \mathcal{FD} t_1, \\
N_1' & \mathcal{FD} t_1, \quad N_1' \mathcal{SP} N_2, \quad N_2 \mathcal{FD} t_2, \\
& \vdots \quad \vdots \quad \vdots \\
N_{i-1}' & \mathcal{FD} t_{i-1}, \quad N_{i-1}' \mathcal{SP} N_i, \quad N_i \mathcal{FD} t_i, \\
N_i' & \mathcal{FD} t_i, \quad N_i' \mathcal{SP} N_{i+1}, \quad N_{i+1} \mathcal{FD} t_{i+1}, \\
& \vdots \quad \vdots \quad \vdots \\
N_n' & \mathcal{FD} t_n, \quad N_n' \mathcal{SP} Y, \quad Y \mathcal{FD} y,
\end{align*}
\]

where \(t_i\) is a terminal in \(G\).

When \(x\) precedes \(y\) directly, then \(X = N_n', t_n = x, Y = N_1, \) and \(t_1 = y\).

As it will be useful to refer to the set of terminals fully dominated by a node, the
deinition for this set is given below. When this set contains any terminals that have a
pronunciation, we say that the node has phonological content. (Some terminals may be
“pronounced” as a pause or piece of intonation. What I mean here by a terminal with no
pronunciation is a terminal with no phonological realization at all.)

(17) Definition of \(t_X\)

\(t_X = \{w \mid w \in T, \text{ and } X \mathcal{FD} w\}\)

i.e. \(t_X\) is the set of terminals fully dominated by node \(X\).
(18) Definition of node with phonological content
A node X has phonological content iff
\[ \exists w \in T \text{ such that} \]
a. \( w \in t_X \), and
b. \( w \) is a pronounced terminal

In a syntactic graph with no multi-dominance, i.e. in a syntactic tree, there is no difference in the outcome with respect to the resulting precedence relation whether we use cousin-precedence with full-dominance, as in (14b), or without, as in (6b). For a non-tree syntactic graph, a graph with multi-dominance, (14b) eliminates some otherwise conflicting orderings. To see how precedence of terminals works in graphs with multi-dominance, let us return to example (9). Consider the sister precedence pairs that affect the multidominated node. I sister precedes IP, yielding the ordering \textit{will} precedes \textit{This}, \textit{pig}, and \textit{fly}. (Recall that IP does not fully dominate \textit{will}.) I also sister precedes VP, yielding the ordering \textit{will} precedes \textit{fly}. The other sister precedence pairs will provide the remaining ordering information needed. DP sister precedes I', yielding the information that \textit{This} and \textit{pig} precede \textit{fly}. (Note that I' does not fully dominate \textit{will}.) D sister precedes N, yielding the information that \textit{This} precedes \textit{pig}. The chart in (19) summarizes the ordering information directly derived from full-dominance cousin-precedence.

(19) Full-dominance cousin-precedence derived orderings in (9)

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>SP</td>
<td>IP</td>
<td>( \rightarrow )</td>
<td>\textit{will}</td>
</tr>
<tr>
<td>DP</td>
<td>SP</td>
<td>I'</td>
<td>( \rightarrow )</td>
<td>\textit{This}, \textit{pig}</td>
</tr>
<tr>
<td>D</td>
<td>SP</td>
<td>N</td>
<td>( \rightarrow )</td>
<td>\textit{This}</td>
</tr>
<tr>
<td>I</td>
<td>SP</td>
<td>VP</td>
<td>( \rightarrow )</td>
<td>\textit{will}</td>
</tr>
</tbody>
</table>
Derived precedence of terminals for graph (9)

\[ \Rightarrow = \{ \text{will, This}, \text{will, pig}, \text{will, fly}, \]
\[ \text{This, pig}, \text{This, fly}, \]
\[ \text{pig, fly} \} \]

Recall that \( x \Rightarrow y \) denotes \( (x, y) \in \Rightarrow \), i.e. “\( x \) precedes \( y \).”

The table below shows that the pronunciation that is consistent with these ordering statements is “Will this pig ____ fly?” not “___ This pig will fly.”

(21)

<table>
<thead>
<tr>
<th>Will this pig ____ fly?</th>
<th>will » This, pig, fly</th>
<th>will » fly</th>
<th>This » pig, fly</th>
<th>This » pig</th>
</tr>
</thead>
<tbody>
<tr>
<td>___ This pig will fly</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Finally, we also provide a definition of a linearization. A linearization is a total, transitive, and antisymmetric relation. For the \( \Rightarrow \) relation of a graph to be a linearization, \( \Rightarrow \) must specify an ordering between any two terminals and must not contain any terminals that mutually precede one another.

(22) Definition of linearization

A linearization is a total, transitive, and antisymmetric relation
Definition of a graph that satisfies antisymmetry

A graph $G$ satisfies antisymmetry iff

$\forall x, y$, where $x$ and $y$ are terminals in $G$:

if $(x \gg y)$, then $\neg (y \gg x)$

Definition of a graph that satisfies totality

A graph $G$ satisfies totality iff

$\forall x, y$, where $x$ and $y$ are terminals in $G$ and $(x \neq y)$:

$(x \gg y)$ or $(y \gg x)$

Note that whereas transitivity follows from the definition of $\gg$, antisymmetry and totality do not. In (25), for example, full-dominance cousin precedence tells us that the terminal $x$ precedes $w$ and that $w$ precedes $x$. Thus, the $\gg$ relation for the graph in (25) is not a linearization. In (26), full-dominance cousin precedence directly derives neither an ordering between $u$ and $w$ nor an ordering between $u$ and some terminal that precedes $w$.

Before moving on to show how the system works for cases besides yes-no questions, we turn to a short discussion on transitivity and its effects on the system.

Graph that violates antisymmetry
5.2.4 Transitivity

Full-dominance cousin precedence provides precedence information between two terminals $x$ and $y$ if and only if there exists a node $X$ that fully dominates $x$, a node $Y$ that fully dominates $y$, and $X$ and $Y$ are sisters. It is transitivity that links terminals for which no such nodes $X$ and $Y$ exist. The precedence relations that can be derived directly from full-dominance cousin precedence can be thought of as the basic building blocks upon which the rest of the precedence statements hang. It is not surprising, then, that transitivity may add ordering statements that can not be derived via the full-dominance
cousin precedence condition alone to the set of ordering statements associated with a graph. Similarly, there exist graphs where full-dominance cousin precedence fails to yield a set of pairs that constitute a linearization, but the transitive closure of those pairs does yield a linearization. This subsection abstracts away from specific linguistic examples and considers some general consequences of requiring the derived precedence relation » to be transitive. Later, in Chapter 6, we shall consider linguistic examples where transitivity affects the whether or not » is a linearization.

5.2.4.1 Transitivity and Antisymmetry

Above, example (25) showed that it is possible for the set of precedence pairs directly derived from full-dominance cousin precedence to contain an antisymmetry violation. When we take the transitive closure of the set of those precedence pairs, additional violations of antisymmetry of » can arise. For example, in the graph below, the sister precedence pairs give rise to symmetry between w and x. The sister precedence pairs do not, however, directly yield symmetry between w and u: We are told only that u precedes w. It is transitivity that allows us to add the pair w precedes u, via the information that w precedes x and x precedes u.
5.2.4.2 Transitivity and Totality

The precedence pairs that are added via transitivity may have the effect of satisfying totality where totality was violated in the set of precedence pairs directly derived from full-dominance cousin-precedence. For example, consider the graph depicting remnant movement in (28b). (As the node contraction representation of the remnant movement is not a familiar one, a representation reflecting the traditional movement account of the same remnant movement is given in (28a).) In this graph, some node that fully dominates \( v \) is in a sister precedence relation with some node fully dominating each of the other terminals, as the chart summarizing the orderings directly derived from the full-dominance cousin-precedence condition shows. However, none of the nodes that fully dominate \( x \) are in a sister precedence relation with any of the nodes that fully dominate \( z \). The ordering \( x \) precedes \( z \) depends on transitivity.
5.3 EXAMPLE CASES

Above, we considered in detail how the precedence relation given in (14) allows the linearization of terminals in a multi-dominance analysis of a yes-no question. We now turn to examples of some different constructions with multi-dominance analyses, cases in which the precedence relation » is a linearization of the terminals. The first example, Right Node Raising, is given in most detail. Understanding the discussion above of yes-no questions and the discussion immediately below should be sufficient for rendering the remaining cases straightforward.

5.3.1 SHARED OBJECT (RIGHT NODE RAISING)

The graph in (29) represents the two primitive relations given in (30).
(29) Structure for *Joe bakes and Sam decorates cookies*.

\[
\begin{array}{c}
\text{Structure:} \\
\end{array}
\]

\[
(30) \begin{align*}
\mathcal{JD} &= \{(BP, VP1), (BP, B'), \\
& \quad (VP1, DP1), (VP1, V1'), \\
& \quad (B', B), (B', VP2), \\
& \quad (DP1, Joe), \\
& \quad (V1', V1), (V1', DP3), \\
& \quad (B, and), \\
& \quad (VP2, DP2), (VP2, V2'), \\
& \quad (V1, bakes), \\
& \quad (DP3, cookies), \\
& \quad (DP2, Sam), \\
& \quad (V2', V2), (V2', DP3), \\
& \quad (V2, decorates) \} \\
\mathcal{SP} &= \{(VP1, B'), \\
& \quad (DP1, V1'), \\
& \quad (B, VP2), \\
& \quad (V1, DP3), \\
& \quad (DP2, V2'), \\
& \quad (V2, DP3) \}
\end{align*}
\]

Note first the members of sister precedence that affect the contracted node: (V1, DP3) and (V2, DP3). Consider first the first pair. Since V1 fully dominates *bakes* and DP3 fully dominates *cookies*, our full-domination cousin-precedence condition specifies that *bakes* must precede *cookies*. Next, consider the second pair, (V2, DP3). Since V2 fully dominates *decorates* and DP3 fully dominates *cookies*, our full-domination cousin-precedence condition specifies that *decorates* must precede *cookies*. Thus, the following pairs are members of »: (*bakes, cookies*) and (*decorates, cookies*).
Also note that the remaining sister precedence pairs (VP1, B'), (DP1, V1'), (B, VP2), and (DP2, V2') do not provide information about the linear placement of cookies, because none of the nodes in these pairs fully dominate cookies. These pairs tell us the relative ordering of other terminals.

Earlier, in the section on transitivity and its effects, we alluded to forthcoming examples of cases where satisfying totality depended on transitivity. This Right Node Raising is one such example. Note that it is only via transitivity that Joe, and, and Sam are each ordered with respect to cookies. Only one ordering consistent with the members of » is available. This makes the correct prediction for where cookies is pronounced:

(31)  a.  Joe bakes ____ and Sam decorates cookies.
    b.  * Joe bakes cookies and Sam decorates ____. 

(32)
5.3.2 Shared subject (coordinated VPs)

(33) Structure for *Joe eats cookies and drinks tea.*

Again, let us first examine the members of sister precedence that affect the contracted node:

<table>
<thead>
<tr>
<th>DPs</th>
<th>SP</th>
<th>V1'</th>
<th>→</th>
<th>Joe</th>
<th>eats, cookies</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPs</td>
<td>SP</td>
<td>V2'</td>
<td>→</td>
<td>Joe</td>
<td>drinks, tea</td>
</tr>
</tbody>
</table>

As before, the other members of sister precedence will provide the remaining ordering information needed:

<table>
<thead>
<tr>
<th>B</th>
<th>SP</th>
<th>VP2</th>
<th>→</th>
<th>and</th>
<th>drinks, tea</th>
</tr>
</thead>
<tbody>
<tr>
<td>VP1</td>
<td>SP</td>
<td>B'</td>
<td>→</td>
<td>eats, cookies</td>
<td>and, drinks, tea</td>
</tr>
<tr>
<td>V1</td>
<td>SP</td>
<td>DP1</td>
<td>→</td>
<td>eats</td>
<td>cookies</td>
</tr>
<tr>
<td>V2</td>
<td>SP</td>
<td>DP2</td>
<td>→</td>
<td>drinks</td>
<td>tea</td>
</tr>
</tbody>
</table>

This makes the correct prediction for where *Joe* is pronounced:

(34) a. *Joe eats cookies and _____ drinks milk.*

b. * _____ Eats cookies and Joe drinks milk.*
5.3.3 **Shared Subject and Object (Coordinated Vs)**

(35) Structure for *Joe bakes and decorates cookies.*

The members of $S\mathcal{P}$ that affect the contracted nodes provide the information below:

<table>
<thead>
<tr>
<th>DPs</th>
<th>$S\mathcal{P}$</th>
<th>V1'</th>
<th>$\rightarrow$</th>
<th>Joe</th>
<th>bakes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPs</td>
<td>$S\mathcal{P}$</td>
<td>V2'</td>
<td>$\rightarrow$</td>
<td>Joe</td>
<td>decorates</td>
</tr>
<tr>
<td>V1</td>
<td>$S\mathcal{P}$</td>
<td>DPo</td>
<td>$\rightarrow$</td>
<td>eats</td>
<td>cookies</td>
</tr>
<tr>
<td>V2</td>
<td>$S\mathcal{P}$</td>
<td>DPo</td>
<td>$\rightarrow$</td>
<td>decorates</td>
<td>cookies</td>
</tr>
</tbody>
</table>

The remaining members of $S\mathcal{P}$ order the terminals as below:

<table>
<thead>
<tr>
<th>VP1</th>
<th>$S\mathcal{P}$</th>
<th>B'</th>
<th>$\rightarrow$</th>
<th>bakes</th>
<th>and</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>$S\mathcal{P}$</td>
<td>VP2</td>
<td>$\rightarrow$</td>
<td>and</td>
<td>decorates</td>
</tr>
</tbody>
</table>

This correctly predicts:

(36) a. *Joe bakes _____ and ___ decorates cookies.*

b. *Joe bakes cookies and ___ decorates ______.*

c. *___ bakes ______ and Joe decorates cookies.*

d. *___ bakes cookies and Joe decorates ______.*
5.3.4 Subject wh-questions

(37) Structure for *Who did Emmy make eat spinach?*

The members of $S_\mathcal{P}$ that affect the contracted nodes provide the information below:

<table>
<thead>
<tr>
<th>$DP_1$</th>
<th>$S_\mathcal{P}$</th>
<th>$C'$</th>
<th>$\rightarrow$</th>
<th>Who</th>
<th>did, Emmy, make, eat, spinach</th>
</tr>
</thead>
<tbody>
<tr>
<td>$DP_1$</td>
<td>$S_\mathcal{P}$</td>
<td>$V2'$</td>
<td>$\rightarrow$</td>
<td>Who</td>
<td>eat, spinach</td>
</tr>
<tr>
<td>$I$</td>
<td>$S_\mathcal{P}$</td>
<td>$IP$</td>
<td>$\rightarrow$</td>
<td>did</td>
<td>Emmy, make, eat, spinach</td>
</tr>
<tr>
<td>$I$</td>
<td>$S_\mathcal{P}$</td>
<td>$VP_1$</td>
<td>$\rightarrow$</td>
<td>did</td>
<td>make, eat, spinach</td>
</tr>
</tbody>
</table>

The remaining members of $S_\mathcal{P}$ order the terminals as below:

<table>
<thead>
<tr>
<th>$DP_2$</th>
<th>$S_\mathcal{P}$</th>
<th>$I'$</th>
<th>$\rightarrow$</th>
<th>Emmy</th>
<th>make, eat, spinach</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_1$</td>
<td>$S_\mathcal{P}$</td>
<td>$VP_2$</td>
<td>$\rightarrow$</td>
<td>make</td>
<td>eat, spinach</td>
</tr>
<tr>
<td>$V_2$</td>
<td>$S_\mathcal{P}$</td>
<td>$DP_3$</td>
<td>$\rightarrow$</td>
<td>eat</td>
<td>spinach</td>
</tr>
</tbody>
</table>

This correctly predicts:

(38) a. *Who did Emmy make ___ eat spinach?*

b. *___ did Emmy make who eat spinach?*
5.4 CONCLUSION

This chapter has taken syntactic graphs to be comprised of extremely local relations, immediate dominance and sister precedence. Additionally, the chapter has shown how using the transitive closure of the ordering derived via full-dominance cousin precedence can yield a linearization of the terminals in a syntactic graph. This precedence of terminals relations is not sensitive to how the phrase structure was generated. We have also seen how the proposed linearization process straightforwardly derives a linearization for a number of multi-dominance analyses. These include core cases of coordinate structures as well as multi-dominance analyses of a yes-no question and a subject wh-question. The astute reader will have noted that all these cases involve contraction of nodes with consistent status as a left sister or right sister. This leaves object wh-questions, earlier analyzed as a right sister object position contracted with a left sister operator position, still to be addressed. We will do so in Chapter 6.
6.1 INTRODUCTION

As we mentioned briefly in Chapter 5, the derived precedence relation does not guarantee
a linearization for every graph. Applying the linearization process to some graphs will
give rise to conflicting information, a failure to satisfy antisymmetry. Other graphs may
not yield enough information to order all the terminals, a failure to satisfy totality.

