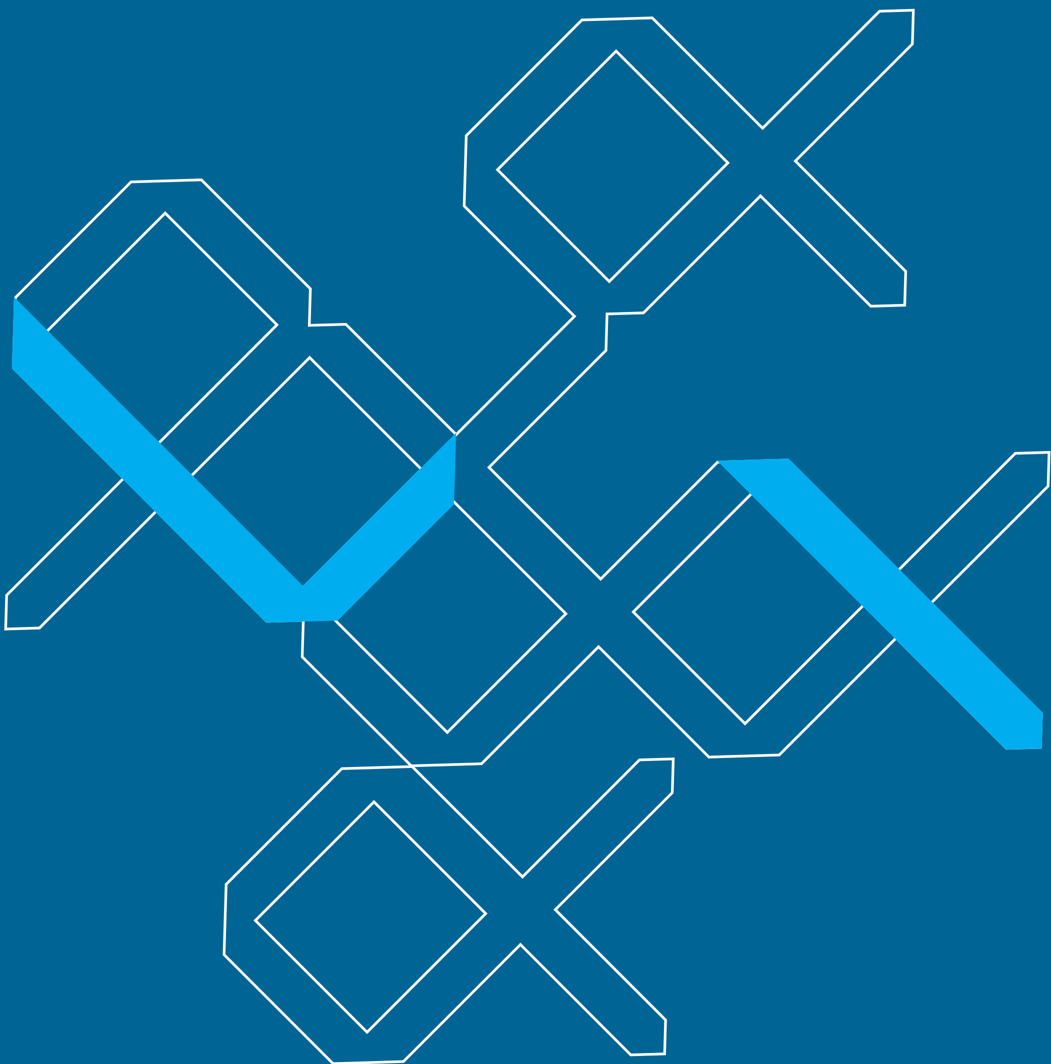
Linguistic Inquiry
Monograph Fifty-One

Locality in Minimalist Syntax

Thomas S. Stroik



Locality in Minimalist Syntax

Linguistic Inquiry Monographs
Samuel Jay Keyser, general editor

A complete list of books published in the Linguistic Inquiry Monographs series appears at the back of this book.

Locality in Minimalist Syntax

Thomas S. Stroik

The MIT Press
Cambridge, Massachusetts
London, England

MIT Press books may be purchased at special quantity discounts for business or sales promotional use. For information, please e-mail special_sales@mitpress.mit.edu or write to Special Sales Department, The MIT Press, 55 Hayward Street, Cambridge, Mass. 02142.

This book was set in Times New Roman and Syntax on 3B2 by Asco Typesetters, Hong Kong and was printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Stroik, Thomas S.
Locality in minimalist syntax / Thomas S. Stroik.
p. cm. — (Linguistic inquiry monograph ; 51)
Includes bibliographical references and index.
ISBN 978-0-262-01292-8 (hardcover : alk. paper)—ISBN 978-0-262-51276-3 (pbk. : alk. paper)
1. Grammar, Comparative and general—Syntax. 2. Minimalist theory (Linguistics) I. Title.
P295.S71 2009
415—dc22 2008035931

10 9 8 7 6 5 4 3 2 1

Contents

Series Foreword

We are pleased to present the fifty-first in the series *Linguistic Inquiry Monographs.* These monographs present new and original research beyond the scope of the article. We hope they will benefit our field by bringing to it perspectives that will stimulate further research and insight.

Originally published in limited edition, the *Linguistic Inquiry Monographs* are now more widely available. This change is due to the great interest engendered by the series and by the needs of a growing readership. The editors thank the readers for their support and welcome suggestions about future directions for the series.

Samuel Jay Keyser
for the Editorial Board

Preface

In its largest sense, my book is a study of the optimal design of human language (HL). Although I will focus on what Chomsky (2000b) calls the "displacement property" of language, I use my investigations of this property to expose the nature of the computational system that generates language.

Since Saussure first inaugurated the study of synchronic linguistics at the end of the nineteenth century, theoretical linguists have attempted to ascertain and explain the structural and logical properties of human language. Although we have made significant headway in our understanding of some of the processes humans engage in to compute and compile large, complex syntactic units from simple lexical terms (words), we still cannot fully explain the cognitive operations we use to arrange grammatical constituents into complex sentential patterns. Particularly problematic for our understanding of the design of HL is the fact that language permits a single constituent to be associated with more than one grammatical function or structural position (Chomsky calls this property the "displacement property" of HL). Given that the displacement property is a salient property of language, we will not having a working theory of language until we have a viable explanation for this property.

Several theorists have recently proposed mechanisms to account for the displacement property; however, these various mechanisms are ad hoc solutions to the displacement problem because they merely posit "displacement" operations to explain "displacement" facts. The theoretical gain here is negligible—that is, explaining displacement properties in terms of displacement operations themselves leaves us with the problem of explaining the properties of the displacement operations (we have replaced a first-order problem with a second-order problem). Subsequent inquiry into the properties of displacement operations has led theorists to posit economy conditions (a third generation of constructs) to account for the behavior of displacement operations, thereby leaving us with the (next) problem of explaining the properties of economy conditions. This solution to the displacement problem is an extremely costly one, requiring a design that multiplies theoretical ontologies.

In this book, I show that the displacement property of human language is not, as Chomsky (2001) argues, a special property of language; rather, as I demonstrate, it follows from the same combinatory operations that compile large, complex syntactic units from the merger of small lexical units. That is, all grammatical operations reduce to a single type of locally defined feature-checking operation, and all grammatical properties, no matter how displaced and dislocated they may seem, are the cumulative effects of local grammatical operations. Hence, displacement arises from iterated local operations. The significance of my analysis is not only that it offers a compelling reanalysis of the displacement problem that has long troubled grammatical theories, but also that it argues for a radically simple design for the computational system of human language—one that restricts computational operations to a single type of local operation.

This book grows out of many long hours of discussion with Luis Lopez when we taught together at the University of Missouri–Columbia. Our discussions led me to see some of the inherent problematics with minimalist assumptions and arguments (which I write about in Stroik 2000), but also to see ways to respond to these inherent problematics, as I do in this book. I cannot thank Luis enough for all his help with the earliest stages of this book project. My most recent work on the book has been shaped by some long discussions with Elly van Gelderen and especially with Michael Putnam, with whom I have subsequently collaborated on several projects related to the analysis I propose in this book. I owe Elly and Michael much for the generosity they have extended to me as I have struggled with my reanalysis of minimalism. I want to thank them profusely for all their support.

I would also like to thank the University of Missouri system for giving me a research grant to write this book. And finally, I would like to thank the editors of *Linguistic Analysis* for allowing me to include work in chapter 2 that they previously published (Stroik 1999).

And finally, I would like to thank my wife, Michelle Boisseau, for her unflagging love and her enduring support. This book is dedicated to her.

1 Optimal Design for Human Language

Theoretical Challenge

There may be no more widely held assumption about syntax than the one Rizzi (1990) invokes when he observes that "locality is a pervasive property in natural-language syntax." Theories of syntax over the past sixty years have devoted much attention to accounting for locality-in-syntax. Structuralists such as Harris (1946), Wells (1947), and Haugen (1951) developed an Immediate Constituent (IC) analysis to explain certain aspects of the local nature of constituent relationships. In addition, generative theorists have developed Phrase Structure Rules (Chomsky 1965 and Gazdar et al. 1985), X-bar theory (Jackendoff 1977), and the Merge operation (Chomsky 1995) to express the local relations among syntactic constituents.

Although all syntactic theories subscribe to some version of (1) to define "local" grammatical relations, Chomskyan generative theories since 1965 have not limited grammatical relations to the local ones defined in (1).

(1) $\ldots[_{C}\ldots A\ldots B]\ldots$(where A,B are both daughters of C)

Rather, in an attempt to account for the displacement property of human language (HL), these theories have posited various second-order locality relations (or minimality relations) that specify the structural distance beyond which two elements (X,Y) can no longer engage in a syntactic relationship (see (2)).

(2) $\ldots X\ldots Z\ldots Y\ldots$

That is, these second-order locality relations state the structural conditions of Z that must obtain for X and Y to have a syntactic relationship with one another. Many of these second-order locality constraints have been formulated over the years: the A-over-A Principle (Chomsky 1964), the Specified Subject Condition and the Nominative Island Condition (Chomsky 1977, 1980), Government-Binding Theory (Chomsky 1981 and Rizzi 1990), several economy conditions, such as the Minimal Link Condition (Chomsky 1995 and Collins 1997), the Minimal Match Condition (Aoun and Li 2003), and the Phase Impenetrability Condition (Chomsky 2001, 2002), to name a few.

The question I investigate in this book is whether a syntactic theory requires both strictly local relations (operations), as stated in (1), and minimality relations (operations), as stated in (2). More particularly, since the Minimalist Program is the only current syntactic theory that posits minimality relations in addition to strictly local ones, I focus my investigation on whether even the Minimalist Program requires minimality relations. What I hope to demonstrate is that minimality relations are neither conceptually necessary nor empirically justified within the Minimalist Program.

Before outlining my arguments against minimality relations, let me first present a brief overview of the Minimalist Program. According to Chomsky (1995, 2000b), the Minimalist Program is a research program that attempts to define an optimal design for HL by postulating only those assumptions minimally required on conceptual grounds. These assumptions include (i) a grammar generates Logical Form and Phonetic Form pairs ⟨LF,PF⟩ for all sentences in a language L that are interpreted at the conceptual-intentional interface and the sensorimotor interface, respectively; (ii) these pairs are compiled from the feature sets of lexical items by an optimal computational system; (iii) these pairs must have the morphosyntactic features of all lexical elements checked at the interfaces for appropriate interpretability (⟨LF,PF⟩ pairs that satisfy (iii) are said to be "convergent"); (iv) the computational system includes two concatenative, feature-checking operations—(External) Merge, which introduces lexical elements into a syntactic derivation, and versions of Move (including Internal Merge), which relocates elements (for feature-checking purposes) from one position in the derivation to some other position in the derivation; and (v) the computational system also includes Economy Conditions that determine which of the convergent derivations is the optimal (or grammatical) one.

Notice that the above assumptions fall into two quite distinct categories. On the one hand, assumptions (i)–(iii) posit a derivational theory of syntax in which the structural meaning (LF) and the phonetic/physical form (PF) of sentences are compiled out of lexical features; on the other hand, assumptions (iv) and (v) posit a computational system that includes strictly local mechanisms (Merge), nonlocal mechanisms (Move), and minimality mechanisms (Economy Conditions). It is important to recognize the separability of the sets of assumptions discussed above. That is, a commitment to a (derivational) theory of syntax does not necessarily carry with it a commitment to the computational system posited in (iv) and (v).

At this point, let us assume that, as (i)–(iii) suggest, syntactic structure is derived by an optimal computational system set up to check the interface legibility of lexical features. I will show, however, that even if we assume a derivational theory of syntax, there is no motivation for having either nonlocal operations or minimality relations within the computational system.

So, are the Merge operation and the Move operation (or any of its variants) "conceptually necessary"? Some version of a Mergelike, concatenative operation would seem to be. As an operation that forms a larger unit K out of two smaller units A and B, Merge builds the constituents of sentences in accordance with Frege's compositional semantics. It is the Merge operation that combines the verb *likes* and the Determiner Phrase (DP) *linguistics* to form the verb phrase (VP) *like linguistics* and it also combines *likes linguistics* with the DP *Mary* to form the sentence *Mary likes linguistics* (see (3)) and to ensure the appropriate interpretation of this sentence.