Following Kayne (1994), we take linearizability to be a property of well-formed syntactic
structures. That is, neither antisymmetry violations nor totality violations are tolerated.

Adopting such a position has the consequence of restricting the set of potential
syntactic analyses. Section 6.2 examines the analyses we are forced to pursue as a result
of the antisymmetry requirement, the properties of graphs that satisfy antisymmetry, and
the relevance to the particular linguistic phenomena of Scandinavian Object Shift and
Mandarin sub-command. Section 6.3 examines the analyses we are forced to pursue for
some cases of coordination as a result of the totality requirement, and the properties of
graphs that satisfy totality. Following the conclusion to the main part of the chapter in
section 6.4, section 6.5 argues that the totality requirement suggests an anaphoric
approach to gapping constructions, and section 6.6 takes a preliminary look at deletion of
structure as an operation that evades instances of the two types of violations
6.2 THE ANTISYMMETRY REQUIREMENT

6.2.1 OVERVIEW

We ended chapter 5 with the admission that the structure proposed for object \(wh\)-questions was problematic under the assumption that the precedence relation of licit syntactic graphs must satisfy antisymmetry. In 6.2.2, we consider alternative structures for object \(wh\)-questions. In 6.2.3, we move towards a general characterization of the graphs that can be linearized under the proposed linearization process, and we see that a type of local order preservation is in effect. This is the type of order preservation that has been observed in Scandinavian object shift, which we discuss before returning to efforts to formally characterize syntactic graphs without antisymmetry violations in 6.2.4. Ideally, we would like to provide necessary and sufficient conditions for such graphs. Unfortunately, we will have to settle for a more modest goal here, a non-trivial necessary condition for satisfying antisymmetry, established in subsection 6.2.4.2. Following this subsection, an example that shows the condition is not a sufficient condition is given. The example is a case of remnant movement, described traditionally as the result of two movements: first, an element moves out of a constituent, and second, that entire constituent (with a gap from the first movement) moves. We then turn to an examination of different classes of remnant movement in subsection 6.2.5. It turns out that some types of remnant movement can be linearized by the proposed linearization process we have been exploring, though the predicted linearizations diverge from traditional linearizations associated with remnant movement. Finally, in 6.2.6, we see how this divergence allows
us to adopt a previously unavailable analysis for the binding of the Mandarin reflexive
zi4ji3.

6.2.2 AN APPARENT VIOLATION: OBJECT WH-QUESTIONS

Consider the structure that was proposed in Section 3.3, Part One, for wh-questions, repeated below as (1).

(1) Structure for *What did Emmy eat?*

Under the assumption that the system does not tolerate ordering conflicts, the structure given in (1) is not a possible analysis. Both (DP, CP1) and (V, DP) are members of $S\mathcal{P}$. By the full-dominance cousin-precedence condition, the first pair tells us that *what* is ordered before *eat*, but the latter tells us *eat* is ordered before *what*. This point of symmetry is clearly incompatible with a linear ordering.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>CP1 →</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DP</td>
<td>$S\mathcal{P}$</td>
<td>CP1</td>
<td>→</td>
<td><em>what</em></td>
<td>→</td>
<td><em>eat, did, Emmy</em></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>$S\mathcal{P}$</td>
<td>DP</td>
<td>→</td>
<td><em>eat</em></td>
<td>→</td>
<td><em>what</em></td>
<td></td>
</tr>
</tbody>
</table>
The problem arises because, as a daughter of VP, the object is a right sister (to a node with phonological content), but as a daughter of QCP, the object is a left sister (to a node with phonological content). To maintain the proposed linearization system, there are three possible analyses: the object must be analyzed as a left sister, as in (2), having a silent left sister, as in (3), or having a no sister, as in (4). Under the first possibility, the question immediately arises of the identity of the right sister of the object. Several possible right sisters of the object suggest themselves. We could appeal to Fukui and Takano’s (1998) suggestion that all arguments are positioned to the left of the verb. Unfortunately, such an analysis would create difficulties in linearizing the previously straightforward Right Node Raising case. Recall that in the structure assumed for RNR (see example (29) in subsection 5.3.1), the object’s status as a right sister is maintained in both conjuncts. As a result, the shared object is linearized as being pronounced in the right conjunct. Alternatively, we could posit a sentence final question marker, an element responsible in English for the intonational rise of the last element of a yes-no question or echo question. This is vulnerable to the criticism that question intonation is

---

1 Were the shared object a left sister in both conjuncts, an ordering conflict would be avoided, but the linearization proposal would give rise to a totality violation. Consider the structure below. Ignoring the ordering between the verb and the object (assume that it is correct or that elementary tree internal node contraction could establish the English V-O order during some other derivational step), the object would be unordered with respect to the subject in first conjunct, John. Note that the only nodes that fully dominate cookies are its preterminal, DPo, and the root, BP. None of the nodes that fully dominate John stand in a sister precedence relation with any node that fully dominates cookies nor any node that fully dominates a terminal preceding cookies. We discuss totality violations more fully in section 6.3.
likely to take scope of an entire phrase, though the suggestion is still a reasonable one. We could also posit a silent element carrying the operator component of the $wh$-element (which could also move). A number of works have assumed that the question operator is universally generated as a null operator and that the $wh$-DP is universally a variable (Cole and Hermon 1998, Cheng 1991, Aoun and Li 1993, Wantanabe 1993, and Tsai 1994).²

(2) A graph with the object as a left sister.

This suggestion of a silent operator may be the most plausible scenario under the possibility that the object is a right sister with a silent left sister. Again, if we posit that the operator also moves higher, an asymmetric ordering between the object and the operator ensues. However, since the proposed operator is silent, we could allow what is

---

² In such approaches, crosslinguistic differences in $wh$-movement are attributed to whether or not the operator and the variable are lexicalized as a single word. Cole and Hermon (1998) show how this approach predicts languages with overt $wh$-movement (e.g. English), $wh$-in-situ (e.g. Chinese), and partial $wh$-movement (Malay). In a node-contraction system, a parallel account is possible. It may be that the operator is universally marked for contraction while the differences across languages may be attributed to whether or not the $wh$-word is always marked for contraction (e.g. English), never marked for contraction (e.g. Chinese), or optionally marked for contraction (e.g. Malay).
technically an ordering conflict. Recall that the driving motivation for requiring syntactic structures to be linearizable is to ensure that the phonology can construct a pronunciation. We can reasonably assume that ordering conflicts between a pronounced terminal and a silent terminal are not problematic for the phonological system.

(3) A graph with the silent left sister.

The last possibility, that the object has no sister, has the appeal of allowing the object to move to the left, as in *wh*-questions, and to the right, as in Right Node Raising, while still satisfying the antisymmetry requirement. If this structure is to be generated by the MD-TAG system proposed in Part One, however, then we would need to permit a head node, D, to contract with a node in the spec CP position, a position thought to be a DP node, a phrase level node, not a head level node.
(4) A graph in which the object has no sister.

This discussion of object wh-questions is but one example of how the antisymmetry requirement bears on the space of possible analyses.

6.2.3 INFORMAL PROPERTIES OF ANTISYMMETRIC GRAPHS

6.2.3.1 Order preservation between sister nodes

Based on the discussion above and the examples in Chapter 5, it appears that in graphs that satisfy antisymmetry, each node has a consistent position relative to all its sisters. That is, a left sister should only contract with another left sister and a right sister should only contract to a right sister. In general, this intuition is correct. Maintaining sister consistency is a sufficient condition for maintaining antisymmetry. However, maintaining sister consistency is not necessary to ensure antisymmetry. Consider the case when two sisters are both multiply dominated. In this case, a node may be both a
right daughter and a left daughter so long as its position relative to its multiply dominated sister is maintained. This is possible when sister nodes are in crossing dependencies, as exemplified in (5). In (5a), the sister precedence pair (F sister precedes H) tells us that the terminals fully dominated by F precede those fully dominated by H. The sister precedence pair (H sister precedes I) tells us nothing about ordering, because node I does not fully dominate any material. The sister precedence pair (F sister precedes G) tells us that the terminals fully dominated by F precede those fully dominated by G, a set that includes the terminals fully dominated by H. This ordering is consistent with the ordering established by (F sister precedes H). In contrast, when the sisters are in nested dependencies, such as those in (6), antisymmetry violations arise. Consider (6a). The sister precedence pair (H sister precedes F) tells us that the terminals fully dominated by H precede those fully dominated by F. The sister precedence pair (F sister precedes G), however, tells us that the terminals fully dominated by F precede those fully dominated by G, a set that includes the terminals fully dominated by H.
Before more fully formalizing the structures that satisfy antisymmetry, let us turn to a linguistic example of the sort of preservation I have been discussing.

6.2.3.2 Scandinavian object shift: order preservation in use

It has been observed (Holmberg 1986) that Scandinavian languages maintain the relative ordering of the verb and the object, even when the two are not adjacent. In the presence of an overt auxiliary, as in example (7), the verb follows negation and certain types of adverbs and immediately precedes the object.

(7) Subj Aux Neg V Obj Swedish
    Jag har inte kysst Anna.
    I have not kissed Anna. (Sells 2001)

In the absence of an overt auxiliary, the verb appears before the negation and adverbials, as shown in (8). As the negation and adverbials are usually taken to mark the left edge of the VP, the order in (8) has been taken to show that finite main verbs in main clauses move to a higher position.

(8) Subj V Neg Obj Swedish
    Jag kysste inte Anna.
    I kissed not Anna. (Sells 2001)
When the verb has moved out of the VP and the object is a pronoun, the object also appears before elements denoting the left edge of the VP. As in the cases above, the verb continues to precede the object.

(9)  Subj V Obj Neg  
Jag kysste henne inte. 
I kissed her not (Holmberg 1999)

It is this leftward movement of the object pronoun out of VP that the term object shift refers to. Object shift is not possible when the verb remains inside the VP. In example (10), the presence of the auxiliary prevents the verb from moving, which subsequently blocks object shift. Similarly, in (12), the pronoun object of an embedded clause cannot be moved, because embedded main verbs do not move.

(10)  * Subj Aux Obj Neg V  
* Jag har henne inte kysst  
I have her not kissed. (Holmberg 1999)

(11)  Subj Aux Neg V Obj  
Jag har inte kysst henne.  
I have not kissed her. (Holmberg 1999)

(12)  * . . . Subj Obj Neg V  
* . . . att jag henne inte kysste  
. . . that I her not kissed (Holmberg 1999)
The prohibition against object shift without verb movement out of the VP has become the most familiar case of Holmberg’s generalization (1986). Holmberg’s observation, however, is not that an unmoved verb blocks object shift; Object shift is blocked by any overt material to its left within VP. This includes both particles and indirect objects.

(14)  a. * Subj V Obj Neg Prt       Swedish
       * Dom kastade me inte ut.
       they threw me not out. (Holmberg 1999)

   b. Subj V Neg Prt Obj
       Dom kastade inte ut me.
       they threw not out me. (Holmberg 1999)

(15)  a. * Subj V Obj Neg IO       Swedish
       * Jag gav den inte Elsa.
       I gave it not Elsa (Holmberg 1999)

   b. Subj V Neg IO Obj
       Jag gav inte Elsa den.
       I gave not Elsa it (Holmberg 1999)

Thus far, our object shift examples show the verb immediately preceding the object, suggesting an approach in which the coupling of object shift with verb movement arises from fusion of the pronoun and verb. However, the V topicalization example in
(16) and the adverbial intermingling shown in (17) show that adjacency of the verb and object is actually not a requirement. These examples cast doubt on a fusion approach.

(16) $V$ aux Subj Obj Neg Swedish

Kysst har jag henne inte (bara hållit henne I handen).

kissed have I her not (only held her by the hand) (Holmberg 1999)

(17) a. . . . . $V$ Subj Obj Adv Adv Adv Neg Swedish

Igår läste han dem ju alltså tolingen inte.

yesterday read he them as you know thus probably not

b. . . . . V Subj Adv Obj Adv Adv Neg

Igår läste han ju dem alltså tolingen inte.

yesterday read he as you know them thus probably not

c. . . . . V Subj Adv Adv Obj Adv Neg

Igår läste han ju alltså dem tolingen inte.

yesterday read he as you know thus them probably not

d. . . . . V Subj Adv Adv Adv Obj Neg

Igår läste han ju alltså tolingen dem inte.

yesterday read he as you know thus probably them not

(Sells 2001)

Instead, Fox and Pesetsky (2005) cast the object shift pattern in order preservation terms. In their proposal, word order is fixed over a small domain during intermediate points in a derivation, and the ordering may not be overwritten later in the derivation. In the linearization process used here, the notion of local domain is different, but, as we have seen above, order preservation is a central concept. When $verb \gg object$ order is
established via two sister nodes in each conjunct, then verb » object order will be maintained in the graph globally.

Further, if we take the verb and object to be sisters in Scandinavian languages, then the ban against symmetry in the proposed system predicts precisely the attested pattern of movement: movement of the verb occurs independently of movement of the object, but movement of the object depends on movement of the verb. The discussion above yielded the observation that satisfying the antisymmetry requirement generally constrained multiply dominated nodes to maintain a consistent position with respect to all its sisters. For example, a right daughter could only contract with another right daughter position. The exception, however, is when both sisters are multiply dominated. When two sisters are both multiply dominated, a node may be both a right daughter and a left daughter so long as its position relative to its multiply dominated sister is maintained. That is, a graph like (18a) or (19a) fails to satisfy antisymmetry while a graph like (18b) ((5a) repeated) or (19b) does satisfy antisymmetry.

(18) Comparison of schemas violating and satisfying antisymmetry
(19) Examples of attested and unattested object shift

Although the linearization process is not sensitive to categories, Holmberg’s generalization (1986) can be recast in this system as a prohibition against contradicting the ordering established between sisters. It is no surprise that overt material within the VP blocks object shift. Given either of two reasonable structures for ditransitive verbs and verbs with particles, the antisymmetry requirement predicts certain prohibitions on multi-dominance that straightforwardly concur with the pattern reported by Holmberg.

6.2.3.2.1 Ternary branching

For both a construction with a ditransitive verb and a construction with a verb particle, one possible structural configuration is a ternary branching configuration: the verb, the direct object, and indirect object/particle are all in the sister precedence relation with one another. In (20a), for example, the preterminal dominating *kastade*, the preterminal dominating *ut*, and the node dominating *me* are all daughters of the node labeled V'2. Under these assumptions, movement of the rightmost sister will not be
possible without movement of the middle sister. When the middle sister does not move, there are two conflicting orderings established. One ordering is established via the middle sister’s sister precedence relation with its right sister. For example, Prt sister precedes the DP object in (20a), establishing the order *ut precedes me*. However, the sister precedence relation between the DP object and V’1, which fully dominates the middle sister, establishes the reverse ordering. In (20a), V’1 fully dominates *ut*, so the sister precedence pair (DPO, V’1) yields the ordering *me precedes ut*.

(20) Unavailability of direct object shift across a verb-particle or an indirect object

\[
\begin{align*}
\text{a.} & \quad \star \quad \text{VP} \\
& \quad \text{DP} \quad \text{Dom} \\
& \quad \text{T'} \\
& \quad \text{X} \\
& \quad \text{V'1} \\
& \quad \text{Neg} \\
& \quad \text{V'2} \\
& \quad \text{Prt} \\
& \quad \text{ut} \\
& \quad \text{DP} \quad \text{Dom} \\
& \quad \text{me} \\
\end{align*}
\]

\[
\begin{align*}
\text{b.} & \quad \star \quad \text{VP} \\
& \quad \text{DP} \quad \text{Jag} \\
& \quad \text{T'} \\
& \quad \text{X} \\
& \quad \text{V'1} \\
& \quad \text{Neg} \\
& \quad \text{V'2} \\
& \quad \text{Prt} \\
& \quad \text{ut} \\
& \quad \text{DP} \quad \text{Dom} \\
& \quad \text{Elsa} \\
\end{align*}
\]

6.2.3.2.2 Binary branching

If instead, one commits to binary branching structure, then a reasonable analysis is one in which one of the daughters of the V’2 node dominates the verb and indirect object/particle and the other dominates the direct object. Together, the verb and indirect object/particle comprise a constituent that sister precedes the direct object. In this case, movement of the direct object will not be possible without movement of all phonological
material in its sister. Consider (21a), in which the verb moves but the particle does not.
Since the V'3 node fully dominates the particle *ut*, and V'3 sister precedes the DP object,
*ut* should precede *me*. However, the DP object also sister precedes V'1. V'1 also fully
dominates *ut*, which implies the reverse ordering, *me* precedes *ut*.

(21) Unavailability of direct object shift across an indirect object

![Diagram]

In the treatment of Scandinavian object shift in this system, the order preservation
effects are an expected outcome of the antisymmetry requirement.

6.2.4 Formal properties of antisymmetric graphs

6.2.4.1 Restriction to binary branching and self-c-commanding multi-dominated
nodes

It is difficult to say much more about the properties of graphs that satisfy
antisymmetry without further structural restrictions. As the hypothesis that a formal
universal of human language is that human grammar allows at most binary-branching nodes (Kayne 1984) has been especially influential, let us consider the set of graphs that conform to this hypothesis. That is, we limit the discussion to binary branching graphs.

(22) Definition of binary branching graph

Graph $G$ is a binary branching graph iff

$\forall$ nodes $\alpha$, $\beta$, and $\gamma$,

\begin{align*}
\text{if} & \quad \alpha \mathcal{S} \beta, \\
\gamma & \mathcal{I} \mathcal{D} \alpha, \text{ and} \\
\gamma & \mathcal{I} \mathcal{D} \beta,
\end{align*}

then $\neg \exists \delta$ such that ($\gamma \mathcal{I} \mathcal{D} \delta$) and ($\delta \notin \{\alpha, \beta\}$).

When a node $\alpha$ is both a right and left sister, it appears that maintaining the ordering between its sister (even a multi-dominated sister) $\beta$ can be achieved when the least node fully dominating $\alpha$ does not also fully dominate $\beta$. The notion of least node fully dominating $\alpha$ is defined in (24). Informally, if $\alpha$ has a mixed sisterhood status, then $\alpha$’s “lower” sister, $\beta$, contracts “past” $\alpha$. Formally, the conjecture is given below.
Conjecture 13:
if $G$ is a graph that satisfies antisymmetry,
then: $\forall \alpha, \beta, X, Y$ with phonological content,
( if $\alpha \mathcal{SP} \beta$ and
$X \mathcal{SP} \alpha$,
then $\neg (\text{LNFD}_\alpha \mathcal{FD} \beta)$ )
and
( if $\alpha \mathcal{SP} \beta$ and
$\beta \mathcal{SP} Y$,
then $\neg (\text{LNFD}_\beta \mathcal{FD} \alpha)$ )

Definition of least node fully dominating $\alpha$ ($\text{LNFD}_\alpha$)
$\forall$ nodes $\alpha, \gamma, \text{and} \delta \in N$ where $\alpha \neq \gamma \neq \delta$
$\gamma$ is the $\text{LNFD}_\alpha$ iff
$\gamma \mathcal{FD} \alpha$, and
$\neg \exists \delta \in N$ such that ($\delta \mathcal{FD} \alpha$) and ($\gamma \mathcal{D} \delta$).

Proving even Conjecture 1 has proven challenging, because it is unclear what information relevant to ordering can be derived from knowing the least node fully dominating a node does not fully dominate the node’s sister. Let us further restrict the graphs under consideration to graphs in which multi-dominated nodes self c-command. C-command is a derived relation that has proved to be useful in characterizing a number of syntactic phenomena, including movement. This c-command condition on multi-dominance is a direct incorporation of the traditional c-command condition on movement.

---

3 Conjecture 1 is expressed in the manner given for the sake of transparency. It can be expressed more succinctly as follows: if $G$ is a graph that satisfies antisymmetry,
then: $\forall X, Y, Z$ with phonological content,
if $X \mathcal{SP} Y$, and
$Y \mathcal{SP} Z$,
then $\neg (\text{LNFD}_Y \mathcal{FD} Z)$ and $\neg (\text{LNFD}_Y \mathcal{FD} X)$
whereby a moved element’s final position must stand in a c-command relation to its original position. Informally, this means that a node’s “upper” address must c-command its “lower” address. This can be restated as in the definition in (26): if a node $\gamma$ has two parents, the parents must stand in a dominance relation with one another. Because of the transitivity of dominance, we know that whichever parent dominates the other parent will then dominate $\gamma$ “twice”: once as $\gamma$’s parent node and once via the other parent, which also dominates $\gamma$.

(25) Definition of c-command\(^4\)

$X$ c-commands $Y$ iff

$X \neq Y$

$\neg (X \mathcal{D} Y)$ and $\neg (Y \mathcal{D} X)$ (i.e. neither $X$ nor $Y$ dominates the other), and

$\exists Z \in N$ such that $(Z \mathcal{I} \mathcal{D} X)$ and $(Z \mathcal{D} Y)$

(26) Definition of c-command multi-dominance graph

Graph $G$ is a c-command multi-dominance graph iff

$\forall$ nodes $\alpha$, $\beta$, and $\gamma$,

if $\alpha \mathcal{I} \mathcal{D} \gamma$,

$\beta \mathcal{I} \mathcal{D} \gamma$, and

$\alpha \neq \beta$,

then $(\alpha \mathcal{D} \beta)$ or $(\beta \mathcal{D} \alpha)$.

---

\(^4\) The notion of c-command is refined in slightly different ways in various works. Traditionally, the definition is given as follows:

(i) $X$ c-commands $Y$ iff

$X \neq Y$,

neither $X$ nor $Y$ dominates the other, and

the lowest branching node that dominates $X$ also dominates $Y$.

The last condition is sometimes given as: “every node that dominates $X$ also dominates $Y,”$ because in trees, they are equivalent. The version used here is modified to accommodate multi-dominance. It says that there is a parent of $X$ that also dominates $Y$. 

137
Note that I do not mean to claim that binary branching and c-command multi-dominance are necessary conditions on grammaticality. For example, the structures assumed for coordination examples in subsections 5.3.1, 5.3.2, and 5.3.3 (see (29), (33), and (35) in Chapter 5), graphs in which a multi-dominated node does not self c-command are shown to be linearizable. The restriction to binary branching, c-command multi-dominance graphs is imposed for the purpose of carving out a manageable subset of the graphs that can be generated by the MD-TAG system for exploration.

6.2.4.2 Necessary conditions for satisfying antisymmetry

Under the restriction to binary branching, c-command multi-dominance graphs, knowledge about the relationship between a node Z and the LNFD_α can lead to knowledge about the relationship between a node Z and the daughters of the LNFD_α.

Consider the schema below. Suppose α is a multi-dominated node with a parent LNFD_α and a parent Y. α’s sister under LNFD_α is X and α’s sister under Y is β.

(27)

Saying that the LNFD_α does not fully dominate β is the same as saying that the “upper” sister of α, X, does not fully dominate the “lower” sister of α, β. Because of binary branching, the LNFD_α has only two children: α and one of its sisters X. Because α must
self-c-command, X dominates \( \alpha \)’s “lower” address, the address in which it is a sister of \( \beta \). This means that X also dominates \( \beta \). If \( \beta \) were not fully dominated by LNFD\( \alpha \), \( \beta \) must have contracted with a node that c-commands LNFD\( \alpha \). In traditional terms, \( \beta \) must have moved above LNFD\( \alpha \). If this were true, then \( \beta \) would not be fully dominated by X.

Consider the converse. If we knew that \( \beta \) was not fully dominated by X, then \( \beta \) must have contracted with a node that c-commands X. Only one position, the position of the sister of X, c-commands X and remains fully dominated by LNFD\( \alpha \). The position of the sister of X is already occupied by \( \alpha \), and since sister precedence is irreflexive, \( \alpha \neq \beta \). Thus, \( \beta \) must have contracted with a node above LNFD\( \alpha \), which means LNFD\( \alpha \) does not fully dominate \( \beta \). Therefore, under the restriction to binary branching and multi-dominated nodes that self-c-command, we can rewrite Conjecture 1 as Theorem 1, a more restrictive condition than Conjecture 1.
Theorem 1: Necessary Conditions for satisfying antisymmetry

if $G$ is a graph that satisfies antisymmetry, then: $\forall \alpha, \beta, \gamma, \delta$ with phonological content,

(if $\alpha SP \beta$, and

$\gamma SP \alpha$,

then $\neg (\gamma FD \beta)$

and

(if $\alpha SP \beta$, and

$\beta SP \delta$,

then $\neg (\delta FD \alpha)$


Below, (29) provides two configurations that fail to meet the conditions given in the consequent. The first case corresponds to a scenario in which $\alpha$ has contracted: $\alpha$’s lower position is a left sister while $\alpha$’s higher position is a right sister. The second case is the mirror scenario. $\beta$ has contracted, but it is $\beta$’s higher position that is a left sister while $\beta$’s lower position is a right sister.

\[ (29) \]

5 As with Conjecture 1, the form of Theorem 1 in the text is intended to facilitate transparency. Theorem 1 can also be expressed more succinctly: if $G$ is a graph that satisfies antisymmetry, then: $\forall X, Y, Z$ with phonological content, if $X SP Y$, and $Y SP Z$, then $\neg (X FD Z)$ and $\neg (Z FD X)$
Proof by contradiction.

Suppose not.

Then \( \exists \) graph \( G' \) that satisfies antisymmetry where one of the following cases is true:

Case 1: \( \alpha, \beta, \) and \( \gamma \) are nodes in \( G' \) with phonological content

\[
\alpha \mathcal{SP} \beta, \\
\gamma \mathcal{SP} \alpha, \text{ and} \\
\gamma \mathcal{FD} \beta.
\]

By the full-dominance cousin-precedence condition, \( \alpha \mathcal{SP} \beta \) tells us that the terminals fully dominated by \( \alpha \) all precede the terminals fully dominated by \( \beta \).

\[
\alpha \mathcal{SP} \beta \quad \rightarrow \quad t_\alpha \triangleright t_\beta
\]

By the full-dominance cousin-precedence condition, \( \gamma \mathcal{SP} \alpha \) tells us that the terminals fully dominated by \( \gamma \) all precede the terminals fully dominated by \( \alpha \).

\[
\gamma \mathcal{SP} \alpha \quad \rightarrow \quad t_\gamma \triangleright t_\alpha
\]

Since full dominance is transitive and \( \gamma \) fully dominates \( \beta \), every terminal fully dominated by \( \beta \) is also fully dominated by \( \gamma \).

\[
t_\beta \subseteq t_\gamma
\]

By the definition of subset, whatever is true of the members of \( t_\gamma \) is also true of each member of \( t_\beta \).

\[
t_\gamma \triangleright t_\alpha \text{ and } t_\beta \subseteq t_\gamma \quad \rightarrow \quad t_\beta \triangleright t_\alpha
\]
Since $\alpha$ and $\beta$ have phonological content, this is a contradiction.

$$t_\alpha \triangleright t_\beta \quad \text{and} \quad t_\beta \triangleright t_\alpha \quad \Rightarrow \quad \exists x \in t_\alpha, y \in t_\beta \text{ such that } x \mathcal{P} y \text{ and } y \mathcal{P} x.$$ 

$G$ is not a graph that satisfies antisymmetry under case 1.

Case 2 can be shown to result in a contradiction in parallel fashion.

Case 2: $\alpha$, $\beta$, and $\delta$ are nodes in $G'$ with phonological content

$$\alpha \mathcal{SP} \beta,$$

$$\beta \mathcal{SP} \delta,$$

$$\delta \mathcal{FD} \alpha.$$

1. $\alpha \mathcal{SP} \beta$ \hspace{2cm} \text{given}
2. $t_\alpha \triangleright t_\beta$ \hspace{2cm} \text{Full-dominance cousin-precedence}
3. $\beta \mathcal{SP} \delta$ \hspace{2cm} \text{given}
4. $t_\beta \triangleright t_\delta$ \hspace{2cm} \text{Full-dominance cousin-precedence}
5. $\delta \mathcal{FD} \alpha$ \hspace{2cm} \text{given}
6. $t_\alpha \subseteq t_\delta$ \hspace{2cm} \text{transitivity of FD and definition of } t_N
7. $t_\beta \triangleright t_\alpha$ \hspace{2cm} \text{ln 4, ln 6, and definition of subset}
8. $t_\alpha \neq \emptyset, t_\beta \neq \emptyset$ \hspace{2cm} \text{given}
9. $\exists x \in t_\alpha, y \in t_\beta \text{ such that } x \mathcal{P} y \text{ and } y \mathcal{P} x$ \hspace{2cm} \text{ln 7 and ln 8}

9 contradicts the premise that $G'$ satisfies antisymmetry.

Both Case 1 and Case 2 lead to a contradiction to the premise that $G'$ satisfies antisymmetry. Thus, our theorem is true.
6.2.4.3 Counterexample to the converse of Theorem 1

The converse of Theorem 1, however, is not true. This is demonstrated by the structures proposed for coordination discussed in subsections 5.3.1, 5.3.2, and 5.3.3. However, these counterexamples involve multi-dominated nodes that do not self-c-commanding relationship. Here we discuss another counterexample, one in which binary branching and the restriction to self-c-commanding multi-dominated nodes are both maintained: the graph in (30b), a case of remnant movement. In linguistics, “remnant” is used to refer to a constituent from which material has been extracted, and moving such a constituent with a missing element is referred to as remnant movement. (When we briefly considered an example of remnant movement in section 5.2.4.2, we presented both the multi-dominance representation side by side with a representation reflecting the traditional movement account of the same remnant movement, which is (30a) here. We will continue to present both representations together.) In this graph, \( \alpha \), \( \beta \), and \( \delta \) have phonological content. \( \alpha \) sister precedes \( \beta \), and \( \beta \) sister precedes \( \delta \). Crucially, as the result of remnant movement, the node \( \delta \) does not fully dominate \( \alpha \). Note that \( \alpha \) has a path that does not include the \( \delta \) node (R T \( \alpha \)). Thus, the graph in (30) satisfies the condition in the consequent of Theorem 1.

In spite of this, the antecedent of Theorem 1 is false. Note that \( \beta \) and \( \delta \) are daughters of S, and \( \beta \) sister precedes \( \delta \). This gives rise to the ordering \( b \) precedes \( u \). Node U sister precedes T, which gives rise to the ordering \( u \) precedes \( a \). By transitivity, \( b \) must also precedes \( a \). At the same time, \( \alpha \) and \( \beta \) are daughters of T. Since \( \alpha \) sister precedes \( \beta \), \( a \) must precede \( b \). If \( b \) must precede \( a \) and \( a \) must precede \( b \), then the graph
fails to be antisymmetric. We see that Theorem 1 provides only a necessary condition for antisymmetry, not a sufficient condition.

(30)  Counterexample to the converse of Theorem 1

<table>
<thead>
<tr>
<th>R</th>
<th>FD</th>
<th>b, u, a</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>FD</td>
<td>u</td>
</tr>
<tr>
<td>T</td>
<td>FD</td>
<td>a</td>
</tr>
<tr>
<td>β</td>
<td>FD</td>
<td>b</td>
</tr>
<tr>
<td>δ</td>
<td>FD</td>
<td>u</td>
</tr>
<tr>
<td>U</td>
<td>FD</td>
<td>u</td>
</tr>
<tr>
<td>α</td>
<td>FD</td>
<td>a</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S</th>
<th>SP</th>
<th>T</th>
<th>→</th>
<th>u</th>
<th>»</th>
<th>a</th>
</tr>
</thead>
<tbody>
<tr>
<td>β</td>
<td>SP</td>
<td>δ</td>
<td>→</td>
<td>b</td>
<td>»</td>
<td>u</td>
</tr>
<tr>
<td>U</td>
<td>SP</td>
<td>T</td>
<td>→</td>
<td>u</td>
<td>»</td>
<td>a</td>
</tr>
<tr>
<td>α</td>
<td>SP</td>
<td>β</td>
<td>→</td>
<td>a</td>
<td>»</td>
<td>b</td>
</tr>
</tbody>
</table>

On a final note, it is interesting that the graph above does not provide a counterexample to the converse of Conjecture 1. The least node fully dominating T is the root R, which also fully dominates U. The least node fully dominating β is also R, which, unlike S, does fully dominate α. Since both the consequent and antecedent of Conjecture 1 are false, the graph tells us nothing about the veracity of Conjecture 1.
6.2.5 Remnant movement and the antisymmetry requirement

A natural question that now arises is whether or not the ban on symmetry rules out all cases of remnant movement. As remnant movement is believed to be possible in natural language, ruling out such structures would be an undesirable outcome. In this subsection, we use remnant movement to refer to cases where a node $\beta$ is multi-dominated, and a node $T$ that dominates $\beta$ is also multi-dominated. We examine cases where the final configuration is as follows: $T$ c-commands all of $\beta$’s parents, but $\beta$ does not c-command all of $T$’s parents (roughly, $T$’s upper address is higher than $\beta$’s upper address). Let us consider extracted elements ($\beta$s) that do or do not maintain consistent sisterhood in combination with remnants ($T$s) that do or do not maintain consistent sisterhood.