(3) a. Merge ⟨likes, linguistics⟩ → likes linguistics
 b. Merge ⟨Mary, ⟨likes, linguistics⟩⟩ → Mary likes linguistics

The conceptual necessity of the Merge operation, then, is that it links syntactic structure to semantic interpretation.

The conceptual necessity of the Move operation, on the other hand, is far less apparent. What the Move operation seeks to explain is how a single grammatical constituent can possess more than one structural relationship in a syntactic derivation—Chomsky (2000b) calls this the "displacement property" of HL. We can observe this property in (4).

(4) a. Chris was fired (Chris)
 'Chris was fired.'
 b. What was Mary reading (what)
 'What was Mary reading?'

(Note: the elements within parentheses are nonovert copies.) In (4a), the DP *Chris* has two grammatical functions: it is both the logical object of the verb *fired* and the grammatical subject of the sentence. In (4b), the *wh*-element is both the object of the verb and a fronted *wh*-operator. The Move operation (or its Remerge variant proposed in Epstein et al. 1998 or its Internal Move variant in Chomsky 2002, 2005) explains the displacement property by allowing syntactic constituents to move from one syntactic position to another. Given the Move operation, sentence (4a) will be derived as in (5).

(5) a. Merge ⟨fired, Chris⟩ → fired Chris
 b. Merge ⟨was, ⟨fired, Chris⟩⟩ → was fired Chris
 c. Move ⟨Chris, ⟨was, ⟨fired, Chris⟩⟩⟩ → Chris was fired (Chris)

Under the Move analysis, the DP *Chris* is moved (or copied) from the object position into the subject position (the copy in the object position is later deleted); hence, this DP can possess more than one grammatical function because it can Merge in one position and then Move subsequently into another position.

It is clearly the case that within a derivational theory of syntax, there must be a derivational explanation for the displacement property, and it is also clearly the case that the Move operation offers one such explanation. But is this explanation an optimal one? Is the Move operation conceptually necessary? Chomsky (2000b) argues that it is. According to Chomsky, a constituent X must move (or be copied) from a position in a YP projection to another position in a ZP projection because the head Z of the ZP projection has morphological features that attract compatible morphological features of X. Movement, then, is necessary to satisfy the morphological requirements of heads of constructions. (In (5), the DP must move into the subject position to have its Case feature, and perhaps other agreement features, checked.) Appealing as this notion of movement may be, it has an unwanted side effect: it permits unconstrained, long-distance movement. That is, the Move operation allows the long-distance displacement of the *wh*-constituent *what* in (6) from its position as object of the verb *reading* to the fronted position in the matrix sentence.

(6) *What does Pat like the woman that was reading (what) (* means ungrammatical)

The ungrammaticality of (6) suggests that the Move operation overgenerates possible grammatical constructions. To reduce the power of the Move operation, minimalist theorists have, over the years, introduced various economy conditions, such as Procrastinate, Shortest Move, Minimal Link Condition, and the Phase Impenetrability Condition, among others, to regulate the output of the Move operation (see Collins 1997, Epstein et al. 1998, and Hornstein 2000 for relevant discussions; see Johnson and Lappin 1999 for a critique of such conditions). Since these economy conditions are output conditions on derivations, they play a crucial role in determining the grammaticality of a derivation—in fact, a derivation will be grammatical if and only if it satisfies economy conditions.

As we can see, if we accept Chomsky's notion of Move as an attract-based (or agree-based) operation, then we must also accept economy conditions. Having to have economy conditions, however, multiplies and complicates the ontological commitments of our theory, and it leaves us with the problem of ascertaining the precise nature of these economy conditions. Over the last ten years of minimalist analysis, so many different economy conditions have come and gone (such as Greed) and so few have remained constant that we seem to be left with the general theoretical need to have economy conditions but without identifying any lasting economy conditions. This state of affairs raises some serious questions about the need to have economy conditions.

So, why are economy conditions conceptually necessary? They are conceptually necessary because without them the Move operation would overgenerate the grammar. And again, why is the Move operation conceptually necessary? Well, it isn't, despite Chomsky's feature-checking arguments to the contrary. Conceptual necessity does not mandate that apparent movement effects be produced by long-distance feature attraction. In fact, there is reason to believe that long-distance attraction is not even possible in a grammar. To see this, consider the following arguments. One thing we know about attractors (heads) is that a distant attractor cannot pull a constituent away from an equally powerful local attractor. Hence, in (7a), the DP *Chris* cannot be attracted by the features of the matrix T(ense) head because of the attraction that the embedded T head has on the DP (so the matrix subject position must be filled by an expletive *it* to check the features of the matrix T head, as in (7b)); and in (8), the T head cannot attract the DP *Sam* away from the Case-assigning preposition *to*.

(7) a. *Chris was believed (Chris) was happy
'Chris was believed was happy.'
b. It was believed that Chris was happy

(8) *Sam gave a book to (Sam)
'Sam gave a book to.'

And yet, minimalist syntax generally derives (9a) and (9b) by moving the DP *Chris* from the Spec of TP2 to the Spec of TP1—that is, at some point in the derivation, the T head to_1 is able to, in essence, attract the DP *Chris* away from the T head to_2.

(9) a. Sam wants [$_{TP1}$ Chris to_1 be believed [$_{TP2}$(Chris) to_2 be wealthy]]
 'Sam wants Chris to be believed to be wealthy.'
 b. Chris was expected [$_{TP1}$(Chris) to_1 appear [$_{TP2}$(Chris) to_2 be happy]]
 'Chris was expected to appear to be happy.'
 c. [Chris was believed [$_{TP2}$(Chris) to_2 be brilliant]]
 'Chris was believed to be brilliant.'

Since the infinitival heads in (9a) and (9b) all have exactly the same features, it should not be possible for one of the heads to attract a DP away from another, more local head, though, as (9c) illustrates, if the two heads have different features, one of the heads might be able to attract a constituent away from another. The fact that the DP *Chris* in (9a–b) moves to the Spec of TP1 when it cannot be attracted to this position suggests that constituent movement does not result from attraction. If this argument is correct, then we must reconsider the long-distance, attraction-motivated Move operation, together with its attendant economy conditions. At a minimum, we should reexamine the Move operation, looking not only at the effect an "attracting" head might have on a constituent K but also at the effects that all heads that come into relation with K might have on K.

A similar argument surrounds the nonovert pronoun PRO. According to Chomsky and Lasnik (1995) and Radford (1997), the features of PRO, including null Case, are checked by an infinitival T-head to, as in (10).[1]

(10) Sam wants [PRO to [(PRO) leave]]

In (10), all the features of PRO cannot be checked by the verb *leave*; therefore, either PRO or some of its features (as Baltin (1995) and Radford (1997) argue) must move to the Spec of TP to have features checked. But if the features of PRO can be exhaustively (and finally) checked by an infinitival head, how/why can a PRO constituent move from infinitive to infinitive in (11), especially if the infinitival heads have the same features? That is, why can't the features of PRO be so exhaustively checked in the most embedded infinitive that it can remain in the infinitive, thereby allowing an expletive to be the subject of the highest infinitive, as in (12)?

(11) Sam wants [PRO to appear [(PRO) to have (PRO) left]]

(12) *Sam wants [it to appear [PRO to have (PRO) left]] (on the it-expletive reading)

The fact that PRO must move from infinitive to infinitive in (11), together with the fact that DP raising must occur in (9a–b), casts serious suspicion on any attract theory of constituent (or feature) movement.

To accommodate data such as those in (9) and (11) within an attract analysis of movement, Chomsky posits an Extended Projection Principle (EPP) feature—a feature that, as McCloskey (2002, 203) notes, "demands that the associated specifier-position be occupied." Under Chomsky's analysis, all the infinitival heads in (9a) and (11) carry an EPP feature, and this feature, which requires the infinitive to have a subject, is sufficiently strong to attract subject DPs away from other heads, including other infinitival heads, which also carry EPP features. Although this analysis can explain the data in (9) and (11), there are reasons to be skeptical about the analysis. For one, as Lasnik (2001) and Epstein and Seely (2002a) argue, the EPP feature is not compatible with Chomsky's (2001) Interpretability Condition, which permits lexical items to have no features other than those associated with properties of sound and meaning. The EPP feature is not a feature interpreted at the sound and/or meaning interfaces; rather, it is a configurational feature that stipulates a structural requirement (the presence of a subject) quite independent of sound and meaning.[2] Because of this, Epstein and Seely (2002b, 82) conclude that the "[EPP] seems to us to undermine the entire Minimalist theory of movement based on feature interpretability at the interfaces." The EPP, then, would appear to lack conceptual necessity and should be jettisoned from minimalist syntax—this is the position advocated by Martin (1999). Another reason for rejecting an EPP feature has to do with its unusual strength. Recall that the Case feature on a head H in English is not strong enough to attract a constituent from another head with a Case feature (see (7a) and (8)). The same is true of the [WH] feature. If a [WH] feature on constituent K is checked by a head H in English, another head with a [WH] feature cannot attract K. We can see this in (13)—the [WH] feature on *where* is checked in the embedded clause (as (13a) shows)) and it cannot undergo further attraction (as (13b) illustrates).

(13) a. Mary wonders [where [John put the money (where)]]
b. *[where does Mary wonder [(where) [John put the money (where)]]]

In English, neither the Case feature nor the [WH] feature of constituent K can undergo multiple attraction to heads carrying the relevant feature. But this is typical of all features in English interpreted at the interfaces: none of them undergo iterated attraction. The EPP feature alone permits such attraction, a peculiar situation that needs to be explained (but never has). What makes the EPP feature even more peculiar is the fact that it has the strength to attract DPs in (9a–b) and in (11), but it does not have the strength to attract the DP *Sam* in (14).

(14) a. *Mary tried [Sam to seem [(Sam) will (Sam) leave]]
    'Mary tried Sam to seem will leave.'
  b. *Mary wanted [Sam to seem [(Sam) left]
    'Mary wanted Sam to seem left.'

Why can the EPP feature of the infinitival head attract PRO from another infinitival head yet cannot attract *Sam* from a tensed head? This is a pressing question because these cases are similar in that the EPP feature in both cases attempts to attract a constituent that putatively has had its features exhaustively checked by another head. In one case (11), the EPP feature successfully attracts PRO; in the other case (14), the EPP feature cannot attract the DP *Sam* (even though this attraction would have no effect on any other the other features of the DP—not the Case feature, not any thematic features, and so on). The fact that the EPP feature is both unique and idiosyncratic makes it highly suspect as a theoretical construct.

And yet, even if we could discover ingenious ways to justify the EPP feature, and thereby preserve Chomsky's attract analysis of movement against data such as those in (9) and (11), we would still find other data that cast doubt on this analysis. Take, for example, anaphors—reflexive and reciprocal pronouns. The prevailing wisdom among linguists is that some anaphors require a local relationship with their antecedents to ensure proper interpretation. Chomsky and Lasnik (1995, 104) characterize this locality as follows: "[It] is plausible to regard the relation between a reflexive and its antecedent as involving agreement. Since agreement is generally a strictly local phenomenon, the reflexive must move to a position sufficiently near its antecedent." For Chomsky and Lasnik, agreement features of a head force the LF movement of reflexives, establishing local relations with their antecedents, as in (15b).[3]

(15) a. Mary was reading a book to herself
  b. Mary herself was reading a book to (herself)

Anaphors exhibiting agreement must have a local antecedent; hence the ungrammaticality of (16), where the anaphor with subject agreement lacks a local antecedent in its agreement domain.

(16) *John believes [himself is clever]

However, anaphors without available agreement are exempted from having local antecedents. This exemption licenses the long-distance anaphors in (17): in (17a) the anaphor does not have any available (subject) agreement domain so it can take a nonlocal antecedent, as can the anaphor in (17b) because Chinese lacks agreement and, therefore, agreement domains.

(17) a. Mary thinks [[pictures of herself] are on display]
b. Zhangsan shuo ziji hui lai (Huang 1982, 331)
Zhangsan say self will come

Despite some explanatory success, Chomsky and Lasnik's attract analysis of anaphors flounders in two ways. First, the Chomsky and Lasnik analysis appears inconsistent with Woolford's (1999) constraint on anaphor agreement—that is, anaphors are incompatible with agreement, unless the agreement is a special form of anaphor agreement. Given Woolford's observations about anaphor agreement, it would be highly problematic if, as Chomsky and Lasnik suggest, anaphors move because they are attracted to/by agreement features with which they are incompatible. Second, if anaphors are attracted by heads with agreement features and if their antecedents must c-command anaphors in their agreement domains at LF, then the anaphor in (18a) will be attracted to the agreement-bearing inflectional head (see (18b)), deriving a logical representation in which the only possible antecedent for the anaphor is the subject of the sentence.