(31) Four cases of remnant movement

<table>
<thead>
<tr>
<th>Case 1:</th>
<th>Case 2:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element maintains consistent sisterhood</td>
<td>Element maintains consistent sisterhood</td>
</tr>
<tr>
<td>Remnant maintains consistent sisterhood</td>
<td>Remnant does not maintain consistent sisterhood</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 3:</th>
<th>Case 4:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element does not maintain consistent sisterhood</td>
<td>Element does not maintain consistent sisterhood</td>
</tr>
<tr>
<td>Remnant maintains consistent sisterhood</td>
<td>Remnant does not maintain consistent sisterhood</td>
</tr>
</tbody>
</table>

In the section immediately above, the figure in (30) provides an example of case 3: $\beta$ is both a left sister and a right sister while $T$ is a right sister, never a left sister. Via transitivity, the graph in (30) violates antisymmetry. There are, however, graphs which fall under case 3 that do satisfy antisymmetry. One such example is given in (32)
Second example of case 3

Unsurprisingly, when we turn to an exemplar of case 1, the case that maintains consistent sisterhood of both contracted nodes, we find a linearizable graph. What is surprising, however, is that the predicted linearization diverges from that expected by strong/weak feature theory. That is, the linearization of the terminals in (33) is \( b a u \), not \( a b u \).
Below, we examine exemplars of the two remaining cases. Like case 3, cases 2 and 4 include some node with inconsistent sister status. In particular, in both cases, the remnant is both a left and right sister. Note that unlike case 3, these graphs fail to satisfy the necessary conditions for antisymmetry identified above. Consider (34). As above, T has a left sister, U, and a right sister, S. T’s right sister, S, fully dominates T’s left sister U. As a result, the terminals fully dominated by T and U are in conflicting orderings: $a$ precedes $u$, and $u$ precedes $a$. This is another subtle difference between the graphs in (30) and (34). Recall that it is via transitivity that symmetry arises in (30). The precedence statements for (30) that are extracted directly via full-dominance cousin precedence contain no contradictions among themselves.
(34) exemplar of case 2

Changing the sister precedence between $\alpha$ and $\beta$ does not eliminate the failure to meet the necessary conditions for antisymmetry that we observed above. Just as in (30), the graph in (35) includes node $T$ with a left sister, $U$, and a right sister, $S$. $T$’s right sister, $S$, fully dominates $T$’s left sister $U$, yielding the same conflicting orderings: $a$ precedes $u$, and $u$ precedes $a$. 
Exemplar of case 4

The first graph in this subsection, namely (32), showed that consistent sisterhood is not necessary, even in cases where a multidominated node’s sister remains uncontracted. The picture drawn from the examples of these four cases suggests that consistent sisterhood is sufficient for guaranteeing antisymmetry, and that consistent sisterhood is necessary for the remnant without a contracted sister. These examples have also shown how the ordering predicted by derived precedence of terminals in the proposed system diverges from the ordering predicted by a traditional remnant movement account. The next section provides a linguistic example where the proposed linearization process proves useful.
6.2.6 THE MANDARIN REFLEXIVE *ZIJI*: REMNANT MOVEMENT IN USE

The Mandarin Chinese reflexive *ziji* has been noted as an apparent exception to Principle A of the Binding Theory (Tang 1989, Huang and Tang 1991, Cole, Hermon, and Sung 1993). Not only does *ziji* exhibit long-distance binding, as shown in (37) where *ziji* may be bound by an NP outside its clause, *ziji* can also be bound by an apparent non-c-commanding element, as in (38).

(36) Principle A (as given in Haegeman and Guéron, 1999, based on Chomsky 1981)
Anaphors must be locally bound
An NP is bound when it is co-indexed with a c-commanding NP in an A position.
The local domain of an element is delimited by the first c-commanding subject.

(37) Zhangsan ren4wei2 [Lisi hai4-le ziji3]
Zhangsan think Lisi hurt-ASP self
‘Zhangsan thought that Lisi hurt himself.’ (Huang and Tang 1991)

(38) [Wo3 de jiao1ao4] hai4-le ziji3
I DE pride hurt-ASP self
‘My pride hurt myself.’ (Huang and Tang 1991)

---

6 A non-c-commanding, non-subject NP is also a possible antecedent for *ziji* when the NP is an experiencer argument of a ‘psychological’ verb:

[Ziji de xiao3hai2 mei2 de2 jiang3 de xiao1xi3] shi3 Lisi hen3 nan2guo4.
self DE child not get prize DE news make Lisi very sad
“The news that his own child did not get a prize made Lisi sad.” (Huang and Tang 1991)

Huang and Tang (1991) point out that psychological verbs often exhibit atypical behavior cross-linguistically.
A non-c-commanding antecedent is only available, however, when the NP is the “most prominent” animate subject NP within an inanimate NP that c-commands zi\textsuperscript{4}ji\textsuperscript{3} (Huang and Tang 1991). Huang and Tang (1991) provide example (39) to show how the availability of an animate, c-commanding NP blocks the coindexation of the non-c-commanding pronoun.\footnote{Cole and Hermon (1993) note that Korean exhibits the possibility of an apparent non-c-commanding antecedent under the same conditions. The contrast between the Mandarin examples in (38) and (39) in the text is mirrored in examples (i) and (ii) here.}

(39) [Wo3\textsubscript{i} de mei4mei4\textsubscript{j} hai4-le zi4ji3\textsuperscript{*ij}.]  
I DE\textsuperscript{8} little sister hurt-ASP self.  
‘My little sister hurt herself.’ (Huang and Tang 1991)

These non-c-commanding antecedents are not, however, absolute last resorts. A non-c-commanding NP is also able to block otherwise potential binders of zi\textsuperscript{4}ji\textsuperscript{3}. The ability of an embedded non-c-commanding binder to block long-distance binding by a matrix c-commanding NP, as shown in (40), indicates that non-c-commanding binders are syntactically determined rather than arising from pragmatic principles.

(40) Zhangsan\textsubscript{i} shuo1 [wo3\textsubscript{j} de jiao1ao4 hai4-le zi4ji3\textsuperscript{*ij}]  
Zhangsan say I DE pride hurt-ASP self  
“Zhangsan said that myi pride hurt myself\textsuperscript{*ij}.”

\footnote{Cole and Hermon (1993) note that Korean exhibits the possibility of an apparent non-c-commanding antecedent under the same conditions. The contrast between the Mandarin examples in (38) and (39) in the text is mirrored in examples (i) and (ii) here.}

(i) John\textsubscript{n}-uy camansim-ij casin\textsubscript{n-o}\-ul mangchi-ess-ta  
John-Gen pride-Nom self-Acc ruin-Past-Decl  
“John’s pride ruined himself.”

(ii) John\textsubscript{n}-uy apeci\textsubscript{n}-ka casin\textsubscript{n-o}\-ul mangchi-ess-ta  
John-Gen father-Nom self-Acc ruin-Past-Decl  
“John’s father ruined himself.”

\footnote{De is the possessor marker in Mandarin.}
One rational approach at this point would be to posit that the antecedent actually does c-command zi4ji3 as the result of movement, and it is from this c-commanding position that binding is possible. The consequence of this hypothesis, however, is that the moved antecedent would no longer form a constituent with the rest of the subject. This predicts that additional material, such as a sentential adverb, might be able to appear between the antecedent and the subject it has moved out of, but the prediction is not borne out. The examples below show that jing1tian1 ‘yesterday’ can appear before the subject or after the subject, but it cannot appear between ta1 ‘his,’ the antecedent, and de jiao1ao4 ‘DE pride,’ the other elements of the subject.

(41) Zuo2tian1, ta1 de huai4 xing2wei2 mei2you3 huai4 jie2guo3, dan4shi4 . . .
yesterday he DE bad behavior not have bad consequence, but . . .

a. . . . * ta1, jing1tian1 [e, de jiao1ao4] zhong1yu2 hai4-le zi4ji3.

. . . he today (e) DE pride eventually hurt-ASP self

b. . . . jing1tian1, [ta1 de jiao1ao4] zhong1yu2 hai4-le zi4ji3.

. . . today he DE pride eventually hurt-ASP self

c. . . . [ta1 de jiao1ao4] jing1tian1 zhong1yu2 hai4-le zi4ji3.

. . . he DE pride today eventually hurt-ASP self

“Yesterday, his bad behavior had no consequences, but today, his pride ultimately hurt himself.”

9 The same string as (41a) is possible with a different reading, when jing1tian1 ‘today’ modifies jiao1ao4 ‘pride’:

(i) Zuo2tian1, ta1 de huai4 xing2wei2 mei2you3 huai4 jie2guo3, dan4shi4 . . .
yesterday he DE bad behavior not have bad consequence, but . . .

. . . . [ta1, [jing1tian1 de jiao1ao4]] zhong1yu2 hai4-le zi4ji3.

. . . he today DE pride eventually hurt-le self

“Yesterday, his bad behavior had no consequences, but the pride he has today ultimately hurt himself.”

More literally, the subject can be glossed as [his [today’s pride]].

152
Instead of pursuing an account in which the antecedent does c-command the reflexive, Tang (1989) introduces the notion of *sub-command* to capture the structural relationship to *zi4ji3* that is required of a non-c-commanding antecedent.

(42) \[ B \text{ sub-commands } A \text{ iff} \]
\[ \text{B is contained in an NP that c-commands A or that sub-commands A, and} \]
\[ \text{Any argument containing B is in subject position.} \]

(Tang 1989)

Huang and Tang attribute the inclusion of certain non-c-commanding NPs as potential binders to a peculiarity of the *zi4ji3* reflexive: *zi4ji3* relaxes the c-command requirement for binders, allowing coindexation between elements in a sub-command relationship as well.

In contrast, Cole, Hermon, and Sung (1993) attribute the possibility of a sub-commanding antecedent to a lexical peculiarity of inanimates. In their account, a c-command requirement is maintained, but it is not the antecedent itself that must c-command *zi4ji3*. Rather, an element with the [antecede i] feature needs to c-command i. Typically, as in (35), a head’s feature values take precedence as features percolate up, blocking the co-indexation of *wo3* and *zi4ji3*. Inanimate elements, however, have no feature value for [antecede]. When a head is inanimate, as in (38), this allows the [antecede] feature of the sub-commander to percolate all the way up to a position c-commanding *zi4ji3*.\(^\text{10}\)

\(^{10}\) Cole, Hermon, and Sung (1993) also show how their account can be extended to Korean honorific agreement. When the head of the subject is inanimate, an honorific on the verb shows agreement with a sub-commanding element.
The linearization proposal under discussion here, however, reopens the door to analyzing these apparent deviations from principle A as c-commanding antecedents, thereby avoiding reference to notions like subcommand or an [antecede] feature.

Suppose that the NP antecedent of zi4ji3 moves from a non-c-commanding position into a c-commanding position, followed by movement of the remnant subject. The proposed structure is given in (43) (in both traditional and multi-dominance representations). We shall see that the linearization process both derives the correct word order and disallows an adverbial to intervene between the antecedent of zi4ji3 and the rest of the subject when remnant movement occurs.

(43) Structure for \[Wo3, de jiao1ao4 \ hais-le \ zi4ji3i.\]
I DE pride hurt-ASP self.
‘My pride hurt myself.’
In example (35) above, we observed a divergence between structural position and pronounced position that arises when the subject of a remnant is moved prior to movement of the remnant. Here, we exploit that divergence to derive the correct linearization. The terminals fully dominated by each node are given in the table below.

Note that \( wo_3 \) is not fully dominated by either of its preterminal’s parents. Its three paths to the root are: \((wo_3, DP2, XP DP1, VP_{11}, VP_1, VP_{root})\), \((wo_3, DP2, XP DP1, VP_{root})\), and \((wo_3, VP_1, VP_{root})\).

<table>
<thead>
<tr>
<th></th>
<th>FD</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DP1</td>
<td>DP</td>
<td>de, jiao1ao4</td>
</tr>
<tr>
<td>VP1</td>
<td>DP</td>
<td>hai4-le, zi4ji3</td>
</tr>
<tr>
<td>XP</td>
<td>DP</td>
<td>de</td>
</tr>
<tr>
<td>NP</td>
<td>DP</td>
<td>jiao1ao4</td>
</tr>
<tr>
<td>DP2</td>
<td>DP</td>
<td>wo3</td>
</tr>
<tr>
<td>X</td>
<td>DP</td>
<td>de</td>
</tr>
<tr>
<td>VP_{11}</td>
<td>DP</td>
<td>hai4-le, zi4ji3</td>
</tr>
<tr>
<td>VP_{111}</td>
<td>DP</td>
<td>hai4-le, zi4ji3</td>
</tr>
<tr>
<td>V</td>
<td>DP</td>
<td>hai4-le</td>
</tr>
<tr>
<td>DP3</td>
<td>DP</td>
<td>zi4ji3</td>
</tr>
</tbody>
</table>

Full-dominance cousin precedence yield the following orderings. The sister precedence relations of DP2 tell us that \( wo_3 \) precedes \( de \) and \( hai4-le \) and \( zi4ji3 \).

<table>
<thead>
<tr>
<th></th>
<th>SP</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DP2</td>
<td>SP</td>
<td>X ( \rightarrow ) wo3 ( \Rightarrow ) de</td>
</tr>
<tr>
<td>DP2</td>
<td>SP</td>
<td>VP_{11} ( \rightarrow ) wo3 ( \Rightarrow ) hai4-le, zi4ji3</td>
</tr>
</tbody>
</table>

The sister precedence pair (XP, NP) tells us that \( de \) precedes jiao1ao4. So far, we know that \( wo_3 \) precedes \( de \) which precedes jiao1ao4, and that \( wo_3 \) precedes hai4-le and zi4ji3.

<table>
<thead>
<tr>
<th></th>
<th>SP</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>XP</td>
<td>SP</td>
<td>NP ( \rightarrow ) de ( \Rightarrow ) jiao1ao4</td>
</tr>
</tbody>
</table>
DP1 fully dominates \textit{jiao1ao4}, and DP1’s sister precedence pairs order \textit{jiao1ao4} with respect to \textit{hai4-le} and \textit{zi4ji3}. Now, we conclude that \textit{wo3} precedes \textit{de} which precedes \textit{jiao1ao4} which precedes \textit{hai4-le} and \textit{zi4ji3}.

\begin{center}
\begin{tabular}{|c|c|c|}
\hline
\textbf{DP1} & \textbf{S\text{\textsuperscript{\text{\scriptsize{\text{}}}}}} & \textbf{VP\text{\textsubscript{1}}} \rightarrow \ de, jiao1ao4 » hai4-le, zi4ji3 \\
\hline
\textbf{DP1} & \textbf{S\text{\textsuperscript{\text{\scriptsize{\text{}}}}}} & \textbf{VP\text{\textsubscript{111}}} \rightarrow \ de, jiao1ao4 » hai4-le, zi4ji3 \\
\hline
\end{tabular}
\end{center}

The ordering between \textit{hai4-le} and \textit{zi4ji3} is established by the remaining sister precedence pair.

\begin{center}
\begin{tabular}{|c|c|c|}
\hline
\textbf{V} & \textbf{S\text{\textsuperscript{\text{\scriptsize{\text{}}}}}} & \textbf{DP3} \rightarrow \ hai4-le » zi4ji3 \\
\hline
\end{tabular}
\end{center}

As this exhausts the list of sister precedence pairs, this leads us to the linearization \textit{wo3 de jiao1ao4 hai4-le zi4ji3}, not \textit{de jiao1ao4 wo3 hai4-le zi4ji3} as the structure might suggest.

Having examined linearization without adverbials, we now turn to cases where adverbials are present. We first explore cases where the structure indicates that the antecedent c-commanded \textit{zi4ji3} and the subject has moved above (i.e. to a position c-commanding) a sentential adverb. The three configurations that meet these requirements are given in (44) – (46). Of particular interest is (44), where the structure indicates that an adverb separates the remnant from the antecedent. All three have the surface linearization reported in (41c). (Note that usually, the use of the term \textit{remnant movement} indicates not only that the remnant moved, but, in a system allowing only leftward movement into a c-commanding position, also that the movement was to a position to the left of the extracted item. Thus, movements like those in (46), (48), and (49) are not classified as remnant movement. I will use the term \textit{remnant} more loosely, however, to
mean a constituent which includes a multi-dominated element. Here, *remnant* does not necessarily mean that the constituent has undergone remnant movement.

(44) structure: Rem Adv DP
linearization: DP Rem Adv
attested case: *Ta1 de jiao1ao4 jing1tian1 zhong1yu2 hai4-le zi4ji3.*

a.

b.
Note that VP1 and VP11 do not FD ta1. DP2 has a path that does not include either of these VP nodes (VProot ID DP1 ID X ID DP2)
(45) structure: Rem DP Adv
linearization: DP Rem Adv
attested case: *Ta1 de jiao1ao4 jing1tian1 zhong1yu2 hai4-le zi4ji3.

```
| DP1       | FD | de, jiao1ao4 |
| VP1       | FD | jing1tian1, zhong1yu2, hai4-le, zi4ji3 |
| XP        | FD | de          |
| NP        | FD | jiao1ao4    |
| DP2       | FD | ta1         |
| X         | FD | de          |
| VP11      | FD | jing1tian1, zhong1yu2, hai4-le, zi4ji3 |
| Adv1      | FD | jing1tian1  |
| VP111     | FD | zhong1yu2, hai4-le, zi4ji3 |
| VP1111    | FD | zhong1yu2, hai4-le, zi4ji3 |
| Adv2      | FD | zhong1yu2   |
| VP11111   | FD | hai4-le, zi4ji3 |
| V         | FD | hai4-le     |
| DP3       | FD | zi4ji3      |
```
Note that, as above, VP<sub>1</sub> and VP<sub>11</sub> do not FD *ta1* in these graphs either. DP2 has a path that does not include either of these VP nodes (VP<sub>root</sub> ID DP1 ID X ID DP2)

(46) structure: NP Rem Adv
linearization: NP Rem Adv
attested case: *Ta1 de jiao1ao4 jing1tian1 zhong1yu2 hai4-le zi4ji3.*

```
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DP1</td>
<td>S&lt;sup&gt;∅&lt;/sup&gt; VP&lt;sub&gt;1&lt;/sub&gt;</td>
<td>→ de, jiao1ao4 » jing1tian1, zhong1yu2, hai4-le, zi4ji3</td>
</tr>
<tr>
<td>XP</td>
<td>S&lt;sup&gt;∅&lt;/sup&gt; NP</td>
<td>→ de » jiao1ao4</td>
</tr>
<tr>
<td>DP2</td>
<td>S&lt;sup&gt;∅&lt;/sup&gt; X</td>
<td>→ ta1 » de</td>
</tr>
<tr>
<td>Adv1</td>
<td>S&lt;sup&gt;∅&lt;/sup&gt; VP&lt;sub&gt;11&lt;/sub&gt;</td>
<td>→ jing1tian1 » zhong1yu2, hai4-le, zi4ji3</td>
</tr>
<tr>
<td>DP1</td>
<td>S&lt;sup&gt;∅&lt;/sup&gt; VP&lt;sub&gt;111&lt;/sub&gt;</td>
<td>→ de, jiao1ao4 » zhong1yu2, hai4-le, zi4ji3</td>
</tr>
<tr>
<td>Adv2</td>
<td>S&lt;sup&gt;∅&lt;/sup&gt; VP&lt;sub&gt;1111&lt;/sub&gt;</td>
<td>→ zhong1yu2 » hai4-le, zi4ji3</td>
</tr>
<tr>
<td>V</td>
<td>S&lt;sup&gt;∅&lt;/sup&gt; DP3</td>
<td>→ hai4-le » zi4ji3</td>
</tr>
</tbody>
</table>
```
Regardless of the position from which the antecedent c-commands the reflexive, an attested surface order is achieved.

When the subject does not move above the adverb, the configurations under which an apparently sub-commanding antecedent will also c-command zi4ji3 are given in (47)-(49). All six possible orderings of \{Adv, NP, Rem\} are represented in (44)-(49). The first two assume that the remnant-subject has indeed moved to a position below the adverb.\(^\text{11}\) The last configuration does not assume movement of the remnant. So long as the remnant remained below the adverb, moving the remnant would not yield a different linearization.

\[(47)\]  
structure: Adv Rem NP  
linearization: Adv NP Rem  
attested case: Jing\text{\textquotesingle}tian\text{\textquotesingle} ta1 de jiao1ao4 zhong1yu2 hai4-le zi4ji3.

\(^\text{11}\) Again, as mentioned above, even though we use the term \textit{remnant} to refer to the subject with a multi-dominated element, movements like that in (48) and (49) are not typically considered remnant movement.
Adv1 $\text{FD} $ jing1tian1
VP1 $\text{FD} $ ta1, de, jiao1ao4, zhong1yu2, hai4-le, zi4ji3
DP1 $\text{FD} $ de, jiao1ao4
XP $\text{FD} $ de
NP $\text{FD} $ jiao1ao4
DP2 $\text{FD} $ ta1
X $\text{FD} $ de
VP11 $\text{FD} $ zhong1yu2, hai4-le, zi4ji3
VP111 $\text{FD} $ zhong1yu2, hai4-le, zi4ji3
VP1111 $\text{FD} $ zhong1yu2, hai4-le, zi4ji3
Adv2 $\text{FD} $ zhong1yu2
VP11111 $\text{FD} $ hai4-le, zi4ji3
V $\text{FD} $ hai4-le
DP3 $\text{FD} $ zi4ji3

<table>
<thead>
<tr>
<th>Adv1</th>
<th>SΦ</th>
<th>VP1</th>
<th>→</th>
<th>jing1tian</th>
<th>→</th>
<th>ta1, de, jiao1ao4, zhong1yu2, hai4-le, zi4ji3</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP1</td>
<td>SΦ</td>
<td>VP11</td>
<td>→</td>
<td>de, jiao1ao4</td>
<td>→</td>
<td>zhong1yu2, hai4-le, zi4ji3</td>
</tr>
<tr>
<td>XP</td>
<td>SΦ</td>
<td>NP</td>
<td>→</td>
<td>de</td>
<td>→</td>
<td>jiao1ao4</td>
</tr>
<tr>
<td>DP2</td>
<td>SΦ</td>
<td>X</td>
<td>→</td>
<td>ta1</td>
<td>→</td>
<td>de</td>
</tr>
<tr>
<td>DP2</td>
<td>SΦ</td>
<td>VP11</td>
<td>→</td>
<td>ta1</td>
<td>→</td>
<td>zhong1yu2, hai4-le, zi4ji3</td>
</tr>
<tr>
<td>DP1</td>
<td>SΦ</td>
<td>VP111</td>
<td>→</td>
<td>de, jiao1ao4</td>
<td>→</td>
<td>zhong1yu2, hai4-le, zi4ji3</td>
</tr>
<tr>
<td>Adv2</td>
<td>SΦ</td>
<td>VP1111</td>
<td>→</td>
<td>zhong1yu2</td>
<td>→</td>
<td>hai4-le, zi4ji3</td>
</tr>
<tr>
<td>V</td>
<td>SΦ</td>
<td>DP3</td>
<td>→</td>
<td>hai4-le</td>
<td>→</td>
<td>zi4ji3</td>
</tr>
</tbody>
</table>
(48) structure: Adv NP Rem
linearization: Adv NP Rem
attested case: Jing1tian1 ta1 de jiao1ao4 zhong1yu2 hai4-le zi4ji3.

<table>
<thead>
<tr>
<th>Term</th>
<th>FD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adv1</td>
<td>Jing1tian1</td>
</tr>
<tr>
<td>VP1</td>
<td>ta1, de, jiao1ao4, zhong1yu2, hai4-le, zi4ji3</td>
</tr>
<tr>
<td>DP2</td>
<td>ta1</td>
</tr>
<tr>
<td>VP11</td>
<td>de, jiao1ao4, zhong1yu2, hai4-le, zi4ji3</td>
</tr>
<tr>
<td>DP1</td>
<td>de, jiao1ao4</td>
</tr>
<tr>
<td>VP111</td>
<td>zhong1yu2, hai4-le, zi4ji3</td>
</tr>
<tr>
<td>XP</td>
<td>de</td>
</tr>
<tr>
<td>NP</td>
<td>jiao1ao4</td>
</tr>
<tr>
<td>X</td>
<td>de</td>
</tr>
<tr>
<td>VP1111</td>
<td>zhong1yu2, hai4-le, zi4ji3</td>
</tr>
<tr>
<td>Adv2</td>
<td>zhong1yu2</td>
</tr>
<tr>
<td>VP11111</td>
<td>hai4-le, zi4ji3</td>
</tr>
<tr>
<td>V</td>
<td>hai4-le</td>
</tr>
<tr>
<td>DP3</td>
<td>zi4ji3</td>
</tr>
<tr>
<td></td>
<td>SP</td>
</tr>
<tr>
<td>---</td>
<td>----</td>
</tr>
</tbody>
</table>
| Adv1 | S\(^p\) | VP\(_1\)  | → jing1tian1  » ta1, de, jiao1ao4, zhong1yu2, hai4-le, zi4ji3  
| DP2 | S\(^p\) | VP\(_{11}\)  | → ta1  » de, jiao1ao4, zhong1yu2, hai4-le, zi4ji3  
| DP1 | S\(^p\) | VP\(_{111}\)  | → de, jiao1ao4  » zhong1yu2, hai4-le, zi4ji3  
| XP | S\(^p\) | NP  | → de  » jiao1ao4  
| DP2 | S\(^p\) | X  | → ta1  » de  
| DP1 | S\(^p\) | VP\(_{1111}\)  | → de, jiao1ao4  » zhong1yu2, hai4-le, zi4ji3  
| Adv2 | S\(^p\) | VP\(_{11111}\)  | → zhong1yu2  » hai4-le, zi4ji3  
| V | S\(^p\) | DP3  | → hai4-le  » zi4ji3  

(49)  
structure: NP Adv Rem  
linearization: NP Adv Rem  
unattested case: * Ta1 jing1tian1 de jiao1ao4 zhong1yu2 hai4-le zi4ji3.

a. *  

![Diagram a](attachment:image1.png)

b. *  

![Diagram b](attachment:image2.png)
The unacceptability of the pronunciation of the last possibility, in which the antecedent moves above the adverb, suggests that either the node available to contract with NP is below the Adv (at least in cases where there is no remnant movement), that the node available to contract with the remnant is above the Adv, or that these special apparent sub-command cases are linked to true remnant movement, where the remnant c-commands a parent of the extracted element.

Huang and Tang (1991) further note that, in general, zi4ji3 can only be bound by a subject, not to an object. As expected, this prohibition includes sub-commanding NPs as well: objects of sentential subjects cannot antecede reflexives.
(50) * [Jiang³ Zhangsan de gu4shi4] hai4-le zi4ji3.
    talk-about Zhangsan DE story harm-ASP self
    “The story about Zhangsan harmed himself.”

This observation can be related to the class of linearizable graphs in an interesting way. Within the sentential subject, the object (Zhangsan in (50)) must be a right sister, be a left sister, or have no sister. In all three cases, when both the sentential subject and the object of the subject is multi-dominated, the unusual configuration renders the object of the subject not fully dominated by either of its parents. This averts antisymmetry violations, but, interestingly, gives rise to totality violations. For example, in (51), no ordering would be possible between \(x\), the sentential subject’s subject, and the object. We can derive that \(obj\) precedes \(z\) and \(q\), and that \(x\) precedes \(u\), \(z\), and \(q\), but no ordering is derived between \(x\) and \(obj\) or \(u\) and \(obj\). In (52), there is again no ordering derived between \(x\) and \(obj\) or \(u\) and \(obj\).

(51) Disallowed remnant movement with extracted object

\[
\begin{align*}
\text{(a) } & \ast \\
\text{(b) } & \ast
\end{align*}
\]
6.3 THE TOTALITY REQUIREMENT

6.3.1 OVERVIEW
Above, we examined the consequences of the claim that contradictory (i.e., asymmetric) ordering information is not tolerated by the linearization system. We now turn to examine the claim that totality violations cause the system to rule a structure as ill-formed. First, in 6.3.2 we explore the consequences for the potential analyses of coordination. Next, we move towards a characterization that is more formal in nature (though not completely formalized) in 6.3.3.
6.3.2 THE NEED FOR NON-SENTENTIAL COORDINATION

Consider the possible structures for a construction with conjoined subjects. One possible analysis is sentential coordination, with a multidominated verb and a multidominated object.

(53) A potential structure for *John and Mary ate cookies.*

The linearizability requirement, however, is not met in this graph.\(^{12}\) Note that since only the root BP fully dominates *ate* and *cookies*, *John, Mary,* and *and* are all unordered with

\(^{12}\) Recall, however, that in Chapter 3, I posited that contraction of nodes is only possible when at most one of those nodes dominates a terminal, as in the case of auxiliary inversion. In the MD-TAG system, each verb anchors an elementary tree that includes slots for combination with its arguments. We have not allowed contraction of nodes that immediately dominate two lexical anchors as this would result in a node dominating two terminals. In this case, *V* would dominate two instances of *ate*, as in (i)-(iii), which correspond to (53), (54), and (57), respectively. To generate the structures in (53), (54), and (57), it is necessary to relax or eliminate this restriction as well as to propose a mechanism that would keep one terminal from being pronounced.

Also see footnote 15 for Sarkar and Joshi’s (1996) approach to gapping.
respect to *ate* and *cookies*. To maintain a linearizable sentential coordination analysis for *John and Mary ate cookies* would require a structure as in (54).

\[(54)\] Second potential structure for *John and Mary ate cookies*.

Now that $V'$ fully dominates *ate* and *cookies*, these terminals can be ordered with respect to the other terminals.\(^{13}\)

Alternatively, a structure with conjoined subject DPs has no node multi-dominated nodes and its linearization is straightforward.

\[(55)\] A third potential structure for *John and Mary ate cookies*.

\(^{13}\) Consider, however, the same sentence in the present tense:

(i) *John and Mary eat cookies.*

If this is derived from coordinating *John eats cookies* and *Mary eats cookies*, then some additional mechanism is needed for determining the (plural) agreement on the verb.
As has been pointed out by Gleitman (1965), examples such as (52) indicate that not all cases of coordination can be reduced to sentential coordination. Such cases suggest the need for structures like that in (55).

(56)  *John and Mary danced together.* ≠ *John danced together and Mary danced together.*

Interestingly, positing a multi-dominated V' level node is still insufficient to provide a linearizable analysis for a construction with coordinated objects. This is the structure generated in MD-TAG by node contraction of the subject DP, V', and V nodes belonging to two elementary trees.

(57)  A potential structure for *Joe eats cookies and ice cream.*

![Diagram](image)

In addition to appearing to incorrectly indicate that *eats* takes two complements\(^{14}\), this structure is not linearizable. Neither VP1 nor VP2 fully dominate anything. This leaves

\(^{14}\) Perhaps the machinery needed to ensure only one instance of *eats* is pronounced could be constructed in such a way as to allow/require each instance of *eats* to take a complement.
and unordered with respect to all other terminals and leaves cookies and ice cream unordered with respect to each other.

Returning to an analysis that does not involve V' will not solve the problem either. In the structure in which the subject and verb is contracted, the subject and verb fail to be ordered with respect to one another. As should be familiar now, only the root BP fully dominates the multi-dominated terminals. Since the nodes that fully dominate Joe and eats (their preterminals and the root) are not in any sister precedence relation from which their relative ordering could arise, Joe and eats are thus unordered with respect to one another.

(58) A second potential structure for Joe eats cookies and ice cream.

As above, these totality violations can be avoided in a structure with conjoined object DPs. This structure involves no multi-dominated nodes and has no linearization violations.
The ban against totality violations rules out (53) as a potential analysis for sentences with coordinated subjects while keeping (54) and (55) as possible analyses. Unless the proposed generation system is revised to allow two nodes that both dominating terminals to contract, adopting the linearization requirement means rejecting the notion that cases of subject coordination can be reduced to cases of sentential coordination. Even this revision would not prove enough to provide a sentential coordination analysis of coordinated objects. (57) and (58) are ruled out as a potential analyses for sentences with coordinated objects, leaving (59) as the analysis for such sentences.

6.3.3 PROPERTIES OF TOTAL GRAPHS

This subsection summarizes various endeavors to formalize the conditions under which syntactic graphs satisfy totality. We first show how two conditions that are linguistically inspired, a c-command restriction and an edge restriction, are ultimately neither necessary nor sufficient for totality. Like the section above on antisymmetry, we finally focus our attention on a subset of syntactic graphs. We suggest that in the absence of remnant
movement, restricting multi-dominated nodes to be self-c-commanding is sufficient for ensuring totality in binary branching graphs.