(18) a. Tom sold Dave to himself
b. Tom himself sold Dave to (himself)

In (18b), the DP *Tom* can be the antecedent for the anaphor because it locally c-commands the anaphor at LF. However, the DP *Dave* cannot be the antecedent for the anaphor, not only because this DP fails to locally c-command the anaphor at LF, but also because both the DP *Tom* and the anaphor *himself* already agree with the agreement-bearing inflectional head. Should the DP *Dave* be the antecedent of the anaphor, the DP *Tom* will agree with, and c-command, the DP *Dave*, in violation of Chomsky's (1981) Binding Principle C, which requires all referential DPs to be referentially independent of other DP elements in a sentence. What this means is that a Chomsky-Lasnik attract analysis necessarily rules out (contrary to fact) a permissible antecedent-anaphor relationship in (18a) between the object *Dave* and the anaphor.

The foregoing arguments come to an ineluctable conclusion: attraction based analyses (and long-distance agree analyses) of movement in language are highly costly operations requiring additional ontological commitments to economy conditions, and they are operations that encounter serious empirical limitations. We would do well to approach such analyses with caution. Importantly, this conclusion is also reached by Hazout (2004), who argues, contra-Chomsky's agree analysis of expletive constructions, that *there*-expletive constructions contained within *for-to* constructions (see (19)) cannot possibly be explained by long-distance agreement.

(19) It is unimaginable [for there to be a unicorn in the garden]

As Hazout notes, Chomsky's analysis of *there*-expletive constructions (i) licenses postverbal subject via probe-goal agreement relations between a T(ense)-probe and the subject (the goal) and (ii) the expletive *there*, "because of its formal properties, has no need for structural Case" (338). The fact that (19) includes a well-formed *there*-expletive is problematic for Chomsky's analysis, according to Hazout, for two reasons. First, there is no probe available to license the postverbal subject, and second, the expletive *there*, as the data in (19) and (20) suggest, must be Case licensed.

(20) a. *It is unimaginable [there to be a unicorn in the garden]
b. *[For unexpectedly there to be a unicorn in the garden] is unlikely

Hazout, recognizing that Chomsky strongly appealed to expletive constructions to motivate his use of Agree (as an alternative to Move), concludes that data such as those in (19) and (20) cast serious doubt on Agree-based analyses of long-distance dependencies.

Theoretical Complications

But if feature attraction (or feature agreement) is not responsible for displacement (or movement) effects and long-distance dependencies, what is? To answer this question, we need to, as Chomsky (2001), Martin and Uriagereka (2000), and Chametzky (2003) would advise, situate the question within the metatheory of minimalism. Minimalist syntax, according to Chomsky, should be guided by one overarching metatheoretical principle: it should posit only those elements and operations absolutely necessary for interface interpretation (at sensorimotor interface or at the cognitive-intentional interface) or for optimal design considerations. Since, in our last section, we have focused on movement, especially attraction-driven movement, let's reconsider movement from the perspective of our metatheoretical principle.

Syntactic movement, as Brody (1995, 1998, 2000, 2002) has exhaustively argued, is not at all necessary for interface interpretation. What must be interpreted at the interfaces are syntactic representations. These representations have their phonetic content/information and logical form content/information interpreted in sensorimotor and conceptual-intentional domains, respectively. Given that only representations have interface visibility, any derivations of the representations and any (movement) operations involved in the derivations will lack conceptual necessity at the interfaces. So if movement of any stripe is to be consonant with our overarching metatheoretical principle, the movement must be required for optimal design considerations.

Brody (2002) addresses the role derivation (and movement) might play in the design of a minimalist syntax. For him, there are three possible ways minimalist syntax could incorporate derivations:

> (i) A derivational theory is nonrepresentational if the derivational operations create opaque objects whose internal elements and composition are not accessible to any further rule or operation; (ii) a derivational theory is weakly representational if derivational stages are transparent in the sense that material already assembled can be accessed by later principles (i.e., derivational stages are representations); finally (iii) a derivational theory is strongly representational if it is weakly representational and there are constraints on the representations (weak sense) generated. (pp. 22–23)

Since syntactic movement requires "that material already assembled can be accessed by later principles," any derivational theory with movement will be at least weakly representational. Furthermore, since movement operations have a (representational) input and a (representational) output, all derivational theories with movement are necessarily multirepresentational. (The most overt multirepresentational derivational theories are developed by Chomsky (2001, 2002, 2005), who permits phases within derivations—i.e., CPs and some vPs—to undergo iterated Spell-Out in the sensorimotor interface, and by Epstein and Seely (2002b), who argue that Spell-Out is part of every syntactic operation, requiring interface interpretation of each syntactic object—actually of the representation of the syntactic object—generated by the computational system.) Any syntactic theory that uses movement operations, therefore, is a mixed theory—being both a derivational theory and a (multi)representational theory. A mixed theory, Brody maintains, is less restrictive than either a pure representational theory (PRT) or a pure derivational theory (PDT) because it has the generative power of both PRT and PDT (if we compare the one representation generated by a PRT or a PDT with the multiple representations generated by a mixed theory, we can see that mixed theories do have more generative power than do pure theories). A mixed theory also permits a degree of redundancy between the derived movements and the representational chains interpreted at the interface. Arguments from restrictiveness and redundancy lead Brody to conclude that mixed theories fail as an optimal design for HL.

Given that representations are necessarily required at the interfaces and given Brody's arguments against mixed theories, it would seem that minimalist syntax must be a pure representational theory, one that eschews all derivational operations, including any type of movement operation. For Brody, the design of such a representational model of syntax is maximally simple: it includes no derivational operations—no Merge-type operations and no Move-type operations either—and therefore has no input-output capacity to derive multirepresentations; rather, it consists of a single representation R for any sentence S. A representation R, in this model, will possess hierarchically interrelated lexical items and syntactic chains—all of which will receive LF and PF interpretations at the interfaces. This overstates the situation somewhat because not every representation R will be interpretable at the interfaces; only those satisfying well-formedness conditions will be. Hence, a pure representational theory of syntax will have conditions on representations and conditions on interface interpretability; well-formed representations will have to satisfy these conditions. Representations not meeting these conditions will be filtered out. Simple as such a model may seem, it does have a serious problem. The problem is that although well-formedness conditions can, in principle, reduce the set of representations that receive interpretations at the interfaces by filtering out all ill-formed representations, there is no way to reduce the set of representations that must be processed at the interfaces. That is, every possible representation R can show up at the interfaces because all representations come to the interfaces in the same way, as nonderived, fully formed constructs. A pure representational theory of syntax, then, must be prepared to process any and every representation by either interpreting it or filtering it. The processing demands on such a model are, obviously, astronomical—a fact that threatens the psychological plausibility of representational theories.

Although pure representational theories of syntax cannot constrain the representations that show up at the interfaces, Frampton and Gutmann (2002) argue that derivational theories can constrain these representations. According to them, a derivational syntax should be "crash-proof," generating "only objects that are well formed and satisfy conditions imposed by the interface systems" (p. 90). In a crash-proof syntax, for example, a sentence such as (21) is ruled out simply because it cannot be derived in any way.

(21) *There seems a man to be here

Importantly, since (21) cannot be derived, it will not show up at the interfaces to be processed (though it would in a pure representational theory of syntax). A crash-proof syntax aborts, as early as possible, any derivation that will not meet interface conditions, thereby preventing ill-formed derivational output from ever becoming a full representation that must be processed at the interfaces and maximally limiting the amount of processing dedicated to output that is not interpretable at the interfaces (this accords with O'Grady's (2005, 6) observation that the processing properties of the computation system for HL should "minimize the burden on working memory"). If we assume that an optimal theory of syntax must reduce processing demands at the interfaces as much as possible, then we must conclude that a crash-proof derivational theory meets this optimality requirement much better than does a pure representational theory, which has no way to constrain the representations that must be processed.

We seem to have come to an impasse. On the one hand, Brody's arguments suggest that, on restrictiveness and redundancy grounds, the optimal theory of syntax must be a purely representational theory; on the other hand, from Frampton and Gutmann's observations about derivationally reducing the set of representations that can show up at the interfaces, the optimal theory of syntax would appear to be a derivational theory. As daunting as this impasse may be, we can circumvent it if we return to Brody's analysis of derivational theories. Recall that he notes that there are three possible derivational theories—a weakly representational one, a strongly representational one, and a nonrepresentational one. Brody's arguments against derivational theories are almost entirely arguments against weakly and strongly representational derivational theories, theories that are necessarily mixed theories. What he asserts about nonrepresentational derivational theories—theories with derivational operations that would create opaque objects whose internal elements would not be accessible to further operations—is that they cannot exist because they will have to explain the displacement in sentence such as (2) by interrelating two syntactic elements in terms of movement or in terms of Merge and syntactic chains, and in either case this interrelationship will violate the syntactic opaqueness required of nonrepresentational theories.

(22) what was Pat reading (what)

It is indeed the case that movement will violate the opaqueness requirement. However, as Brody himself recognizes, forming chains need not do so. Brody (2002, 22) remarks that linking chain members, such as *what* and its copy (*what*), can be "taken to be matters of interpretation" that are established in the "interpretive components"—that is, chains do not have to be formed by derivational operations that violate syntactic opaqueness. A derivational theory of syntax, then, could be nonrepresentational if it has no movements, does not derivationally form chains, and generates as its output a single representation that is processed at the interfaces.

If we can develop a nonrepresentational derivational theory, we will be able to satisfy the constraints that both Brody and Frampton-Gutmann place on syntactic theories: we will have a crash-proof, nonredundant theory. This theory, of course, will have to explain the displacement property of HL illustrated in (22), but it will also have to explain the locality constraints placed on displacement—for example, why the verbal *wh*-object can be fronted in (23a), while the prepositional *wh*-object in (23b) cannot be.

(23) a. What did Pat read (what) to whom
b. *Whom did Pat read what to (whom)

Furthermore, our explanation of these locality effects will have to satisfy Fitzpatrick's (2002) two metaconditions on theories of grammar: (i) that these theories use "only simple and independently motivated syntactic objects and relations" (p. 443) and (ii) that they use operations that divorce a "moved" constituent from its "landing site." The former condition raises significant questions about the naturalness of the various economy conditions that have been posited over the years—including the Phase Impenetrability Condition. According to this condition, no syntactic operations or relations, including economy conditions, should be introduced into the design of HL unless they can be justified in terms of their simplicity and their independent motivation. Fitzpatrick's latter condition also strongly circumscribes the design of HL. It prohibits syntactic displacement from being driven by operations that require the satisfaction of "landing-site" conditions, thereby challenging all attract-type analyses of displacement, including Chomsky 1995, 2001, 2002, 2005; Richards 1999, 2001; Lasnik 1999b; and many others.

A Proposal

I can now return to the question posed at the beginning of the last section: if feature attraction is not responsible for displacement (or movement effects), what is? I will argue in this book that displacement does not come from an operation that is either long-distance or attraction-based; rather, it comes from an operation that is local and repel-based. When a constituent X is having its features checked by a head Y, if there are features of X that are not compatible with Y and cannot be checked by Y (for example, the Case feature on the DP in (24a) is not compatible with the passive participle *fired*, which lacks a Case-checking feature), these features SURVIVE the checking operation and their incompatibility with Y repels them to the next c-commanding head Z for further feature-checking, as in (24b).

(24) a. Merge ⟨fired, Chris⟩ → fired Chris
b. Repel ⟨Chris, ⟨was, ⟨fired, Chris⟩⟩⟩ → Chris was fired (Chris)

In (25a–c), once the DP *Chris* is in the Spec of TP2, its features are checked by the head T (to_2); the T head checks whatever features it can. However, the DP also has some features—for example, a Case feature—that are incompatible with the infinitival head.

(25) a. Sam wants [$_{\text{TP1}}$Chris to_1 be believed [$_{\text{TP2}}$(Chris) to_2 be wealthy]]
b. Chris was expected [$_{\text{TP1}}$(Chris) to_1 appear [$_{\text{TP2}}$(Chris) to_2 be happy]]
c. Chris was believed [$_{\text{TP2}}$(Chris) to_2 be brilliant]

The incompatible features SURVIVE in the derivation and they force the head T to repel the DP. The DP is forced to the next structurally higher head to have its features checked. If all the features that survive the previous checking operation are appropriately checked by this head, then the DP will have no features that SURVIVE; as a consequence, the DP will not be repelled away from the head, and it will cease to move any more (as in (25a,c)). On the other hand, if the head does not check all the remaining features that must be checked, the DP will be repelled again, and will continue to be repelled until all its checkable features are checked (this happens in (25b)). Importantly, not only does this local theory of repel-driven displacement account for (25a–c), but it also explains (26a). Notice that in (26a), the DP *Chris* has all its features, including its Case and Agreement features, checked in the embedded TP. Since the DP has no features that are incompatible with the T head, the DP will not be repelled out of the embedded TP—that is, the DP cannot move beyond the embedded TP, as (26b) attests.

(26) a. [$_{\text{TP}}$ Chris was happy]
b. *Chris was believed [$_{\text{TP}}$ (Chris) was happy]

The foregoing discussion anticipates some of the empirical advantages of a local repel analysis of displacement over the long-distance attraction analysis (I will discuss these advantages at length in the next chapter). However, my analysis of displacement as a strictly local phenomenon is not just empirically preferable to the attract analysis, it is also conceptually preferable. Because the Repel operation is a strictly local operation, displacement, as the effect of a syntactic operation, is always strictly limited to the next higher syntactic projection; consequently, my Repel analysis does not require any of the economy conditions (or minimality relations) needed to limit the attract-based Move operation (this accords with Fitzpatrick's first metacondition on syntactic locality). Furthermore, a repel analysis makes displacement the result of a local head Y pushing a constituent X to the next head domain; hence, this analysis divorces displacement from any notion of landing site (therefore it satisfies Fitzpatrick's second metacondition on syntactic locality, too).