6.3.3.1 Insufficient restrictions: binary branching and self-c-commanding multi-dominated nodes

As in the discussion of properties of graphs satisfying antisymmetry, we first restrict ourselves to binary branching graphs in which multi-dominated nodes self-c-command. We have already encountered linearizable graphs that do not conform to the c-command condition in the section on coordination. We can therefore conclude that the c-command condition is not necessary for totality. The graph below shows that the c-command condition is also not sufficient to guarantee totality. As in the preceding section, the traditional representation of remnant movement is given in (60a) and the graph representation is given in (60b).
Because of the remnant movement, X and Y are only fully dominated by the root R, resulting in several nodes that do not fully dominate any terminals. The only ordering information available is that $x$ precedes both $v$ and $y$. The transitive closure of this information does not order $v$ and $y$ with respect to one another.

### 6.3.3.2 Insufficient restrictions: binary branching and the edge condition

From the linearizable graphs we have posited for questions, coordinate structures, and the sub-commanding antecedents of the Mandarin reflexive $zi4ji3$, it appears that some notion of edge may be useful in linearization. The multi-dominance analyses for shared subjects (coordinated VPs), shared objects (Right Node raising), shared subjects and objects (coordinated Vs), A'-movement, and topicalization all pronounce the multi-dominated element at an edge of a sentence. When more than one element is multi-dominated, the structure seems vulnerable to totality violations unless each of the elements is at an edge (as with shared subjects and objects). Recall that in our discussion of sentential versus constituent coordination, a graph in which a verb and object were both contracted separately failed to be total. The graph in (53) is repeated here in (61a).
In (54), repeated here as (61b), we noted an alternative structure that was linearizable.\(^\text{15}\)

Whereas the preterminal for *ate* is not at an edge in (61a), the node \(V'\) is at the right edge of the construction in (61b).

\[(61)\]

\[
\begin{array}{ll}
\text{a.} & * \\
\text{b.} & \\
\end{array}
\]

\[
\begin{array}{ll}
\text{VP1} & \text{BP} \\
\text{VP2} & \\
\text{DP} & \text{V} \\
\text{John} & \text{and} \\
\end{array}
\]

\[
\begin{array}{ll}
\text{VP1} & \text{BP} \\
\text{VP2} & \\
\text{DP} & \text{V} \\
\text{Mary} & \text{ate} \\
\end{array}
\]

The notion of an edge (of some domain) also has linguistic relevance. Traditionally, elements that are focused or topicalized are analyzed as moving the periphery of a clause. The association of increased discourse salience with edge positions may be due to increased acoustic salience of the beginning and end of linguistic units.

The notion of edge and the notion of an edge condition must be made more precise. Certainly, we do not want a condition that allows multi-domination only when one of the node’s addresses is at the edge of an entire clause.

\[(62)\] V to I movement

\[
\text{I } \text{know}_i \text{ not } \_	ext{ } \_j \text{ where to hide my head.}
\]

(Early Modern English, Trinculo in *The Tempest*)

\(^\text{15}\) Recall that we ultimately argued that sentences that appear to have coordinated subjects really do involve structurally coordinated subjects, not coordinated sentences. If we commit to the generation system in Part One, our assumption from chapter 3 that verbs anchor elementary trees in combination with the condition that bars node contraction of two nodes with phonological material prohibits contraction of the two V nodes that immediately dominate the verbs that anchor the trees.
What we mean by the \textit{edge condition}, then, is that a multi-dominated node $\alpha$ must have at least one path to the LNFD$_{\alpha}$ whose nodes are all consistent in sisterhood. That is, if node $N_{i-1}$ is the right daughter of $N_i$, then node $N_i$ is the right daughter of $N_{i+1}$. For example, in (61b), the LNFD$_{V'}$ is the root. $V'$ has two paths to the root: $(V', \ VP1, \ root)$ and $(V', \ VP2, \ BP, \ root)$. The latter path consists entirely of right sisters.

Unfortunately, the edge condition is neither necessary nor sufficient for totality. We have already seen that some cases of remnant movement, such as (32b), are linearizable without satisfying this edge condition. Below, we introduce a counterexample to the speculation that the edge condition is sufficient to guarantee totality.

(64) *
Z is the left daughter of S, which is the left daughter of the root R (which is the LNFDz).

U is the right daughter of T, which is the right daughter of R (which is the LNFDu).

Consequently, the edge condition is satisfied. However, the only ordering information available is that z and x both precede u. The transitive closure of this information does not order z and x with respect to one another.

6.3.3.3 Sufficient restrictions: Binary branching, self-c-commanding multi-dominated nodes, and the edge condition

Note that node X in the graph in (60) fails to satisfy the edge condition, while the graph in (64) fails to satisfy the c-command condition. It is natural to next consider graphs that not only abide by the self-c-command condition on multi-dominated nodes but abide by the edge condition as well. This can be achieved by prohibiting remnant movement.

Without remnant movement, the “upper” parent, the parent with a shorter path to the root, of a contracted node α is the same as the LNFDα. As a daughter of the LNFDα, α occupies an edge position. Though this set of graphs is quite restrictive, the set still encompasses the majority of structural analyses attributed to core cases. Interestingly,
proving that all members of this set satisfy totality is still a challenge. The formalization of the conjecture is below, followed by a proof outline.

The antecedent is itself an if-then statement. The first half of the antecedent’s then-statement, \(((\alpha \mathcal{D} \beta) \lor (\beta \mathcal{D} \alpha))\), is the requirement that multi-dominated nodes self-c-command. The second half, \((\forall \, \text{N such that } (\gamma \mathcal{D} \text{N}), (\gamma \mathcal{FD} \text{N}))\), prohibits remnant movement. That is, if \(\gamma\) has two parents (i.e. has been “moved”), then all nodes dominated by \(\gamma\) are fully dominated by \(\gamma\). The consequent states that any two terminals that are not identical must be in some precedence relation.

(65) Conjecture 2:
if \(\forall \, \alpha, \beta, \gamma \in \text{graph } G\),

( if \(\alpha \neq \beta\)

\(\alpha \mathcal{ID} \gamma\), and

\(\beta \mathcal{ID} \gamma\),

then \(((\alpha \mathcal{D} \beta) \lor (\beta \mathcal{D} \alpha))\)

and

\((\forall \, \text{N such that } (\gamma \mathcal{D} \text{N}), (\gamma \mathcal{FD} \text{N}) )\)

then: \(\forall \, x, y \in G\), where \((x \neq y)\)

\((x \mathcal{Py}) \text{ or } (y \mathcal{Px})\)

In the generation section, we attributed generation of syntactic graphs in natural language speakers to the MD-TAG system. However, as formal objects, we can partition syntactic graphs by tree depth and generate the set of trees of depth \(n+1\) from the set of trees of depth \(n\). When the graphs i) are binary branching and ii) limit node-contraction to nodes in a c-command relationship, the possible extensions parallel the
derivational operations *merge* and *remerge* of the Minimalist Program (Chomsky 2001/2004). A graph of depth $n$ can be merged with another graph of depth $\leq n$ to create a graph of depth $n+1$. The merge depicted below merges the graph of depth $\leq n$ on the left of the graph of depth $n$, but it is equally possible to merge on the right.

(66) Merging $R'$ with $R''$

\[ R' \begin{array}{c} \leq n \text{ merges with } R'' \end{array} \begin{array}{c} \text{n to yield} \end{array} \begin{array}{c} R \end{array} \begin{array}{c} n + 1 \end{array} \]

Alternatively, some node within the graph of depth $n$ can be remerged to yield a graph of depth $n+1$, again either to the left or to the right of the graph of depth $n$.

(67) Remerging $R'$ with $R''$

\[ R \begin{array}{c} \text{n to yield} \end{array} \begin{array}{c} R' \end{array} \begin{array}{c} \text{n} \end{array} \begin{array}{c} <n \end{array} \begin{array}{c} \text{n + 1} \end{array} \]

The overarching proof strategy is via induction: we first show that the conjecture holds for the smallest graph, then argue that each of the possible ways to extend the depth of any graph yields a graph in which the conjecture continues to hold.
Outline of a proof by induction on tree depth (Here, what is meant by tree depth is the
length of the longest path length from any preterminal to the root):

Our base case is graph of depth 1.

(68)

The graph has no contracted nodes, which means the graph is a tree.

Trees are total, since all terminals can be directly ordered via full-dominance
cousin-precedence.

Conjecture 2 is true when $n=1$.

Next, we must show that if we know conjecture 2 is true for graphs of depth $\leq n$, then
conjecture 2 is also true for graphs of depth $n+1$. Let us consider in turn the possible
ways to “grow” a graph of depth $n+1$.

Case 1: Merge a graph of depth $\leq n$ with a graph of depth $n$.

Let the root of the resulting graph $G$ be called $R$.

Let the left daughter of $R$ be called $R'$. This node was the root of a graph of depth
$\leq n$ prior to the merge. Let the tree rooted in $R'$ be called $G'$.

Let the right daughter of $R$ be called $R''$. This node was also the root of a graph of
depth $\leq n$ prior to the merge. Let the tree rooted in $R''$ be called $G''$. 
There is no contraction involved in the merging of G' and G''. Therefore, none of the full dominance relations in G' or G'' are absent from G, the new graph of depth n+1.

Since the full dominance relations of G' and G'' are undisturbed and both subtrees are of depth \( \leq n \), we can conclude that all the terminals in G' are ordered with respect to one another and all the terminals in G'' are ordered with respect to one another. Thus, if \( x \) and \( y \) are both in G', they will be ordered with respect to one another, and if \( x \) and \( y \) are both in G'', they will be ordered with respect to one another.

G includes a new sister precedence pair, R' sister precedes R''. Via full dominance cousin precedence, all the terminals in G' precede all the terminals of G''. Thus, if \( x \) is in G' and \( y \) is in G'', \( x \) will precede \( y \), and if \( x \) is in G'' and \( y \) is in G', \( x \) will follow \( y \).

Conjecture 2 is true in case 1.

Case 2: Remerge a node from within a graph of depth n.

Let the root of the resulting graph G be called R.

Let the left daughter of R be called R'. This node was the root of a graph of depth \( \leq n \) prior to the merge. Let the graph rooted in R' be called G'.

182
Let the right daughter of R be called R''. This node was also the root of a graph of depth \( \leq n \) prior to the merge. Let the graph rooted in R'' be called G''.

Subcase 2A: The internal node is remerged to the left.

Let the nodes that are fully dominated by R'' but not fully dominated by R' be called the nodes in G'' - G'.

(70)

Recall that all nodes dominated by R', a multi-dominated node, are fully dominated by R'. Therefore, there are no nodes that are both in G'' - G' and dominated by R'. For terminals x and y, each may be either in G'' - G' (i.e. fully dominated by R'' but not R') or in G' (i.e. fully dominated by R').

If both x and y are in G', they are in a “self-contained” subgraph of depth <n. (G' is isomorphic to an independent graph of depth <n.)

Conjecture 2 is true up to depth n. Therefore, in this situation, x and y are ordered with respect to one another.

If x is in G' and y is in G'' - G', then R' fully dominates x and R'' fully dominates y. Since R' sister precedes R'', x precedes y.
If \( x \) is in \( G'' - G' \) and \( y \) is in \( G' \), then \( R'' \) fully dominates \( x \) and \( R' \) fully dominates \( y \). Since \( R' \) sister precedes \( R'' \), \( y \) precedes \( x \).

If both \( x \) and \( y \) are in \( G'' - G' \), then \( x \) and \( y \) were ordered in \( G'' \) prior to the remerge and the remerge either did not affect their ordering or did affect their ordering.

The latter case, however, leads to a contradiction:

Suppose the latter case is true. Then the sequence of nodes that established the ordering must have included terminals dominated by \( R' \). The goal is to now show that including terminals dominated by \( R' \) is not essential. We can do this by showing that if there is sequence of nodes ordering \( x \) and \( y \) that includes nodes in \( G' \), then there is another sequence of nodes ordering \( x \) and \( y \) that does not include nodes in \( G' \).

To say that the sequence of nodes establishing ordering between two terminals in \( G'' - G' \) includes some terminal dominated by \( R' \) is to say that in the sequence, \( \exists Z \) and \( Z' \) in \( G'' \) and \( \exists z \), and \( z' \) in \( G' \), such that

\[
\begin{align*}
Z \text{ is a right sister to some node in } G'' - G', \\
Z \mathcal{FD} z, \\
Z' \text{ is a left sister to some node in } G'' - G', \text{ and} \\
Z' \mathcal{FD} z'.
\end{align*}
\]

Because \( R' \) fully dominates every node that it dominates, it cannot be the case that \( Z \) or \( Z' \) is dominated by \( R' \) while its sister node
is in $G'' - G'$ (and thus not fully dominated by $R'$). This means that $Z$ and $Z'$ must be $R'$ itself or some node dominating $R'$.

Let us call $Z$’s sister $A$. $A$ is a left sister.

Let us call $Z$’s sister $B$. $B$ is a right sister.

Since $Z$ and $Z'$ are both along the “spine” from $R'$ to $R''$, $A$ and $B$ are in a c-command relation to one another. (If a node is along the “spine,” then its parent is also along the “spine” and its sister is a daughter off the “spine.” Two daughters off the “spine” are in a c-command relation.)

Since contraction is restricted to nodes in a c-commanding relationship, whether $A$ contracts or not, the LNFD$_A$ will be along the “spine.” Whether $B$ contracts or not, the LNFD$_B$ will be along the “spine.”

If LNFD$_A$ dominates the LNFD$_B$, then $A$ will sister precede some node that fully dominates $B$ (and $t_B$).

If LNFD$_B$ dominates the LNFD$_A$, then some node that fully dominates $A$ (and $t_A$) will sister precede $B$.

Since $t_A$ and $t_B$ are terminals in $G'' - G'$, and $A$ and $B$ are nodes in $G'' - G'$, we can replace a subsequence of nodes in $G'$ with nodes in $G'' - G'$ in any sequence of nodes establishing ordering between two terminals that are both in $G'' - G'$.
Thus, if both \(x\) and \(y\) are in \(G'' - G'\), then \(x\) and \(y\) were ordered in \(G''\) prior to the remerge and the remerge does not affect their ordering.

Conjecture 2 is true in subcase 2A.

Subcase 2B: The internal node is remerged to the right.

Conjecture 2 can be argued to hold in this subcase using a line of reasoning that mirrors the line used for arguing conjecture 2 holds in subcase 2A.

Conjecture 2 is true in case 2.

When conjecture 2 is true for graphs of depth up to \(n\), it is also true for \(n+1\).

Therefore, Conjecture 2 is true.

Though this proof outline is not fully formalized, a picture of the conditions that are likely to be involved in establishing the necessary conditions and sufficient conditions for totality in binary branching graphs begins to emerge. The notions of the least node fully dominating a node \(X\) and c-command are likely to be useful ones.

6.4 CONCLUSION

In this chapter, we have explored some linguistic and formal consequences of imposing a linearizability requirement on syntactic graphs. On the linguistic front, the linearization requirement had the effect of closing the door to certain proposals, such as the structure posited in earlier chapters for \(wh\)-object questions and the hypothesis that all cases of
coordination can be recast as sentential coordination. In addition, the linearization requirement also predicted an observed pattern, the order preservation effects in Scandinavian object shift, and opened the door to other proposals, such as an account of the binding of Mandarin *zi4ji3* that conforms to Binding Theory and prohibits adverbial intervention between elements that behave as a constituent. On the formal front, we identified a non-trivial necessary condition for satisfying antisymmetry and provided an outline for the proof for a non-trivial sufficient condition for satisfying totality. Along the way, we also discussed notions that play a role in avoiding violations of antisymmetry and and/or totality, such as consistent sisterhood and a node X’s least node fully dominating X.

The remaining sections include discussion that is related but not central to this chapter: an approach to gapping and deletion as a strategy for repairing otherwise ill-formed structure.