Although a repel analysis of displacement radically simplifies the grammar by eliminating economy/minimality conditions on movement such as Shortest Move or the Minimal Link Condition, it does not simplify the grammar enough. The fact that my repel analysis of displacement is still a movement analysis makes it a mixed (representational and derivational) theory in Brody's (2002) terms. As a mixed theory, it should be discarded as an optimal theory of grammar for reasons discussed in the previous section.

We can preserve the conceptual advantages of our repel analysis, and still prevent it from becoming a mixed theory, if we incorporate a version of Chomsky's (1995) Numeration N into our analysis. For Chomsky, syntactic derivations begin by placing lexical items from our internal lexicon into a Numeration and the derivations proceed by "recursively construct[ing] *syntactic objects* from items in N and syntactic objects already formed" (p. 226). Let us assume we have a Numeration N, but let us also assume that lexical items enter a derivation by being copied into the derivation from N and are, therefore, still present in N. The two lexical items *fired* and *Chris* in (1a) must be in a Numeration (N = {*fired*, *Chris*, *was*, *T*}) before they can be merged. (Representation (24a) is repeated below.)

(24) a. Merge ⟨fired, Chris⟩ → fired Chris

These two lexical items will continue to be in N after they are merged. Furthermore, any feature of a lexical item LI checked in the application of an operation, such as Merge, will be checked both in the derivation and in N. If all the features of an LI are checked via some operation, all the features of the LI will be checked off in N. At this point, the LI will become derivationally dormant: it will not be able to participate in (be copied into) further syntactic operations because it will not have any features to be derivationally processed. On the other hand, if some of the features of an LI are not checked via a syntactic operation, these features will SURVIVE. That is, these features will remain unchecked in the Numeration, and the LI will remain derivationally alive and can Remerge in the derivation. All features of every LI in a derivation must be checked for interface compatibility. Should a derivation terminate with an LI in N that has an unchecked feature, the derivation will abort (or stall)—it will not be processed by the interface systems. We can see how this works in (24a). The lexical item *Chris* in (24a) has (at least) a Case feature and a thematic-role feature (Adger (2003) suggests an alternative to thematic-role feature-checking; he proposes that the c(ategory)-feature of the LI must be checked). The thematic-role feature is checked by the role-checking verb *fired*. Though the passivized verb can check a thematic role, it cannot check Case. The LI *Chris*, then, has its thematic role checked off in the derivation of (24a) and in N, but the Case feature of *Chris* SURVIVEs in N. Having an unchecked feature in the derivation is not problematic; having an unchecked feature in N is. The LI *Chris* will have to Remerge from N into the derivation if it is to have all of its features appropriately checked by the termination of the derivation. To capture the essence of Repel, a land-site indifferent operation that pushes elements along from head to head, let us assume that Remerge is an automatic operation, one that remerges an already-merged element with the next c-commanding head in compliance with O'Grady's (2005) Efficiency

Requirement mandating that dependencies be resolved at the first opportunity. (I will offer empirical support for automatic Remerge in the next chapter.) Under this assumption, the LI *Chris* will have to automatically remerge once the next available head *was* is merged, as in (27).

(27) a. Merge ⟨was, ⟨fired, Chris⟩⟩ → was fired Chris
b. Remerge ⟨Chris, ⟨was, ⟨fired, Chris⟩⟩⟩ → Chris was fired Chris

Since the head *was* is not a Case checker, the LI *Chris* will continue to have an unchecked Case feature, which will SURVIVE in N. Consequently, this LI will Remerge yet again when the next available head T merges into the derivations, as in (28).

(28) a. Merge ⟨T, ⟨Chris, ⟨was, ⟨fired, Chris⟩⟩⟩⟩ → T Chris was fired Chris
b. Remerge ⟨Chris, ⟨T, ⟨Chris, ⟨was, ⟨fired, Chris⟩⟩⟩⟩⟩ → Chris T Chris was fired Chris

The head T can check all the previously unchecked features of the LI *Chris* in the derivation and in N, so *Chris* will not be able to be remerged into the derivation.

A terminated derivation with no unchecked features in its Numeration becomes a single representation submitted via Chomsky's (2002) Transfer operation to the interface systems for interpretation. As Brody (2002) proposes, the linking of chain members will take place in the interpretive components. Hence the three (re)mergings of the LI *Chris* in (5b) will be linked in the interpretative components. This chain, then, will have one of its links (the one with checked morphophonetic features) interpreted in the sensorimotor interface and it will have its link(s) with checked semantic features interpreted in the conceptual-intentional interface. In the *Chris*-chain ⟨*Chris*, *Chris*, *Chris*⟩, the highest (last remerged) link will have its phonetic features interpreted, the lowest link will receive a semantic interpretation, and the middle link (which had no features checked) will not be interpreted by either interface system. Therefore, the phonetic interpretation of (28b) will be (29), but (28b) will be semantically interpreted akin to (27a).

(29) Chris was fired

(It is important to note here that I will expand my abbreviated overview of SURVIVE and Remerge in the next chapter.)

The conceptual advantages of the theory I outline above are as follows. First, this theory is simpler than any movement-based theory or any agree-based theory (see Chomsky 2001, 2005) because it does not require economy (or minimality) conditions to restrain the power of the Move operation, the Attract operation, and/or the Agree operation. Second, this theory satisfies both of Fitzpatrick's (2002) metaconditions on locality because it uses only simple operations and relations and because its use of automatic Remerge divorces displacement effects from landing sites. Third, this theory meets Brody's (2002) requirements that a theory of grammar not be a mixed theory and that an optimal theory of grammar have only a single representation that is interpreted at the interfaces. Fourth, this theory is, as we have argued all optimal theories must be, a nonrepresentational, derivational theory (because the derivational operations Merge and Remerge, which map elements in a Numeration into a derivation, create opaque syntactic objects whose internal elements are not accessible to further derivational operations). Fifth, this theory accounts for the displacement property of HL, not in terms of displacement operations—a circular explanation—but in terms of local Merge operations (nondisplacement operations). And last, but not least, this theory is crash-proof, in the Frampton-Gutmann sense. That is, the computational system described above will generate only well-formed syntactic objects (it cannot derive representations for syntactic objects such as those in (30)) and it requires no mechanism to filter its end-product derivations, which are the representations interpreted at the interfaces.

(30) a. *Chris was believed (Chris) was happy
b. *Sam gave a book to (Sam)

Our computational model, then, satisfies the prevailing optimality conditions placed by minimalist theorists on the design of HL.

Some Minimalist Assumptions

Radically reformulating minimalist syntax along the lines I propose to do here opens the way to questioning many of the underlying assumptions of minimalism. With the hope of bringing additional clarification to the proposal I put forth in the previous section, I will examine some of the core assumptions of minimalism.

One of the most basic assumptions of derivational minimalism is, as Chametzky (2003, 220) states, that it "takes lexical items composed of features to be what is given to syntax." Such a derivational theory of syntax uses syntactic operations to map the features of lexical items LI[F] onto the sensorimotor and conceptual-intentional interface systems. On the surface, postulating an LI-to-interface mapping seems uncomplicated.

However, it is not. Lurking within this assumption is another assumption, one that appears to position lexical items, and our mental lexicon in general, somewhere in our brains separate from (and external to) the interface systems. We can see this assumption "lurking" in Chomsky's (1995, 2000b) claim that, in the course of a derivation, features are checked for interface compatibility, as if these features could be intrinsically incompatible with (and extraneous to) the interface systems. We can also see this "lurking" in Adger 2003, where the general architecture of the HL computation system looks as follows:

NUMERATION → SYNTACTIC OBJECTS → INTERFACES

For Adger's design of HL systems (as well as for the design of most minimalists' HL systems), the lexicon and any Numeration extracted from the lexicon are apparently located somewhere displaced from (and isolated from) the interfaces. But there are reasons to believe that lexical items (and their features) are in fact intrinsically compatible with the interface systems. According to Chomsky (2001), the lexicon satisfies the Interpretability Condition (31).

(31) *The Interpretability Condition (IC)*
Lexical items (LIs) have no features other than those interpreted at the interfaces, properties of sound and meaning.

If LIs have only interface-interpretable features, as suggested by the IC, then these features need not be checked for interface compatibility because they are intrinsically compatible with the interface systems. And this indeed seems to be the case. Take a word such as *she*, which has the following features: [third person, female, singular, nominative] together with some phonetic features. Quite independently of any syntactic checking operations, we know how to produce this word phonetically and what it means semantically directly from the features of the lexical item. The word comes out of the lexicon completely prepared to be interpreted at the interfaces, as does each and every word in our lexicon.[4] There appears, then, to be an intrinsic relationship between lexical items and the interface systems, one that could be most simply explained if the lexical items and their interface-compatible features are located within the interface systems themselves. Furthermore, the fact that we learn words—their sounds and their meanings—at (and through) the interfaces adds strength to the interrelatedness of our mental lexicon and the interface systems. That is, children must use the conceptual-intentional and sensorimotor interfaces to learn the meanings and the physical forms of new words; hence, lexical items must be accessed through, as well as sorted and stored by, interface mechanisms.

Stroik and Putnam (2005) develop the above line of argument in additional detail, proposing that the lexicon is not extraneous to the interfaces—rather, it is actually contained within the interfaces (possibly residing where the interfaces intersect). They argue against the assumption, shared by Chomsky (1995, 2005) and Epstein, Thráinsson, and Zwart (1996), that the lexical items in the lexicon consist of an array of lexical features $\langle F_1, \ldots F_n \rangle$, which includes semantic SEM and phonological PHON features interpreted at the interfaces, as well as SYN features (such as Case features) that are not compatible with (or interpreted by) either interface, but do play a role in building syntactic derivations. According to Stroik and Putnam, "uninterpretable" SYN features are not possible because they are simply not learnable. Since SYN features are not interface-interpretable features, we cannot use the interfaces to identify or compute these features directly. So how do we learn them? How do we know that these features can (or must) show up in a given lexical array? Well, it would seem that we must identify them when we decipher (i.e., process) syntactic structure. In other words, we must learn these features syntactically, extracting them from their relationship with lexical heads that bear counterpart "interpretable" features and subsequently adding them, in the course of our syntactic processing of sentences, to the appropriate lexical array. This sort of feature learning, however, is very dubious. We can see this if we examine Case features. If the Case feature for a nominal element Y must be syntactically extracted from Y's relationship with some head H that bears Case as an interpretable feature, then Case cannot be learned in most languages, including English, because most languages do not generally have heads with Case-interpretable interface features (notice that the prototypical Case assigners in English—verbs and prepositions—lack SEM and PHON interpretable Case features). The foregoing argument suggests that Case, as a SYN feature, cannot be learned syntactically. At this point, we appear to be left with the problem of having to admit that Case (and arguably all other SYN features) cannot be learned within the computational system of HL. From this Stroik and Putnam conclude that SYN features cannot exist—that is, all features must either be SEM features or PHON features (they support this conclusion by noting that the "uninterpretable" features of the Old English word *dagum* are morphophonetically expressed in the *-um* ending). All features, consequently, are interface-compatible features. If all lexical features are interface-compatible and interface-identified features, then this is tantamount to saying that lexical features are interface features and that the lexicon is contained within the interfaces.

Let us assume, therefore, that our lexicon is encapsulated within our interface systems (a position that Jackendoff (2002) also advocates). Under this assumption, we have no need to map features of lexical items onto the interfaces because these lexical items and their features are already contained within the interfaces. Having syntactic operations check the interface compatibility of features would seem to lack conceptual necessity and should be suspect as minimalist operations. So if syntactic operations are not designed to check lexical features for interface compatibility, what are they designed to do? My sense is that syntactic operations do not check the interface compatibility of lexical features; rather they check for concatenative integrity—how well formed a derivation is (how crash-proof it is, in Frampton and Gutmann's sense). A well-formed derivation will exhaustively build its form and meaning, as interpreted at the interfaces, out of the interrelationships of all the features of all its lexical items. Each and every lexical feature contributes something to the concatenative integrity of a derivation by linking appropriately to the features of other lexical items, thereby generating a derivational string that is interpretable at the interface. Syntactic operations serve to link these features by placing them in feature-matching domains where the features can be cross-checked for compatibility. Hence, syntactic operations are designed to build crash-proof syntactic objects from lexical features by linking compatible features together.

All the features of all the lexical items in a Numeration must contribute to the concatenative integrity and the interface interpretation of a derivational string. If even one feature on a single lexical item fails to match in a derivation, the derivation will be incomplete and will not be sent to the interfaces to be processed/interpreted. So, for example, if the Number feature of the lexical item *she* does not match the appropriate number feature of another Number-bearing lexical item, the derivation will stall, as in (32a). Or if the Case feature of *she* fails to find a matching, cross-checking feature on a Case-assigning lexical head, the derivation will stall, as in (32b).

(32) a. *She are happy
b. *Pat read the book to she.

Furthermore, since our syntactic theory generates a single representation for each completed derivation and since all lexical features checked in the course of a derivation are interpretable at one of the interfaces, all lexical features must appear in the representation. That is, no features can be deleted in the course of a derivation. Nor is any feature un-Interpretable, despite counterclaims by Chomsky (1995, 2005), Groat (1999), Freidin (1999), Epstein and Seely (2002a, 2002b), Adger (2003), and many others. For Chomsky, certain features, such as the Case feature on DPs, are not interpreted at LF; therefore, they are un-Interpretable features, and they must be deleted from the derivation before the derivation receives an LF interpretation or the derivation will crash because the conceptual-intentional interface will not be able to assign meaning to them. This analysis is impossible in our derivational theory. If the Case feature on the word *she* is deleted at some point in a derivation, this feature will not appear in the derivation-final representation for interface interpretation; consequently, the Case feature will not be phonetically realized. We would be left with no way to phonetically discriminate the Nominative form *she* from the Objective form *her*. We do, however, make a phonetic discrimination between these two forms, so the Case feature must survive the course of a derivation to be available for PF interpretation. Importantly, my argument against un-Interpretable features is supported by Martin (1999, 39), who observes that "insofar as we think that [the computation system of human language] may be perfect or optimal in some serious sense, the existence of features not interpreted by interface systems is surprising."

What about deleting duplicate features and/or duplicate copies of constituents? Before answering this question, let me make a few remarks about the copying operation in minimalist syntax. Chomsky (2005) allows two types of copying operations. One of these occurs in the formation of the Numeration, where lexical items from the lexicon are copied in the Numeration. This copying operation permits both copies to be present in their respective lexical buffers, with no subsequent deletion operations applying to either of these copies. The second copying operation Chomsky uses is encoded in his External and Internal Merge operations. In External Merge, an element of the Numeration is copied and simultaneously merged into the syntactic derivation; and the original copy in the Numeration is deleted. Similarly in Internal Merge, the copying operation makes a copy of an already-merged constituent and (simultaneously?) Merges it elsewhere in the derivation; and the copied constituent is eventually deleted. Notice that this computational model of grammar requires a copying operation, a deletion operation, and some mechanism to determine whether the copying operation must be followed by a deletion operation. The model I am developing in this book is much simpler. In my model, Numeration formation, Merge, and Remerge all are versions of a single copying operation. No other mechanisms are required in the computation of syntactic derivation—no deletion operations and no mechanisms to determine whether deletion should take place after a copying operation. So, should the design of HL include deletion operations in addition to copying operations? That is, in derivations such as (33)—where (33b) is derived from (33a) to check the Case feature of *Chris*—should it be permissible to delete the lower copy of *Chris* at some point in the derivation, thereby deriving (33c)?

(33) a. [was [fired Chris]]
b. [Chris [was [fired Chris]]]
c. [Chris [was [fired]]]

Chomsky (1995), Fox (2000), Lasnik (1999b), Nunes (2001), Kural (2005) and others, assume some version of copy deletion (or chain reduction deletion), despite Nunes's observation that for economy purposes derivations with deletion operations such as (33c) as more costly than derivations without these operations (like (33b)). Although deletion is costly, Chomsky, Lasnik, and Nunes allow this operation to apply to (33) to secure a proper PF interpretation for (33), an interpretation that has only one copy of *Chris* interpreted phonetically—and this copy is the highest copy. The motivation for applying a costly deletion operation to (33b) is that the two instances of *Chris* in (33b) are nondistinct copies, only one of which will be interpreted at each interface; hence one of the copies is unnecessary at each of the interfaces. Given that only one copy of the DP *Chris* is interpreted at each interface, we arguably still need to have both copies because different copies of the DP are interpreted at the interfaces: the copy in (33a) is interpreted at the conceptual-intentional interface and the copy in (33c) is interpreted at the sensorimotor interface. This is all the more necessary if, as I have maintained, only one representation (the final derivation) shows up at the interfaces for appropriate interpretation: hence if both copies of the DP *Chris* must be eventually interpreted at the interfaces, then both of these copies must be present in the final derivation, as in (33b). Furthermore, recall that syntactic operations do not check the features of individual lexical items; rather, these operations check the legitimacy of concatenations (the way the lexical items are strung together) by cross-checking (matching) features of syntactic objects. And when the final derivations are processed at the interfaces, what is interpreted is not the meaning and form of individual lexical items, one at a time. What is interpreted are the concatenations. That is, the DP *Chris* cannot be interpreted at the conceptual-intentional interface independently of its relationship to the verb *fired*, which checks the thematic feature of the DP, and the DP *Chris* cannot be interpreted at the sensorimotor interface independent of its relationship to the T head that checks its Case and Agreement features. Since each of the two copies of the DP *Chris* engages in concatenating relationships and since both of these relationships must be interpreted at the interfaces, both of the copies must be present in the final derivation, (33b). When derivation (33b) reaches the interfaces, the higher copy will have its Case and Agreement features interpreted at the sensorimotor interface, while ignoring the features of the lower copy, and the lower copy will have its thematic feature interpreted at the conceptual-intentional interface, while ignoring the features of the higher copy. None of the copies of *Chris*, then, can be deleted. I would like to make one final observation about copies, specifically, about the (non-) distinctiveness of copies. If we follow Chomsky (2001) in assuming the Inclusiveness Condition, we must conclude that every copy of a syntactic object must have the same features because no new features can be accreted during a derivation.

Inclusiveness Condition
No new features are introduced by the computational system of human language.

Should the copies of syntactic objects be determined solely by the types of features carried by these objects, all copies will, in accordance with the Inclusiveness Condition, have nondistinct features. However, the fact that syntactic operations affect features means that each "copy" will differ from other copies in terms of the ways their features have been affected (i.e., checked for proper concatenation). The copy of *Chris* in (33a) differs from the copy of *Chris* in (33c) in that the first copy has its thematic feature affected, but not its Case feature or agreement features, while the second copy has its Case and agreement features affected but not its already-affected thematic feature. From the perspective of feature affectedness, the copies of a syntactic object are distinct and, therefore, none of them should be available for duplicative (and recoverable) deletion. What this means is that the syntactic derivation for (34a) is (34b); similarly, the derivation for (35a) is (35b).

(34) a. Chris was fired
b. Chris was fired Chris

(35) a. Who did Pat think would be elected
b. Who did Pat think who would be elected who

Of note in (34b) and (35b) is that none of the copies of *Chris* or of *who* are deleted (or placed in parentheses to denote deletion).

Some support for the approach to copying and deletion that I present above comes from Bruening's (2006) analysis of *wh*-copy constructions. Bruening argues that some languages, such as German and Passamaquoddy, phonetically express more than one copy of a *wh*-chain, as is illustrated in (36) and (37).

(36) *Wovon* glaubst du, *wovon* sie träumt? (from Felser 2004)
Of what believe you of what she dreams
'What do you believe she dreams of'

(37) *Tayuwe* kt-itom-ups *tayuwe* apc k-tol-I malsanikuwam-ok
When 2-say-DUB when again 2-there-go store-LOC
'When did you say you're going to go to the store'

For Bruening, the *wh*-constituents in (36) and (37) are copies; however, they are distinct copies. We can see this in (38), where, as Bruening points out, the higher copy spells out only the formal [WH] feature, while the lower copy spells out a full phonetic articulation of the phrase.

(38) *Wen* denkst du *wen von den Studenten* man einladen
Who think you who of the students one invite
sollte (from Fanselow and Ćavar 2001)
should
'Which of the students do you think one should invite?'

However, under Bruening's analysis, the matrix *wh*-element *wen* is not extracted from the embedded *wh*-phrase *wen von den Studenten*. Instead, the entire phrase is copied in the matrix CP, but only the *wh*-feature of the phrase is produced phonetically. That is, (38) is produced from the derivation in (39).

(39) $[_{CP}$ wen von den Studenten . . . $[_{CP}$wen von den Studenten . . .]]

Importantly, (38) is not formed from (39) by deletion any of the copies of the *wh*-phrase; instead, all the copies are derived as full-phrase copies, though they are phonetically spelled out in variant ways. That Bruening's analysis necessitates the presence of (distinct) copies of constituents within a syntactic derivation squares nicely with the syntactic model I have proposed.

Not only does our Remerge proposal simplify the grammar by not allowing feature or copy deletion, it also obviates the distinction between overt and covert syntactic operations. According to Uriagereka (1999), Chomksy (2001, 2002), Lasnik (1999b), Richards (2001) and others, some operations apply to a derivation, in the overt syntax, prior to Spell-Outand their output is interpreted at the sensorimotor interface; and other operations apply after Spell-Out—these operations, which affect only those features relevant for an LF interpretation, are said to apply in the covert syntax because no evidence of the operations is apparent in the PF interpretation. Given my previous arguments against Spell-Out and against multirepresentational theories of grammar, it is not possible to have two different types of syntactic operations that apply at different points in the derivation. A derivation, under my proposal, uses two syntactic operations—Merge and Remerge—to continually build structure until it either comes to an end product (a derivation in which all concatenative features of all the lexical items have been appropriately checked) or aborts because at least one of its lexical items has a noncheckable feature. Since Merge and Remerge apply to features without regard to where the features will eventually be interpreted, the two operations are blind to the content of the input and apply indiscriminately, freely intermixing the checking of sensorimotor features with the checking of conceptual-intentional features. In this way, my proposal is akin to the "interleaving" proposals of Bobaljik (1995), Groat and O'Neil (1996), and Pesetsky (1998), proposals that generate a single derivational output interpreted at the interfaces, and that also do not derivationally segregate overt syntactic operations from covert syntactic operations. The conceptual advantage of having operations intermix, or interleave, types of features is that operations can be reduced to structure-building operations and need not include both structure-building (overt) operations and structure-remodeling (covert) operations, as is required of the feature-segregation models in Uriagereka, Chomsky, Lasnik, and Richards. The latter models must first compute a structure and then return to this structure to recompute it in terms of "covert" features; intermixing models avoid devoting any processing time to such expensive recomputations of structure.

As we can see from the foregoing discussion, our derivational model of minimalist syntax takes a radically new look at the relationship between lexical items and syntactic derivations. By locating lexical items, and their features, already within the interfaces prior to participating in any derivations, we are forced to recognize that derivations are not, as Chomsky (2001, 2002) suggests, involved in checking the legibility of lexical features. Lexical features are inherently legible at the interfaces, so this legibility has no need to be checked. What syntactic operations derivationally check are not lexical features, but concatenation relations. That is, derivations are concerned with the legibility of larger-than-lexical-item syntactic objects (SOs), not the legibility of the atomic features. It is the SOs that are interpreted at the interfaces. Any operations that might obfuscate, or improperly delay, the concatenation relations of SOs will prevent the SOs from being interpreted legitimately. Hence, feature-deletion operations and copy-deletion operations must be rejected because they disrupt the interpretation of SOs formed in a derivation, and feature-segregation operations must be rejected because they build SOs that are not legible at the interfaces until they are rebuilt by covert syntactic operations.

Organization of This Book

The core arguments of my SURVIVE and Remerge derivational theory of syntax are presented in chapter 2 ("The SURVIVE Principle"). In this chapter, I use multiple-wh constructions and superraising constructions to demonstrate that the displacement property of HL follows from the SURVIVE Principle, a principle that compels a head H to "repel" XP constituents possessing features that are incompatible with the features of H. As I argue at length in chapter 2, the SURVIVE Principle not only offers us interesting, empirically adequate analysis of various displacement phenomena, but more significantly, it also provides an optimal solution to the design problem of HL. In chapter 3 ("Some *Wh* Puzzles"), I extend my analysis. I show how the SURVIVE Principle and Remerge conspire to explain a range of *wh*-phenomena, including superiority effects and *that*-trace effects.

2 The SURVIVE Principle

Introduction

In the previous chapter, I proposed that an optimal design for human language will use two, and only two, syntactic operations to build syntactic structure—Merge and Remerge. These two operations introduce syntactic objects (SOs) into a derivation. Merge does so selectively by matching the thematic features of an SO with the thematic requirements of a (predicational) head. Remerge does so unselectively, automatically reintroducing, from the Numeration, any already merged SO with at least one unchecked, surviving feature (if the SO is merged in XP, it will remerge in YP—the phrase that immediately dominates XP). Importantly, neither Merge nor Remerge has any derivational look-back or look-ahead capabilities, and neither of them needs to be constrained by economy (or minimality) conditions. The derivational simplicity of the Merge and Remerge operations together with the (ontological) exclusion of minimality conditions makes my proposed design for HL a design that is as economical as possible.

My arguments for Merge and Remerge come from conceptual necessity. Building on Brody's (1995, 2002) representational requirements on a theory of syntax, on Epstein, Groat, Kawashima, and Kitahara's (1998) and Epstein and Seely's (2002b) derivational requirements, on Frampton and Gutmann's (2002) crash-proof requirements on derivations, and on Fitzpatrick's (2002) metaconditions on theories of locality, I argue that the only theory of minimalist syntax that can both account for all the syntactic properties, especially the displacement property, of HL and still satisfy the various requirements on a theory of syntax mentioned above is one that uses nothing but strictly local syntactic operations. I also demonstrate that we can optimally constrain a theory of minimalist syntax by reducing syntactic operations to Merge and Remerge.