6.5 DIVERSION I: THE NEED FOR AN ANAPHORIC APPROACH TO GAPPING

Gapping is a coordinate construction in which the verb and sometimes some other material is unpronounced in the second conjunct. (65) is a simple case in which the verb is gapped in the second conjunct:

(71)  *Sam likes beans and Joe, rice.*

In section 6.3.2, we began to consider the possibility of sharing of verbs in coordinate structures in addition to sharing arguments. We have not yet, however, discussed the
sharing of the verb alone. Such a scenario is the obvious multi-dominance analysis for the gapping construction.\footnote{Indeed, node contraction of lexical anchors is the analysis that Sarkar and Joshi (1996) pursue for gapping. They propose that a lexical anchor may project two elementary trees prior to actually being inserted into the tree. That is, a lexical anchor first projects a piece of structure, the projection it is typically assumed to project, but the anchor is itself temporarily a separate piece, not part of that structure. Only after the preterminals for the anchors contract is the lexical anchor inserted into the structure. This ordering sidesteps the need to provide some additional mechanisms to delete/hide one copy of the lexical verb from the phonology and compute agreement. Adopting this strategy for gapping would yield a structure like (66). To handle more complex cases of gapping (discontinuous gaps, verb+object gaps), Sarkar and Joshi allow concurrent contraction of anchors and non-anchors. Unfortunately, without restrictions, allowing concurrent contraction of anchors and non-anchors overgenerates, regardless of where we take those nodes to be pronounced. For example, simultaneous gapping and Right Node Raising is impossible, regardless of whether a contracted object DP node is pronounced in the first conjunct or the second conjunct.}

\begin{equation}
(72) \quad \text{Unlinearizable structure for } \textit{Sam likes beans and Joe, rice.}
\end{equation}

This graph satisfies antisymmetry. The verb is the left daughter of both of its parents. This correctly provides the ordering information that the verb should be pronounced in the first conjunct, never the second. However, the graph fails to satisfy totality. Note that \textit{likes} is only fully dominated by V and the root BP. None of the nodes fully

\footnote{\textit{Too} is not compatible with even core cases of gapping where only the verb is gapped. This likely results from a semantic clash. Gapping requires contrast while \textit{too} requires sameness.}
dominating *Sam*, DP1, VP1, and BP, are in a sister precedence relation that would result in an ordering between *Sam* and *likes*. As the case where only the verb is gapped is the simplest case of gapping, this does not bode well for a multidominance approach to gapping. A slightly more complex example, where both auxiliary and verb are gapped, yields even more totality violations.

(73) Unilinearizable structure for *Harry will buy bread, and Barry, potatoes*.

In addition to the subject of the first conjunct, *Harry*, being unordered with respect to the gapped elements, the gapped elements are also unordered with respect to one another. The only nodes that fully dominate *will* are T and the root. The only nodes that fully dominate *buy* are V and the root. The four sister precedence relations involving T and V and the linearization information they yield are:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>S</td>
<td>VP1</td>
<td>→   will</td>
</tr>
<tr>
<td>T</td>
<td>S</td>
<td>VP2</td>
<td>→   will</td>
</tr>
<tr>
<td>V</td>
<td>S</td>
<td>DP2</td>
<td>→   buy</td>
</tr>
<tr>
<td>V</td>
<td>S</td>
<td>DP4</td>
<td>→   buy</td>
</tr>
</tbody>
</table>
Instead, the totality requirement leads me to propose that gapping constructions do not involve multi-dominance. Rather, gapping involves anaphora between verbal elements. That is, a fuller representation of (65) is: *Sam likes, beans, and Joe pro-V, rice*, and this pro-V depends on a bona fide verb for its interpretation. As the structure associated with gapping would be a tree, the linearization problem is avoided.

(74) Linearizable structure for *Sam likes beans and Joe, rice.*

An anaphoric approach to gapping has been mentioned (Steedman 1990, Johnson 1993), but gapping has more often been analyzed as a type of VP ellipsis (Hankamer 1973, 1979, Larson and May 1990, Neijt 1979, and others), remnant + Across the Board (ATB) movement (Johnson 2003), and actually involving constituents (Steedman 1990). I imagine that the dependency between pro-V and its antecedent would resemble that of the covert pronominal and its antecedent in that both involve a phonologically silent category. Though I have not characterized the structural relation that must hold between a verb and the pro-V on which it depends, we are likely to be able to specify it in various ways such that only coordinated sentences would instantiate such a relationship. In this way, pro-V shares similarities with reflexive pronouns in that both obligatorily require an
antecedent that satisfies certain structural requirements. The statement below is unfortunately not explanatory, but suffices to show that an adequate structural requirement for the antecedent of *pro*-V can be straightforwardly stated.

(75) *pro*-V Semantic Inheritance Condition

*pro*-V inherits the semantic information of a single verb A iff A has an ancestor that c-commands *pro*-V but A does not itself c-command *pro*-V.

The core characteristic that the gap never appears in the first conjunct follows from this condition, as do a number of identity requirements between the verb and the gap, requirements that do not apply to VP ellipsis. First, an active VP may not be identified with a passive VP. (Here, I adopt Johnson’s notation of placing gapped material in strike-outs and use Δ to denote VP ellipsis.)

(76) * identifying an active and passive VP (Johnson 2003)

a. Ellipsis: Botanist: That can all be explained.

   Mr. Spock: Please do Δ.

b. Gapping: * The budget cuts might be defended publicly by the chancellor, and the president might defend publicly her labor policies.

A passive predicate carries with it properties that differentiate it from its active counterpart, such as generally requiring the auxiliary *be* and fewer arguments. If *pro*-V inherits all the properties of its antecedent, then *pro*-V cannot inherit an active interpretation from a passive antecedent.

191
Second, a gap may not have an antecedent fashioned out of two VPs. If pro-V inherits the semantic information of a single verb, then it will not be able to have two antecedents.

(77) * an antecedent fashioned out of two VPs (Johnson 2003)
   a. Ellipsis:  *Wendy is eager to sail around the world and Bruce is eager to climb Killimanjaro, but neither of them can ∆ because money is too tight.
   b. Gapping:  *Wendy should sail the English Channel and Bruce climb Whitney, and their partners should sail and climb the Pacific or Killminjaro.

Third, a gap may not have an antecedent that is inside a DP. If pro-V inherits the semantic information of a single verb, then it will not be able to inherit semantic information from a DP.

(78) * an antecedent from inside a DP (Johnson 2003)
   a. Ellipsis:  ?Sal is a talented forger, but Holly can’t ∆ at all.
   b. Gapping:  *Sal may be a forger of passports and Holly may forge paintings.

However, the identity condition is not so stringent as to demand exact identity between the gap and its antecedent, allowing examples like the ones below, taken from Johnson (2003). These examples are meant only to show that the identity condition is not literal identity.
(79) *Every girl showed her project to the teacher, and every boy ______ to the principal.*

   a. Every girl showed her (e.g. Jane’s) project to the teacher, and every boy showed her (Jane’s) project to the principal. (non-distributed reading) OR
   b. Every girl showed her own project to the teacher, and every boy showed his own project to the principal. (distributed reading)

(80) Some gave a book to Sally and others ______ to Jim.

In (73), the gapped pronoun may refer to the same girl referred to by her or may be bound by every boy. In (74), it need not be the case that Sally and Jim received the same book.

The ban against totality violations rules out (66) as a potential analysis for sentences with gapped verbs. An anaphoric approach to gapping both avoids the linearization difficulties and provides some degree of explanation for the identity requirements that do not amount to literal identity.

6.6 DIVERSION II: DELETION AS A REPAIR?

6.6.1 LASNIK’S (2001) PROPOSAL AND LINEARIZATION VIOLATIONS

Lasnik (2001) has proposed a deletion operation that serves as a last resort for avoiding ill-formed phonological material. Lasnik’s proposal is framed within a feature-checking system, and he casts the ill-formedness in terms of features that are uninterpretable by the phonological system and thus must be eliminated prior to making the information available to the phonological system. Here, we recast his notion of avoiding ill-formed phonological material as the idea that deletion could repair violations of antisymmetry
and totality. This section is an exploratory look at the possibility that there exists an operation that severs a subtree from the original tree and deletes it and has the effect of eliminating asymmetry or totality violations.

6.6.2 Deletion as Antisymmetry Violation Evasion: Pseudogapping

Psuedogapping refers to constructions such as those below from Lasnik (2005) and Levin (1978), respectively.

(81) a. You might not believe me, but you will ___i Bob.
    b. I rolled up a newspaper, and Lynn did ___i a magazine.

Pseudogapping shares some properties of gapping (right side remnant, a portion of the VP survives the deletion) and some properties of VP ellipsis (a finite auxiliary remains). Lasnik’s analysis (1995) adopts an approach attributed to Koizumi (1995) and Johnson (1991) where elements originally belonging to the VP first move out of the VP prior to VP deletion. If we follow Lasnik (1995) in taking pseudogapping to involve the complement of a verb raising to a spec position followed by deletion, then we can also adopt his proposed motivation of avoiding ill formed phonological material for deletion. As an example, consider the structure for (75a) below.
Pre-deletion structure for *You might not believe me, but you will Bob.*

Note that without deletion, DP4 sister precedes V'2. Since DP4 fully dominates *Bob* and V'2 fully dominates *believe*, *Bob* must precede *believe*. However, V2 also sister precedes DP4. Since V2 fully dominates *believe* and DP4 fully dominates *Bob*, this sister precedence relation provides the conflicting ordering information that *believe* precedes *Bob*.

Suppose now that we delete the lowest piece of structure that would eliminate conflicting information. Here, we sever edge connecting V2 to V'2. What the severing represents is the deletion of all pairs from the primitive relations which include V2 as an element. In our immediate dominance relation, this has the effect of removing (V'2, V2) and (V2, *believe*). In our sister precedence relation, this has the effect of removing (V2, DP4) from the set. The resulting linearizable graph is given below.
Post-deletion structure for *You might not believe me, but you will Bob.*

6.6.3 Deletion as Totality Violation Evasion: Island Violations

Ross (1969) observed that island violations could be improved by deleting elements following the fronted wh-element. Two examples from Ross (1969) contrasting post-deletion constructions and their non-deletion counterparts are given below. 17

(84)  

a.  *I believe the claim that he bit someone, but they don’t know who, I believe the claim that he bit ___*.  
b.  *I believe the claim that he bit someone, but they don’t know who, I believe the claim that he bit ____.*  

(85)  

a.  *Irv and someone were dancing together, but I don’t know who, Irv and ___ were dancing together.*  
b.  *Irv and someone were dancing together, but I don’t know who, Irv and ____, were dancing together.*  

17 Ross marks (78b) and (79b) with ?? as he considers them only improved, not fully repaired. For many speakers, however, the b sentences are fully acceptable.
If, within a multidominance approach, we take the identity requirement between the elided section and its antecedent to be literal identity, the structure for (78b) is that given below.

(86) Pre-deletion structure for (78b)

In (80), all the shared elements of the first and second conjunct are fully dominated only by the root, leading to massive totality violations. Neither the full-dominance cousin-precedence condition nor transitivity can order any of the shared elements with respect to one another. If, again, we delete the lowest piece of structure that would eliminate ordering violations, we should sever the edge between CP and TP3. This yields the string in (78b).
6.6.4 Remaining Issues

These examples show only that deletion can have the effect of repairing otherwise unlinearizable structures and that this is a point of contact with Lasnik’s (2001) proposal. A number of questions remain, both about the particular linguistic phenomena and the deletion operation. First, in chapter 4, I claimed that the MD-TAG system does not generate structures with island violations in the first place, leaving open the question of how a structure such as (80) would be derived at all. Second, the pseudogapping example above has an antisymmetry violation, but consider now a psuedogapping example in which a subject remains instead of an object.

(87)  You may not think, Mary [will win the race], but you should ___; Sue _____.

As before, we assume the argument to raise above the verb. Since the subject of the embedded clause is a left daughter that contracts with another left daughter node, there is no antisymmetry violation. However, if we take the identity requirement between the elided material and its antecedent to be literal identity, that is, the shared material is multi-dominated, then there are a number of totality violations. Subsequent deletion at the edge between VP3 and V3’ eliminates these violations. This is an outcome in favor of proposing a deletion operation that rescues otherwise unlinearizable structures.
In the gapping section above, however, we presented data indicating that gapping requires stricter identity than VP ellipsis. This suggests that the type of identity required for gapping, which I argued to be anaphoric, serves an upper bound on the type of identity required for VP ellipsis. It will be difficult maintain that the verbal elements for gapping, which requires tighter identity than VP ellipsis, are not multi-dominated while the verbal elements for VP ellipsis are multi-dominated. We also argued that even the identity restriction for gapping did not require literal identity. A similar example involving VP ellipsis can be constructed.

(89) * You may not think Mary [will sell her house], but you should ___ Sue ______.
If this sentence means that the addressee should think that Sue will sell Sue’s house, then it is difficult to argue that *her house* is a multidominated DP since *her house* does not have a single referent: *her house* = Mary’s house in one conjunct, but *her house* = Sue’s house in the other. If neither the verbal elements nor the object are shared across conjuncts, the linearization difficulties disappear as well.

Finally, I have not made claims about when deletion applies. These examples only show that when deletion occurs, totality violations disappear. They do not show, however, that deletion only applies for the sake of repairing linearization problems. Deletion may have other triggers. Nor have I given a reason for why deletion does not always apply to save structure. For example, the cases of sentential coordination that were rejected on the basis of failing to satisfy totality do not improve when structure is truncated. Consider (53) repeated here as (84).

\[
\text{(90) Structure for } \textit{John and Mary ate cookies}, \text{ with totality violations }
\]

\[
\text{If deletion can eliminate the totality violations, why should the following truncation be unacceptable?}
\]

\[
\text{(91) * John and Mary}\ldots\ldots.
\]
CHAPTER 7
LINEARIZABLE MD-TAG GRAPHS AND GENERATIVE COMPLEXITY

7.1 INTRODUCTION

Earlier, in chapter 5, I pointed out that the graphs generated by an MD-TAG that adopted certain restrictions on elementary structures (e.g. the theta criterion) are a subset of the set of the formal objects I called syntactic graphs. Through the course of chapter 5 and 6, we saw that syntactic graphs that are linearizable are also a proper subset of syntactic graphs. We are now in a position to begin considering the intersection of graphs generated by the MD-TAG formalism described in chapter 3 and linearizable by the proposal in chapter 5.\(^1\) This chapter is a brief exploration of the string languages that such graphs yield. As we shall see, this set includes languages that are not in the class of languages definable by classic TAG. Section 7.2 briefly reviews the previously established finding that TAGs are more expressive than context free grammars but less expressive than context-sensitive grammars. Section 7.3 shows how an MD-TAG grammar can generate structures that, when linearized, yield a language with five counting dependencies, a language that is outside the capacity of TAG. A small modification shows that languages with six counting dependencies can be generated as well. Section 7.4 shows how MD-TAG grammars also allow scrambling languages. Section 7.5 summarizes the ideas presented.

\(^1\) Though this chapter is concerned with the restrictions on syntactic graphs with respect to linearization, it sets aside the restrictions on generation given in Chapter 4. Node contraction in these examples does not always respect the locality condition on node contraction. The locality condition was motivated by the existence of island effects, a phenomenon of natural languages, and it is possible that the particulars of the locality condition vary from language to language. If, however, the particular derivational locality condition given in Chapter 4 is part of the formalism (and thus is supposed to hold across all languages), it may restrict the class of generable languages. I leave open the question of the impact of the locality condition on generative capacity.
7.2 BEYOND CONTEXT FREE LANGUAGES

As was pointed out in the introduction to TAG in Chapter 2, TAG itself is not a linguistic theory but a formalism. The explicit definition of the TAG system as a mathematical object has allowed numerous formal properties of the system to be established. A general overview can be found in Joshi, Vijay-Shanker, and Weir (1991). A few results are highlighted here.

First, the set of languages that can be generated by a TAG contains the set of languages that can be generated by a context free grammar. Each rule of a CFG can be converted into a tree consisting of a root node (the symbol to be rewritten) and its daughters (the symbols post-rewriting). If we take the non-terminal daughter nodes to be substitution nodes, the set of trees is a TAG without auxiliary trees. In addition, a TAG can define languages that are beyond the expressiveness of context free grammars. For example, a language with four counting dependencies, e.g. $a^n b^n c^n d^n$, cannot be generated by a context free grammar, but can be generated by TAG combination of the elementary trees below. (Note that to generate $a^n b^n c^n d^n$, it is necessary to prevent adjoining from taking place at the root X node and foot X node of the auxiliary tree in (1b). To ensure that adjoining is prevented at certain nodes, we use Vijay-Shanker and Joshi (1985)’s proposed system in which nodes may be marked as null-adjoining nodes (NA), nodes at which no adjoining may take place. It is not known whether a TAG without NA nodes can generate $a^n b^n c^n d^n$.)
(1) Tree set for $a^n b^n c^n d^n e^n$

a. 

```
    S
   / \  
  X   d
 /    
 b   c
```

b. 

```
    X(NA)
   /   
  a   d
 /     
 b   X(NA) c
```

The auxiliary tree in (1b) may adjoin into the X node of (1a) or into the middle X node of itself. Each time (1b) adjoins, a single $a$, $b$, $c$, and $d$ is inserted into the string that is read off the tree. The adjoining of (1b) into itself and subsequent adjoining of the result into (1a) is shown below. (Double subscripts are intended to facilitate reconstruction of which nodes belonged to which trees.)