In this chapter, I continue to develop my arguments in support of my proposal, supplementing the conceptual arguments I make in the first chapter with empirical arguments.

On SURVIVE and Remerge

Minimalist analysis, according to Chomsky (2000a, 2001), seeks an optimal solution to the design of the human faculty of language, *FL* Any such solution will provide the "specifications" that permit a language L to generate expressions EXP (EXP = ⟨PHON, SEM⟩) so that these expressions are accessible to performance systems external to *FL*—in particular, so that PHON will give "instructions" accessible to sensorimotor systems and SEM will give "instructions" accessible to conceptual-intentional systems of thought. Further, any such solution will be "optimal" if it maximally simplifies the computation of EXPs able to satisfy the interface conditions imposed by the performance systems.

Chomsky (2001) assumes that to generate the EXPs of any language L, the human faculty of language must possess, as part of its design, the following components:

(1) a. A set of features
  b. Principles for assembling features into lexical items
  c. Operations that apply successively to form syntactic objects SOs of greater complexity; these operations comprise the computational system for human language (CHL)

The complexes of linguistic features F that have been assembled from (1) will be the EXPs of L usable by the performance systems.

The design of *FL* outlined in (1) raises several important questions, some of which concern the empirical adequacy of the design—for example, are syntactic objects generated bottom up, as suggested in (1), or top down (see Phillips 1996, 2003, for a discussion of this issue), and can the syntactic objects compiled from elementary features actually be processed by performance systems in ways that capture the form-meaning correlations of the EXPs in a language? These sorts of questions are currently being investigated (empirically), and answers to them will certainly have effects on the directions minimalist analyses take in the future. At present, it is worth pursuing the design of *FL* stipulated in (1) to test its design limits.

Let us grant, at this point, that (1) provides a "good" (i.e., empirically adequate) solution for the design of *FL*. If so, under what conditions, we might ask, will it be an "optimal" solution? It seems to me that if (1) were to be an optimal solution to the design specifications of *FL*, it would have to maximally simplify (i) the set of features permissible in *FL*, (ii) principles for assembling lexical items (LIs), and (iii) the operations of CHL. Chomsky (1995, 2000a) has addressed (i) and (ii), identifying various interpretable and uninterpretable features that form LIs, and I will add little to this discussion beyond the remarks I made in the previous chapter on the impossibility of uninterpretable features. Instead, I will investigate how we can maximally simplify the operations of CHL.

The operations of CHL, as (1) suggests, combine LIs to form SOs, or they combine SOs with other LIs or SOs to form more complex SOs. Given (1), it should be possible to define SOs recursively with the use of Merge operations, along the lines specified in (2).

(2) a. LIs are SOs.
b. If A and B are SOs, then Merge⟨A,B⟩ is an SO.
c. Nothing else is an SO.

If Merge operations formed SOs as in (2), these operations could licentiously string together any series of LIs and all such strings would be well formed SOs in a language. Obviously, this would terribly overgenerate the SOs permissible in any language. We can conclude, then, that the combinatory power of Merge must be reduced. That is, only certain types of combinations will produce well-formed SOs. That only a subset of possible combinations is well formed can be seen in (3).

(3) a. this child
b. *this children
c. *these child
d. these children

In (3), the only combinations that are well formed are those in which some of the features of the determiner agree with some of the features of the noun. As a working hypothesis, let us assume that Merge requires feature agreement of a sort. If this is correct, then (2) must be reformulated as (4).

(4) a. LIs are SOs.
b. If A[F] and B[F_c] are SOs, then Merge⟨A,B⟩ is an SO.[1]
c. Nothing else is an SO.

(4b) states that A and B can combine only if they share at least one atomic feature together with its value [F] and that the feature [F] of B is specified as a feature-checker, as [F_c]. Requiring B to be marked with the

$[F_c]$ feature rather than just the [F] feature is necessary to prevent an LI from licitly combining with itself—which could occur if Merge merely required two LIs to share features, since an LI necessarily shares features with itself. Condition (4b), then, prohibits the LI *the* from combining with itself to generate *the the*.

Although delimiting the Merge operation to combining two SOs that share a feature does constrain the generative power of Merge a little, it does not constrain the generative power enough. An SO with feature [F] could, in principle, keep combining with any other SOs that possess that feature; hence strings such as (5) could be generated.

(5) this this this this child

To prohibit the recursion in (5), we must disallow some features from participating in more than one Merge operation. One way to do this is, as Chomsky (1995, 2000a) suggests, to deactivate a feature F in A[F] after it has combined with F in $B[F_c]$.[2] Hence, the Merge operation may involve more than merely uniting two SOs. It will also involve checking the features for appropriate unification, as required in (4b), and it may involve the deactivation of features perhaps, as Chomsky notes, to ensure their legibility—or legibility of the checking operations themselves, as I argue in chapter 1—once they enter the performance systems. The relationship between checking and deactivation, according to Collins (1997), is dependent on the LF interpretability of the feature being checked. Collins proposes that all features must be checked; however, only features invisible at LF—such as the Case features of an N, and the phi-features and Case features of V or T—will be deactivated (or canceled). According to Collins, features that are interpretable at LF, on the other hand, will not be deactivated. Hence, categorial features (+/−V, +/−N, D, T, etc.), the *wh*-feature, and the phi-features of N must be checked but cannot be deactivated. The plausibility of Collins's analysis crucially depends on separating Interpretable features from Uninterpretable features, and LF features from PF features, and it depends on canceling/deleting features. Recall that the previous chapter argued against such separations and deletions, so we cannot adopt Collins's analysis completely. However, let's follow Collins in assuming that checked features are deactivated in the remainder of a derivation.

The Merge operation will then be a checking and perhaps a DEACT(ivation) operation. That is, any SO A that enters the computation of a sentence has a set of activated features $\langle F_1, \ldots, F_n \rangle$ all of which must be checked and, perhaps, deactivated for interface compatibility;

hence A must combine with other SOs able to check the features F_i for agreement. Significantly, even though A may combine with B to check/deactivate some of its features, B might not be able to check/deactivate all of the active features of A. Consequently, A will have to combine with other SOs until all the features of A that must be checked/deactivated are properly checked/deactivated. If A emerges from the computation with an unchecked or undeactivated feature, then the entire derivation involving A either will crash because it will have information not readable by the performance systems, as Chomsky (1993, 1995) proposes, or it will stall, in line with Frampton and Gutmann's (2002) crash-proof syntax, and never reach the performance systems.

The fact that the features of A might require more than one SO checker is, as Chomsky (2000a, 2001) notes, an apparent "imperfection" in the design of FL. This imperfection emerges in language as the dislocation property—a property in which an SO must show up in more than one place in a derivation. For example, the *wh*-element in (6) must combine with the verb to check its object features in (6a) and then it must reappear as the fronted *wh*-operator in (6b) to check its operator features.

(6) a. [Mary should have [hired who]]
    b. [who should Mary have [hired (who)]]
    'Who should Mary have hired?'

The dislocation property is an imperfection because it appears to compromise the simplicity of the design of FL, requiring FL to have not only the locally defined Merge operation, but also some other operation that relocates an SO in a position distal from its original Merge site.

There have been several minimalist proposals to account for the dislocation property. Chomsky (1994) suggests that the dislocation property derives from the operation Move, which is motivated by the Principle of Greed (see (7)).

(7) Move raises A to a position B only if morphological properties of A itself would not otherwise be satisfied in the derivation.

However, as Collins (1997, 98) points out, moving an SO to satisfy the Principle of Greed will be allowed only "if not moving [the SO] would result in a derivation where the properties of [the SO] would not be satisfied.... But this means that we must look at the derivation where [the SO] is not moved, and see what the outcome of that derivation is." We can see the particular difficulties that result from (7) if we consider how a derivation might proceed after forming the VP in (8).

(8) [$_{VP}$ put the book where]

Notice that starting from (8), we could derive the three divergent, though related, derivations in (9). (Needless to say, we could derive many other sentences from (8) as well; however, these three are illustrative of the point I am making.)

(9) a. Mary told you where to put the book
b. Where did Mary tell you to put the book
c. Who told you to put the book where

The problem that the sentences in (9) pose is how to continue the derivation once the VP in (8) is formed. Does the *wh*-element *where* move in the derivation (as in (9a) and (9b)) or not (as in (9c))? Given (7), should the *wh*-element *where* have the option to move or not? Importantly, if we compare (9a–b) with (9c), we will observe that it will be impossible to determine whether the *wh*-element moves or not until the matrix subject is Merged into the syntax. If the matrix subject is a *wh*-element, then the *wh*-element *where* will not move; on the other hand, if the matrix subject is not a *wh*-element, then the *wh*-element must move. To ensure that we can derive all the sentences in (9) from (8), we will have to either pursue multiple derivations and rule out all but the one required for each sentence, or pursue a single derivation that will be able to go back into the derivation countercyclically and move an element when/if it is finally determined that a movement is necessary. Each of these two types of derivation requires significant cognitive complexity; hence neither of them meets our criterion of maximal simplicity.

Chomsky (1995) proposes that it is not Greed that motivates the Move operation, but Attract. The Principle of Attract, as formulated by Frampton (1997) in (10), states that an SO will move only to satisfy the feature requirements of some head X. That is, SOs move to satisfy the needs of heads, rather than their own needs (compare the Principle of Greed).

(10) *Attract to X*
A phrase is a candidate for attraction to a head X if it has a feature that can potentially satisfy a formal feature of X under movement of the phrase to the checking domain of X. The corresponding movement operation is taken to be well formed if
(1) there is no closer candidate, and
(2) the candidate and the attracting head have formal features of the same type.

Lasnik (1999b) and Bosković (1997a, 1999b), however, argue against Attract as the cause of syntactic movement. They note that *wh*-movement in Serbo-Croatian requires all *wh*-elements to move to the SpecCP position, as is illustrated in (11).

(11) a. Ko sta gdje kupuje?
who what where buys
'Who buys what where?'
b. *Ko kupuje sta gdje
c. *Ko sta kupuje gdje
d. *Ko gdje kupuje sta

If Attract were responsible for the *wh*-movement in (11), then we would expect all the sentences in (11) to be well formed because the [WH] feature of the C head should be satisfied by a single *wh*-movement to SpecCP. This suggests that motivation for *wh*-movement resides in both the features of the *wh*-elements and in the features of the C head. Hence, Attract needs to be recast either as Lasnik's (1995) Enlightened Self-Interest—a principle that permits elements to move to satisfy either their own morphological requirements or the morphological requirements of an attracting head—or as Bosković's (1999) Attract-n-F—a principle that allows a head to attract more than one phrase with feature F.

All versions of Attract—including Chomsky's (2001, 2002, 2005) probe-goal version in which a head H with "uninterpretable" features probes down into the derived structure seeking an XP goal that has "matching" features—are beset with two problems: a conceptual problem and an empirical problem. The conceptual problem with any version of Attract is itself a twofold problem. First, positing the operation Attract (or any variants thereof, such as the Move operation or the Internal Merge operation) in addition to the operation Merge is not a maximally simple design for *FL*. It would be preferable, on minimalist grounds, to have a single operation, say Merge—that is, minimalist theories should explore the ability of Merge alone to account for the dislocation property of language before they propose a second operation hypothesized solely to resolve the dislocation property. Second, Attract attempts to explain long-distance relations (the dislocation property) by appealing to long-distance feature relations (the Attract operation). This is an uninteresting and marginally explanatory treatment because it merely substitutes one long-distance phenomenon for another one, leaving us with the need to explain long-distance feature relations. In other words, explaining displacement phenomena in terms of displacement operations does not really account for displacement at all; it merely reshuffles the terms of the explanation. A much more interesting line of analysis would explain the long-distance dislocation property in terms of the local relations required for Merge.

The empirical problem with Attract is that it cannot account for *wh*-constructions in English. To see this, let us consider how Attract must work to account for sentence (12).

(12) Where do you think that Sam thinks that Chris put the hammer?

In (12), the *wh*-element *where* has to move from its argument position within the VP headed by the verb *put* to the Spec position of the matrix CP. Given Attract, this seems to suggest that the matrix CP will have a C head with a [WH] probe feature and that this feature of the C head will force the movement of the *wh*-element by attracting its matching [WH] feature. According to Chomsky (2001), a head is not free to probe through the entire derived syntactic structure in search of a feature-matching goal; rather, the probe must conform to the Phase Impenetrability Condition, stated in (13).

(13) *Phase Impenetrability Condition (PIC)*
The domain of H is not accessible to operations outside HP, but only to H and its edge.

The PIC, then, allows the head Z in (14) to Attract the head H or an XP in the edge of HP (the edge includes the Spec position of HP and any adjunct to HP), but not any other YP in the domain of H.