(2) Derivation of $aaabbbccccddd$

```
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>a1</td>
<td>b1</td>
<td>c1</td>
</tr>
<tr>
<td>S1</td>
<td>a1</td>
<td>b1</td>
<td>d1</td>
</tr>
<tr>
<td>X2</td>
<td>a2</td>
<td>b2</td>
<td>c2</td>
</tr>
<tr>
<td>X2</td>
<td>a2</td>
<td>b2</td>
<td>d2</td>
</tr>
<tr>
<td>X1</td>
<td>a1</td>
<td>b1</td>
<td>c1</td>
</tr>
<tr>
<td>X1</td>
<td>a1</td>
<td>b1</td>
<td>d1</td>
</tr>
</tbody>
</table>
```

Second, though TAG languages include languages outside the class of context-free languages, they are not equivalent to the next set of languages in the Chomsky hierarchy, the context-sensitive languages. For example, the context-sensitive language
a^n b^n c^n d^n e^n cannot be generated by any TAG (Vijay-Shanker 1987). Informally, we can see that TAG elementary trees have no place to “hang” a fifth element e such that inserting an auxiliary tree could keep the fifth element distinct from the other four. TAG languages fall into an intermediate class, the class of mildly context-sensitive languages.

The notion of mildly context-sensitive languages was first introduced by Joshi (1985). As the purpose of the notion was to describe a class of formal languages that would be well-suited for describing natural languages, the notion is not fully formalized. Rather, the class of languages is characterized by having at least the properties in (3).

(3) Criteria for the class of mildly context-sensitive language
a. The class of languages contains the context-free languages as a proper subset.
b. The languages of the class can be parsed in polynomial time.
c. The languages in the class capture only certain types of dependencies, such as nested dependencies and certain kinds of crossing dependencies that have been observed in natural languages (Culy 1985, Shieber 1985), but not others, such as those exhibited by the mix languages (in which the strings are composed of the same number of a fixed number of different terminal symbols, e.g. strings with an equal number of a’s, b’s, and c’s in any order).
d. The languages in the class have the constant-growth property. i.e. if all the strings of a language are arranged in increasing order of length, then there is some bound (for that language) on the difference in length between two consecutive strings.

Interestingly, a number of other formalisms proposed for modeling natural language, head grammars (Pollard 1984), linear indexed grammars (Gazdar 1988), and categorial grammars (Steedman 1990, 1996), have been shown to generate only mildly context-sensitive languages. Not only so, but these formalisms have been shown to be
equivalent, even though the formal objects and operations employed are quite different. These results are summarized and synthesized in Joshi, Vijay-Shanker, and Weir (1991). As they point out, the fact that these formalisms with the same goal but different approaches turn out to be equivalent suggests that the community is on the right track for understanding formal properties of linguistic objects.

7.3 BEYOND FOUR COUNTING DEPENDENCIES

While languages with five counting dependencies are outside the class of languages generated by traditional TAG grammars, an MD-TAG grammar can define such a language. As we have seen earlier with the linguistic examples of question formation and interleaved constructions, node contraction between elementary trees allows what is traditionally considered the base-generated position and the surface position to be in different elementary trees. Here, we will posit that the node that ultimately dominates the fifth terminal is a multi-dominated node that originated as two contraction-marked nodes in two different elementary trees. The set of elementary trees that generate the language $a^n b^n c^n d^n e^n$ is given in (4) below.

(4) Tree set for $a^n b^n c^n d^n e^n$

a. $S \rightarrow a X Y$
   $X \rightarrow b c d e$

b. $X(\text{NA}) \rightarrow a X Y$
   $Y \rightarrow b c X(\text{NA}) d e$

[c. $S \rightarrow S Z$
   $Z \rightarrow e$]
(5) illustrates the derivation after tree (4b) adjoins into itself, (4c) adjoins into itself and the resulting structures adjoin into (4a), the derivation for \textit{aaabbbccccdddeee}.

(5) \textbf{Derivation of \textit{aaabbbccccdddeee}}

Here, the requirement that the finished derivation be contraction-node-free is what ensures that all the e’s move to the left of all the d’s. Further, because we prohibit contraction of two nodes when both dominate additional structure (a recasting of the Phonological Content Condition on Node Contraction (chapter 3, (7)) in non-linguistic terms), to ensure a contraction-node free derivation, every instance of (4b) in the derivation must also have a corresponding instance of (4c). This in turn ensures that for every group of terminals \textit{a, b, c,} and \textit{d} introduced (by (4b)), there is also an \textit{e} introduced (by (4c)). (The derivation of this language also depends on the assumption that a node marked for contraction loses this marking once it undergoes contraction.) Below, we give the derived phrase structure at the end of the derivation in (5).
This graph is linearizable according to the linearization process given in chapter 5. Since $S_{3,5}$ fully dominates $e_4$, $e_4$ will be linearized with respect to $e_2$. The sister precedence pairs relevant to ordering $e_4$ and $e_2$ are $(S_{1,3}, Z_{4,3})$ and $(S_{3,5}, Z_{2,5})$:

<table>
<thead>
<tr>
<th>$S_{1,3}$</th>
<th>$S$</th>
<th>$Z_{4,3}$</th>
<th>$a_1, a_2, a_4, b_4, b_2, b_1, c_1, c_2, c_4, d_4, d_2, d_1, e_1, e_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{3,5}$</td>
<td>$S$</td>
<td>$Z_{2,5}$</td>
<td>$a_1, a_2, a_4, b_4, b_2, b_1, c_1, c_2, c_4, d_4, d_2, d_1, e_1, e_3, e_5$</td>
</tr>
</tbody>
</table>

The final linearization is $a_1, a_2, a_4, b_4, b_2, b_1, c_1, c_2, c_4, d_4, d_2, d_1, e_1, e_3, e_5$.

A slight modification of the MD-TAG grammar above allows us to define a language with six counting dependencies, $a^n b^n c^n d^n e^n f^n$. Just as the top right daughter of the root in (4a) and (4b) both branch, the top left daughter of these roots may also branch.
We can use the mirror image of the same strategy as before. The set of elementary trees that generate the language $f^{n} a^{n} b^{n} c^{n} d^{n} e^{n}$ is given below.

(7) Tree set for $f^{n} a^{n} b^{n} c^{n} d^{n} e^{n}$

a. 

```
  S
 /\   
W  X   Y
 /\    /\  
f   a   b   c
```

b. 

```
  S
   X(NA)
    W  X  Y
     /\ /\ /\ 
    a  b  c  d
```

c. 

```
  S
   V
    S
     S
      Z
```

d. 

```
  S
   f
   e
```

Below, we give the derived tree after tree (7b) and tree (7d) have each adjoined into tree (7a) once, and tree (7c) has adjoined into the root of tree (7d) once. (Subscripts have been added to assist in bookkeeping.)

(8) Derived tree for $f a a b b c c d d e e$

```
  S4
 /\   
S3,4  S1,3
     /\   /\  
W1  X1,2  Y1
    /\  /\ /\  
f1  a1  d1  e1
```

```
  S2,4
 /\   
W2  X2
 /\ /\ /\ 
 f  a  b  c  d 
```

```
  S1,2
 /\   
W3  X3
 /\ /\ /\ 
 d  b  c  d 
```

```
  Z2,3
   e3
```

208
This graph is linearizable as well. The sister precedence relation between $e_3$’s preterminal and its sister will order $e_3$ behind all terminals fully dominated by $S_{1,3}$. The sister precedence relation between $f_4$’s preterminal and its sister will order $f_4$ in front of all terminals fully dominated by $S_{3,4}$.

<table>
<thead>
<tr>
<th>$S_{1,3}$</th>
<th>$SP$</th>
<th>$Z_{2,3}$</th>
<th>$\rightarrow$</th>
<th>$f_1, a_1, a_2, b_2, b_1, c_1, c_2, d_2, d_1, e_1, e_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{2,4}$</td>
<td>$SP$</td>
<td>$S_{3,4}$</td>
<td>$\rightarrow$</td>
<td>$f_4, f_1, a_1, a_2, b_2, b_1, c_1, c_2, d_2, d_1, e_1, e_3$</td>
</tr>
</tbody>
</table>

The final linearization is $f_4, f_1, a_1, a_2, b_2, b_1, c_1, c_2, d_2, d_1, e_1, e_3$.

We have seen that the set of languages that can be generated by a MD-TAG intersected with the set of string languages yielded by graphs that are linearizable includes languages with five counting dependencies and languages with six counting dependencies. My intuition is that it is possible for the MD-TAG system to generate structures that go beyond six counting dependencies. We can continue to pursue the same strategy used thus far: replace one of the terminal nodes in tree in (1a) with a branching non-terminal instead and include additional trees that keep each type of terminal separated out. However, it seems unlikely to me that structures with more than six counting dependencies will be linearizable. This conjecture is related to the edge-condition and how it relates to linearizability discussed in chapter 6. As there are only two edges to a clause, node contraction may only be able to add two dependencies to the four dependencies that were possible in traditional TAG. We turn now to another language outside the class of TAG languages.
7.4 SCRAMBLING

Scrambling languages are also outside the class of languages definable in TAG. In fact, under the constraint that all and only coindexed symbols are introduced at the same point in the derivation, scrambling has been shown to be beyond Linear Context Free Rewriting systems by Becker, Rambow, and Niv (1992).

The following set of trees generates the language $ww_s$, where $w_s$ is any ordering of the elements in the string $w$. The two non-auxiliary elementary trees are in (9a) and (9b). In order to satisfy the requirement that the final derived structure be contraction node free, note that at least one instance of the auxiliary tree in (9e) must participate in a derivation involving (9a) and at least one instance of the auxiliary tree in (9f) must participate in a derivation involving (9b). The auxiliary trees in (9c) and (9d) may substitute into any X node. Again, for the sake of deriving a final structure that is contraction node free, for each instance of tree (9c) that participates in the derivation there must be an instance of tree (9e) that participate in the derivation. Each instance of (9c) contains a node marked for contraction with the label A. Assuming contraction of two nodes which both dominate terminals is prohibited, an A node in tree (9c) cannot contract with another A node in another instance of tree (9c). Thus, an instance of (9e) is needed to eliminate the contraction node in (9c). Similarly, every instance of tree (9d) in the derivation must correspond to an instance of (9f) in the derivation to eliminate B-labeled nodes marked for contraction.
While these trees do introduce associated a’s and associated b’s at the same time, the nodes whose presence is required to accomplish scrambling are not simultaneously introduced. That is, the preterminals A and B which are contracted are not introduced in the same derivation step as the occurrences of the As and Bs with which they contract. This is a different system than any of those considered by Becker, Rambow, and Niv (1992). If we modified their constraint on the introduction of coindexed symbols to be a constraint on nodes that contract (that is, say nodes that ultimately contract must be introduced in the same derivation step), then the grammar in (9) does not satisfy the node-contraction version of the constraint. For the grammars they consider, requiring that the verb and its arguments are introduced in the same derivational step is the same as requiring all coindexed symbols to be simultaneously introduced. In an MD-TAG, however, nodes that are contracted are allowed to belong to different trees, and hence, to different derivational steps. This extra flexibility allows associated elements (suppose, for example, that one of the a’s in (9c) is a verb while the other a is its argument) to be simultaneously introduced in a derivation step while also allowing scrambling. The resulting structures also are linearizable. The nodes marked for contraction are all left
sisters, so no conflicting orderings will arise since maintaining consistency is sufficient for ensuring antisymmetry. Likewise, the graphs satisfy the conditions we have argued are sufficient for preventing totality violations, binary branching in conjunction with only self-c-commanding multi-dominance nodes and no remnant movement. Because the elementary trees are binary branching, the derived structure will also be binary branching. Since the trees in (9e) and (9f) may only adjoin at an S node, all of which are above the preterminal A and B nodes, it appears that node contraction will be limited to nodes in a c-command relationship. There is also no remnant movement.

7.5 CONCLUSION

The examples in this chapter show that adding a node contraction operation increases the generative power of an MD-TAG system beyond that of classic TAG. A grammar for languages with up to six counting dependencies is possible, but it is not obvious if we will be able to generate linearizable graphs for counting dependencies beyond six. A grammar for a scrambling language is not definable in classic TAG is also definable in MD-TAG. Since every classic TAG grammar is a valid grammar in MD-TAG, it is clear that TAG languages are a proper subset of MD-TAG languages.

Finally, it has been noted that the TAG framework bears certain similarities to the Minimalist Program (Frank 2002). However, one aspect where classic TAG differs from minimalist grammars, specifically Stabler’s (1997) formalization of the Minimalist Program, is in the area of generative power. Stabler (1997) shows that his formalization can define languages with five counting dependencies and the scrambling language \( w_ww_s \). This is a point of similarity with MD-TAG, but it is not clear yet what the relationship is
between languages definable with a MD-TAG and those definable with a minimalist grammar. However, we do know something about the relationship between minimalist grammars and a different extension of TAG called multi-component TAG (first discussed by Joshi, Levy, and Takahasi 1975, and later defined precisely by Weir 1998). Multi-component TAG is equivalent to linear context-free rewriting systems (Vijay-Shanker, Weir, and Joshi 1987, and Weir 1988). Thus, the results of Michaelis (1998, 2001) that show equivalence between minimalist grammars and linear context-free rewriting systems also establish the equivalence of minimalist grammars and multi-component TAGs. It appears to me likely that some sort of equivalence between MD-TAG and multi-component TAG can be established, though, at the moment, this is only a conjecture. We will not be able to establish equivalence of the phrase structures in the traditional sense since MD-TAG allows explicitly multi-dominated nodes while MC-TAG does not. It is also unclear how to establish equivalence of the string languages of MD-TAG and MC-TAG, because MD-TAG generates graphs that do not yield a unique linearization of terminals. Instead, we will need to use the notion of syntactic graphs defined in chapter 5, a five-tuple consisting of a set of nodes, a set of labels, an immediate dominance relation, a sister-precedence relation, and a labeling function. Also, we will need to consider co-indexed nodes in MC-TAG to be the same node as far as immediate dominance is concerned. We will then need to show that for every MD-TAG, the set of syntactic graphs that are generated could have been generated by an MC-TAG and vice versa, that for every MC-TAG, the set of syntactic graphs that are generated could have been generated by an MD-TAG.
I imagine the first construction (converting an MD-TAG to an MC-TAG) would involve taking the elementary trees with nodes marked for contraction and creating sets of such trees. A tree with a single XP node marked for contraction would be in a set with every tree with a single XP node marked for contraction. A tree with two nodes marked for contraction, XP and YP, would be in a two-tree set with every tree with an XP node and YP node marked for contraction and also in a three-tree set with all combinations of a tree with an XP contraction node and a tree with a YP contraction node (an so on). The number of elementary trees is finite, and the number of nodes in each tree itself is finite, so the number of sets is finite. In each set, the contraction marking would be replaced with a co-indexation.

The second construction (converting an MC-TAG to an MC-TAG) would involve 1) re-labeling the co-indexed nodes of each multi-component set with a label unique to that set, 2) marking those nodes of contraction, and 3) changing the status of each member of the set to be a stand-alone elementary tree.

If it is the case that the syntactic graphs generated by MD-TAG and MC-TAG are equivalent, then it may be that the class of linearizable MD-TAG graphs yields languages that fall into a class between those of traditional TAG and that of multi-component TAG/minimalist grammars/LCFRS.

Of course, if an equivalence can be established between the two systems, the question arises as to why we should suppose MD-TAG as a model of human grammar rather than MC-TAG. Recall, however, that we are interested not only in generating the strings in a natural language but also in capturing the linguistically motivated relationship between the elements. Further, we do not want to create linguistically unmotivated
relationships. Consider the coordinated structures (e.g. *Joe eats cookies and studies bridges.*) that motivated Sarkar and Joshi (1996) to consider node contraction in the first place. A multi-component analysis is possible: an elementary tree headed by *eats* and an elementary tree headed by *studies* with co-indexed subject nodes would be members of the same multi-component set. Clearly, this set does not have a unique lexical head/anchor. It is not particularly appealing to posit that *eats* and *studies* together form a composite concept that could be considered the anchor of the set. What MD-TAG affords us is a more linguistically well-motivated analysis.
REFERENCES


Joshi, Aravind K. 1985. How much context-sensitivity is required to provide reasonable structural descriptions: Tree adjoining grammars. In David Dowty, Lauri Karttunen, and Arnold Zwicky (eds.), Natural Language Parsing: Psychological,
Computational, and Theoretical Perspectives, 206-250. Cambridge: Cambridge University Press.


VITA

Joan Chunghwei Chen-Main was born in Washington, D.C. on September 23, 1979. She received a B.A. with honors in Cognitive Science from Johns Hopkins University in 2001. In the same year, she began graduate studies in the department of Cognitive Science at Johns Hopkins University. In the fall of 2006, she will begin a Language and Communication Postdoctoral Fellowship at the Institute for Research in Cognitive Science at the University of Pennsylvania.