(14) $[_{ZP}$ Z . . . $[_{HP}$ XP [H YP]]]

(Note: The PIC applies only if HP is a strong phase—that is, a fully propositional phrase such as a CP with a force indicator or a v*P that selects a VP with a complete argument structure.) If we apply the PIC to (15), we will observe that the matrix C will not be able to probe the *wh*-element in its base argument position because there are several strong phases (CPs and v*Ps) intervening between the matrix C and the *wh*-element.

(15) [C [you $[_{v^*P}$ think $[_{CP}$ that Sam $[_{v^*P}$ thinks $[_{CP}$ that Chris $[_{v^*P}$ put the hammer where]]]]]]]

The PIC prevents the long-distance movement of the *wh*-element in (15) to the Spec of the matrix CP. In fact, the PIC requires the *wh*-element to be moved to the edge of each successive strong phase so that it can undergo Attract to the next higher strong phase until it finally reaches the

Spec of the matrix CP. But what sort of features of v*P and CP will attract the *wh*-element (and no other argument along the way)? And why don't these features attract the *wh*–in situ element in (16a), deriving one of the sentences in (16b–d)?

(16) a. Who thinks that Chris will put the hammer where
b. *Who [thinks [that Chris will [$_{v*P}$ where put the hammer]]]
c. *Who [thinks [$_{CP}$ where that Chris will [put the hammer]]]
d. *Who [$_{v*P}$ where thinks [that Chris will [put the hammer]]]

What prevents the *wh*-element *where* in (16b–d) from being attracted to the edges of all the strong phases and from being stranded in one of these phases as a phase in situ element (in this way (16b–d) could be as well formed as (16a))? The fact that we must require the head of each CP and v*P to have a [WH] feature, or some EPP-type feature that searches for a [WH] feature, able to attract the *wh*-element *where* casts doubt on attract-type analyses of the dislocation property of FL because, as the evidence in (16) suggests, there appears to be no empirical evidence for assuming that these heads have a [WH] feature.

I have demonstrated thus far that both Greed-based analyses and Attract-based analyses of the dislocation property are problematic. What this means is that the dislocation property is not conditioned either by features of the dislocated element alone or by relations between an attracting head and an attracted element. So what then motivates movement in human language? Let us hypothesize that movement is not derived from any long-distance pulling operation such as Attract, but from a local pushing operation SURVIVE, which expels YPs from the domain of head H if YP possesses features that are incompatible with the features of H. SURVIVE, roughly formulated as in (17), is an operation that pushes a YP from the domain of one head to the domain of another head.[3]

(17) *The SURVIVE Principle (first formulation)*
If YP is an SO in an XP headed by X and YP has an unchecked feature [F] that is incompatible with the features of X, YP must Move to the Spec position of the ZP immediately dominating the XP, where the features of X are incompatible with [F] if and only if X has never had a [F_c] feature.

Let me make two clarifications about the SURVIVE Principle. First, according to the SURVIVE Principle, the feature compatibility of YP with the head X reduces to a could-potentially-be-checked-by relation. If

X at any point in the syntactic derivation had a [F_c] feature (which may or may not have been deactivated), X is a potential checker of the [F] feature of YP; consequently, X and YP would not have feature incompatibility. The importance of defining feature incompatibility as "potential feature checking" will be explored when I discuss superraising constructions such as (21). Second, note that the SURVIVE Principle requires YP movement to a higher SpecZP position, rather than to a higher Z′ adjunct position. This constraint on movement follows Chomsky (1995, 2000a, 2001), who argues that (i) there can be no movement to Z′ because there is no Z′ position and (ii) potential feature-checking transpires in Spec positions.

The SURVIVE Principle will force the movements we see in (18).

(18) a. [$_{ZP}$ YP [Z [$_{XP}$ [X (YP)]]]]
b. [$_{ZP}$ YP [Z [$_{XP}$ (YP) [X]]]]

In (18), any YP in the complement position of XP or in the Spec position of XP that has unchecked features incompatible with the features of X must Move into the Spec position of the next highest ZP to try to have its features checked by the head of ZP. If YP cannot have its features checked in ZP, then the SURVIVE Principle will require YP to Move once again to the next highest phrase to have its features assessed. YP will continue to be repelled as long as it has a single unchecked feature, and it will continue to be repelled until it moves into some HP in which all of its remaining unchecked features of YP can be checked by the head H. Under the SURVIVE Principle, the dislocation property emerges as the consequent of incompatible feature relations between a head H and a YP within the HP, and long-distance dislocations such as the one exhibited in (12) result from iterated application of the SURVIVE Principle. That is, (12) is derived from (19) only after the *wh*-element *where*, which has a [WH] feature, is forced by the SURVIVE Principle to move into and then out of each XP headed by an X that lacks a [WH] feature; the *wh*-element will Move until its reaches the matrix CP in which the [WH] feature of the *wh*-element can be appropriately checked.

(12) Where do you think that Sam thinks that Chris put the hammer?

(19) [$_{CP}$ C [$_{TP}$ you [$_{v^*P}$ think [$_{CP}$ that [$_{TP}$ Sam [$_{v^*P}$ [$_{VP}$ thinks [$_{CP}$ that [$_{TP}$ Chris [$_{v^*P}$ [$_{VP}$ put the hammer where]]]]]]]]]]]

(Note: The *wh*-element will move into the Spec of each XP as the XP is being compiled. In this way, the *wh*-element will never undergo any countercyclic movement.)

The SURVIVE Principle has significant theoretical advantages over versions of Attract. First, SURVIVE is a strictly local operation, one that involves relations between a head H and a YP within HP. As such, it is consistent with the locality of Merge, which also involves relations between a head H and a YP. By having only local operations, we have maximally simplified the typology of operations allowed in CHL—an optimal solution. Second, SURVIVE complements Merge in a way that Attract cannot. Merge is a feature-sharing operation; SURVIVE is a feature-nonsharing operation. These two operations then cover the logically possible feature relations between the features of a head H and the features of YPs within HP—that is, H and YP can either share features or not share features. Attract, on the other hand, is a feature-sharing operation, and, in some ways, duplicates Merge. Having redundant operations such as Merge and Attract does not appear to be an optimal solution to the design of *FL*. And finally, Attract requires additional theoretical machinery not required by SURVIVE. In particular, Attract must have some version of Shortest Move or the Minimal Link Condition or the Phase Impenetrability Condition (see condition (1) in the definition of Attract formulated in (10)) to ensure the "locality" of Attract/Move. (For proposals of economy conditions to constrain Attract/Move, see, among others, Chomsky 1993, 1995, 2002, 2005; Collins 1997; Kitahara 1997; and Aoun and Li 2003.) The SURVIVE Principle, which permits only strictly local (repelled) Movements, does not require additional economy conditions such as Shortest Move. Since the SURVIVE Principle necessitates fewer attendant economy conditions than does Attract, the former principle simplifies CHL more than does Attract.

The SURVIVE Principle also has several empirical advantages over versions of Attract. SURVIVE offers a natural explanation of quantifier floating constructions, whereas Attract cannot. To see this, let us consider the data in (20).

(20) a. They (both) were (both) expected (both) to (both) have (both) been (both) elected to the Senate.
b. You (both) seem (both) to (both) be very happy.

If we assume, as does Sportiche (1988), that the quantifiers in constructions such as (20a) and (20b) are left behind by the DPs that eventually occupy the subject position of the matrix clauses, then the data in (20) demonstrate that these DPs must move through all the XPs from the base argument positions of the DPs to their final positions as matrix subjects. Such phrase-to-phrase Movement is predicted by SURVIVE because each DP in (20) has a Case feature that must be checked and this feature cannot be checked in the base argument position of the DP, so SURVIVE pushes the DP upward one XP at a time, looking for an appropriate head H able to check the feature. Each DP in (20) will be repelled through the derivation until the SpecTP position of the matrix clause, where T will check the Case feature of the DP. As the DPs are pushed upward, they have the opportunity to strand a quantifier in any Spec position through which they pass. Attract, on the other hand, has difficulty explaining the data in (20). Under an attract analysis, we can explain the data in (20) only if we assume that the DPs in (20) must be attracted to each XP above their base argument positions that contains a quantifier. But what feature(s) of the heads X will attract the DPs to the XPs? Does each XP with a quantifier in (20) have some EPP feature (or D feature) capable of attracting a DP? (If so, this contradicts Chomsky (2000a, 2001), who suggests that only C and v* have the EPP feature that triggers DP movement, plus, of course, it would be problematic to assume the existence of an EPP-feature in the face of our discussion in the last chapter.) It would seem, then, that Attract requires these XPs to have special features only to account for the data in (20). Unless these attracting features have some independent motivation, the Attract analysis of (20) reduces to an ad hoc solution, rather than to an optimal one.

Not only does SURVIVE provide a simple and economical explanation for quantifier floating, it offers a similarly natural solution for cases of Super-Raising, illustrated in (21).

(21) *Chris was believed it is certain (Chris) to leave soon
'Chris was believed it is certain to leave soon.'

In (21), the DP *Chris* cannot have its Case feature checked in the most embedded sentence, so this DP must have its Case feature checked elsewhere. An Attract analysis attributes the ungrammaticality of (21) to the fact that the DP *Chris* moves to the SpecTP position in the matrix clause and that this Movement is illicit because it violates Shortest Move by moving the DP *Chris* over another DP (*it*) that is closer to the SpecTP position. Importantly, the attract analysis cannot rule out the possibility that there is a well-formed derivation for (21) until it moves the NP *Chris* out of the TP containing the expletive, at which point Shortest Move is violated; hence, the attract analysis cannot rule out a derivation for (21) until it computes the structure of the matrix clause, as in (22a). A SURVIVE analysis rules out (21) for much different reasons. The SURVIVE Principle will force the DP *Chris* to move to the Spec of the TP that contains the expletive, as in (22b).[4]

(22) a. [$_{TP}$ Chris was believed [$_{TP}$ it [is certain [(Chris) to leave soon]]]]
b. [$_{TP}$ Chris [it [is certain [(Chris) to leave soon]]]]

When the DP *Chris* reaches the SpecTP position in (22b) it will not be repelled by the head T because the features of the DP *Chris* are not incompatible with the features of the T head (which has already checked the Case feature of the expletive and has had its Case feature deactivated in the process). The head T cannot check/deactivate the features of this DP, but it cannot repel the DP either; consequently, the DP *Chris* cannot continue to move and its Case feature will be uncheckable. At this point, under Chomsky's (2000b, 2001) assumption that Spell-Out/Interpretation takes place at the phase level (including at the CP phase) and the uncheckable Case feature of the DP *Chris* will violate interpretability conditions at Spell-Out, the derivation will crash; or under the analysis I gave in the last chapter, it will stall. That (22b) crashes, or stalls, disallows any syntactic derivation involving (22b) to proceed any further. Hence, it will be impossible to derive anything that looks like a final/completed derivation for (21); the derivation could never reach that point. This is quite different from an attract analysis, which would permit a derivation for (21) to get beyond (22a) to (22b). The fact that attract analyses permit computations beyond those required of a SURVIVE analysis makes the former analyses less cognitively simple than the latter analysis.

Finally, the SURVIVE Principle explains *wh*-constructions in a much more straightforward manner than does Attract and its variants. Here, I will discuss only the *wh*-constructions in (23); I will analyze other *wh*-constructions latter in this chapter, as well as in the next chapter.

(23) a. *Where did Chris tell you what to put
b. *What did Chris tell you where to put

As Collins (1997) and Kitahara (1997) point out, an Attract analysis of data such as those in (23) must rule out both a cyclic and a countercyclic derivation for these sentences. A cyclic derivation for both (23a) and (23b) is ruled out because once (24a) is derived for the sentences in (23) by moving one of the *wh*-elements to the Spec of the embedded CP, it will be impossible to move the second *wh*-element without violating the Shortest Move economy condition.

(24) a. [*wh*-element$_1$ [to put *wh*-element$_2$ (*wh*-element$_1$)]]
b. [*wh*-element$_2$ [did Chris tell you [*wh*-element$_1$ [to put (*wh*-element$_2$) (*wh*-element$_1$)]]]]

To rule out (25)—a countercyclic derivation for the sentences in (23)—Collins (1997) and Epstein et al. (1998) appeal to Kayne's (1994) Linear

Correspondence Axiom, which constrains linear ordering in phrase markers (see Collins and Epstein et al. for the details of their analyses).

(25) a. [*wh*-element$_2$ [did Chris tell you [[to put (*wh*-element$_2$) *wh*-element$_1$]]]]
b. [*wh*-element$_2$ [did Chris tell you [*wh*-element$_1$ [to put (*wh*-element$_2$) (*wh*-element$_1$)]]]]

It is important to note here that to account for the ungrammaticality of the sentences in (23), attract analyses need to compute two different derivations and they require not only the operation Attract, but also the economy condition Shortest Move and a constraint on the linear ordering of syntactic elements, the Linear Correspondence Axiom. A SURVIVE analysis is much simple than attract analyses. According to the SURVIVE Principle, since both *wh*-elements merged into the VP (see (26a)) have [WH] features that are incompatible with the features of the verb, with the light verb v, and with the T head of TP, these *wh*-elements will be pushed up to the Spec position of the embedded CP, as in (26b).

(26) a. [$_{VP}$ put what where]
b. [$_{CP}$ what [where [C [$_{TP}$ to [$_{VP}$ put (what) (where)]]]]]

Once in the Spec position of the embedded CP, the C head will DEACT/check the [WH] feature of one of the *wh*-elements; the second *wh*-element will not have its feature checked. The features of the second *wh*-element, however, are not incompatible with the C head; as a result, the second *wh*-element will not be repelled from the Spec position of the embedded CP. The unchecked *wh*-feature will cause this derivation to stall when the next phase is derived, regardless of how the derivation proceeds after (26b). We can see, then, that the SURVIVE analysis is indeed simpler than the attract analysis—SURVIVE requires a single derivation to rule out (23a) and (23b), and it does not have to appeal to Shortest Move or to the Linear Correspondence Axiom to explain the ungrammaticality of (23a) and (23b). Further, as with the case of Super-Raising in (21), SURVIVE can terminate the computations of derivation for (23a) and (23b) well before the attract analysis can: SURVIVE could stop the derivation after computing the embedded CP, while Attract must take the computation into the matrix clause, perhaps all the way to the matrix CP. It is clear that SURVIVE permits a much simpler (and less cognitively expensive) derivation for the sentences in (23) than Attract does.

Let us grant, on conceptual and empirical grounds, that the dislocation property of human language follows from the SURVIVE Principle; but let us investigate the SURVIVE Principle further to see if our formulation (17) might be in need of revision.

(17) *The SURVIVE Principle (first formulation)*
If YP is an SO in an XP headed by X and YP has an unchecked feature [+F] that is incompatible with the features of X, YP must move to the Spec position of the ZP immediately dominating the XP, where the features of X are incompatible with [F] if and only if X has never had a $[F_c]$ feature.

The SURVIVE Principle in (17) is formulated as a principle that repels an SO away from any head with which it is incompatible. There is a problem with this "repelling" principle—that is, an SO cannot merely be repelled from an XP, it also has to be repelled to somewhere else (higher) in the derivation. This is problematic since nothing higher in the derivation may exist at that point in the derivation. Consequently, a head X cannot repel a YP until the head that immediately dominates XP is brought into the derivation. The SURVIVE Principle, then, requires the local repelling of a YP out of XP to be delayed until the next head Z is introduced into the derivation. Given that the conditions for the repelling force are established within an XP maximal projection, it is peculiar that the repelling act must be delayed within the derivation until the ZP projection is constructed. For this reason, I would argue that the SURVIVE Principle must be reformulated. If the above arguments are correct, the effects of SURVIVE should be immediate, rather than delayed. Hence, if a YP is "repelled" from XP, it must go somewhere immediately. But where does it go? I would like to propose that in some fashion it "returns" to its pre-Merge position. To clarify my proposal, I must sketch out how LIs are introduced into the syntactic derivation D from the lexicon. Let me begin by following Chomsky (1995, 2001, 2002, 2005) in assuming that lexical items are placed from the lexicon into a lexical buffer, called a Numeration, before they are merged into the syntactic derivation.[5] I diverge from Chomsky, however, in my construction of the Numeration. For Chomsky, the Numeration is created all at once—that is, all the LIs that will eventually appear in D are downloaded into the Numeration from the lexicon prior to the derivation. Chomsky then posits a "smart" Numeration—one that knows in advance of any derivation which LIs will be required to build a well-formed derivation. (Amazingly, even though the Numeration is smart enough to know which derivations will succeed, the derivations themselves are "dumb" in that they do not know in advance if they will succeed or not. If they did, we would not need derivations; syntax could simply generate the well-formed representational output of derivations directly from the lexicon.) For me, neither the derivation nor the Numeration is "smart" enough to predetermine the output of the syntactic derivation. Rather, both the derivation D and the Numeration are built piecemeal and deterministically. Hence in the sentence "Pat likes Chris," the Numeration starts with a single element—the verb {*likes*}—and all subsequent LIs are added to the Numeration from the lexicon on an as-needed basis, eventually building the Numeration {*likes*, *Chris*, *Pat*}. But if Numerations are built piecemeal and have no look-ahead properties, why have such Numerations at all? Why not incorporate LIs into Ds directly from the lexicon? The reason we need to have a Numeration is that LIs do not always map directly into a derivation D; sometimes they must be compiled as syntactic objects (SOs) in a presyntactic derivation workspace (WS) before they are merged into a derivation D. We can see this in (27).

(27) That woman likes Chris

At some point in the syntactic derivation of (27), the DP *that woman* will have to be constructed in the WS before it is merged into the syntactic derivation D. This means that the Merge operation cannot be simply a lexicon-to-D mapping; rather, it must be a mapping, as we see in (27), from a domain that can include both LIs (such as *Chris*) and SOs (such as *that woman*). Defining the Merge domain as the union of the WS and the lexicon creates a strangely disjunctive domain. However, if the Merge domain is the union of the WS and the Numeration, then the domain can be said to be the WorkBench (WB) for D, the "space" that includes all the materials used in constructing D.

Of note, the lexical items (LIs) and syntactic objects (SOs) in any WorkBench are composed of morphophonological, morphosyntactic, and syntacticosemantic features all of which must be checked for interface legibility. Heads will attract LIs or SOs out of the WorkBench to satisfy their own features, merging with the LIs or SOs and checking their features. Heads, however, often will not be able to check all the features of an LI or an SO that require checking. For example, the light-verb head v will Merge with a SU(bject)-argument, but v will not be able to check the Case feature of the subject. If an LI or an SO with features $\langle F_1, \ldots, F_n \rangle$ is Merged with a head H and H deactivates/checks only a proper subset of the features requiring legibility checks, the LI or SO will be said to "survive" in the derivation: it will return to (i.e., be copied in) the WorkBench for Remerge.[6] The surviving LI or SO will be copied in the WorkBench with its checked features marked for deactivation—that is, if the head H checked features F_i and F_j, the LI will be copied in the lexical buffer/Numeration as $\langle F_1, \ldots \#F_i, \#F_j, \ldots F_n \rangle$, where #F indicates that the feature has been checked. All LIs and SOs will "survive" in the derivation as long as they have some unchecked feature F that must be checked for interface legibility. These LIs and SOs must remerge into the derivation from the WorkBench if they are to have their remaining unchecked features appropriately checked, and they must continue to return to the WorkBench and to remerge in the derivation until all their features are checked. An alternative analysis would be to assume that the Merge operation does not involve the movement of lexical material from the WorkBench to a syntactic derivation; rather Merge is an operation that copies lexical material from the WorkBench into a derivation D. Under this analysis, all lexical material merged (copied) into a derivation will also remain, as a master copy, in the WorkBench and any features of an LI, or an SO, that are syntactically checked in a derivation will show up as being checked (and deactivated) in the WorkBench. LIs and SOs in the WorkBench with some, but not all, of their features checked must remerge into the derivation until theireatures are exhaustively, and appropriately, checked or deactivated. In this way, both Merge and Remerge are operations that copy lexical material from the WorkBench into a syntactic derivation D, differing only in what they copy. Merge copies material that has not had any features checked and Remerge copies material that has had at least one of its features checked. Since the latter analysis reduces syntactic operations to copy operations, it should be preferred on simplicity grounds over the former analysis, which required both movement and copy operations.

The foregoing discussion suggests that although SURVIVE looks as if it repels syntactic material up through a derivation, this "repelling" is actually the derivational reappearance of material from the WorkBench. That is, SOs with "surviving" features appear to be repelled because they must be recopied into a derivation via the Remerge operation. Hence, (17) must be reformulated as (28).

(28) *The SURVIVE Principle*
If Y is an SO in an XP headed by X and Y has an unchecked feature incompatible with (i.e., cannot potentially be checked by) the features of X, Y must Remerge from the WorkBench with the next head Z that c-commands XP.

To see how (28) works, we need to address the differences between Merge and Remerge in detail. The Merge operation is a selective operation in which a head H joins with an SO from the WorkBench to satisfy a specific syntactic-semantic (often, a theta-theoretic) feature of the head—for example, a verb such as *admire* will merge with an SO to satisfy the syntacticosemantic Object features of the verb. The Merge operation, which will combine the head H and the SO, will copy the SO from the WorkBench into the derivation, while retaining a master copy of the SO in the WorkBench. Any features of the SO checked by H will also be checked on the copy of the SO extant in the WorkBench. If the head does not check all the features of the SO when they merge, the SURVIVE Principle will require that the SO in the WorkBench be remerged (recopied) into the derivation. Unlike Merge, Remerge is not a selective operation in which a head "attracts" some SO bearing a specific feature; rather, Remerge is an operation in which a head unselectively "attracts" all previously Merged SOs with surviving features to check them for feature agreement. If an SO continues to have unchecked features after Remerge, then SURVIVE will require the SO in the WorkBench to undergo further applications of Remerge until the SO has all of its features checked. Should a derivation terminate with SOs in the WorkBench that possess unchecked features, the derivation will stall because these SOs will have features not properly concatenated for interface interpretation (see my discussion of interface interpretation in the previous chapter).

There are two characteristics of Remerge that deserve special emphasis. First, Remerge appears to be automatic. The conceptual necessity of having automatic Remerge seem to follow from some version of O'Grady's (2005) Efficiency Requirement, which mandates that grammatical dependencies be resolved at the first opportunity. For SOs already in a derivation, they must, in accordance with the Efficiency Requirement, have their unchecked features (grammatical dependencies) resolved/checked at the first opportunity, which would arise as soon as the next head shows up in the derivation. Once a head is introduced into the derivation, it necessarily participates in Remerge, attracting all SOs in the WorkBench available for Remerge. This will explain why all the heads in (28) can host floating quantifiers (I have previously discussed the data in (29); see (20)).

(29) a. They (both) were (both) expected (both) to (both) have (both) been (both) elected to the Senate
b. You (both) seem (both) to (both) be very happy

Additional support for automatic Remerge comes from adverb floating data in *wh*-constructions. Building on some observations by Urban (1999), McCloskey (2000) notes that the adverbs *exactly* and *precisely* can be stranded in ways akin to the quantifier stranding in (29); hence, stranded adverbs can leave a trail of *wh*-movements. With this in mind, consider (30).

(30) a. What (exactly) have you (exactly) been (exactly) saying (exactly)
b. How much money (precisely) have you (precisely) contributed (precisely) to Kerry's campaign

The adverb-stranding data in (30) demonstrate that *wh*-elements can appear in every available Spec position. Importantly, the data in (29) and (30) appear to resist a merge and move analysis since the various heads hosting the stranded quantifiers and adverbs all lack syntacticosemantic features capable of attracting the DPs and the *wh*-elements "moved" in (29) and (30). Merge alone, or Merge and Move together, then, cannot explain why constituents can be stranded in the domain of each and every head in (29) and (30). SURVIVE and Remerge, on the other hand, can account for the data in (29) and (30). The DPs in (29) have Case features that must be checked, and the *wh*-elements in (30) have [WH] features that must be checked. Under SURVIVE, which requires automatic Remerge of SOs with "surviving" features, these elements will be forced to remerge with each successive head looking for feature-checkers, and SURVIVE will compel these elements to undergo remerge until all their checkable features are appropriately deactivated. Hence, since the DP merged in (29) has a surviving Case feature, it must remerge into the derivation with each available head until the Case feature is checked; at the site of each Remerger, the DP can potentially leave a quantifier behind. Similarly, the *wh*-elements merged in (30) have surviving [WH] features, so these elements must be remerged head by head until the [WH] features are checked and they could strand their adjuncts in any of the remerger sites.

The second characteristic of Remerge is that it is structured. That is, not only must a head remerge all previously merged SOs, but the head must check them one at a time, in order of their first appearance in the derivation. This ordering condition on Remerge is necessary to account for the crossing phenomena discussed in Richards 1999. As Richards observes, the "movements" in (31) and (32) exhibit strict ordering relations in which the left-to-right surface ordering of moved-elements reflects their point of merger—if A is merged into the derivation before B is, then

A will show up to the right of B in the surface form of the derivation. (See Richards for other cases of this crossing effect.)

(31) *Bulgarian (from Rudin 1988)*
    a. Koj kogo vizda
       who whom sees
       'who sees whom'
    b. *kogo koj vizda

(32) *Icelandic (from Collins and Thráinsson 1993)*
    a. eg lana ekki Mariu b kurnar
       I lend not Maria the books
       'I do not lend Maria the books.'
    b. eg lana Mariu b kurnar ekki
    c. *eg lana b kurnar Maria ekki

In (31), the *wh*-object must be merged into the syntax before the *wh*-subject, and the surface order of the *wh*-elements must inversely reflect the Merge ordering. In (32), the direct object is merged before the indirect object (as (32a) indicates); however, if the objects are fronted as in (32b–c), they must show up in their inverse Merge order. Since Remerge is responsible for "movements," the data in (31) and (32) suggest that the remerging of elements follows the order in which they merge. One way to ensure this ordering of Remerge is to have a structured buffer in the WorkBench. In particular, SOs with surviving features will be placed into a subbuffer containing once-merged elements and this subbuffer will be structured vertically, in a top-down structure that places new elements for remerger at the bottom of the structure (similar proposals for the lexical storage of elements introduced into a syntactic derivation are made by Kural (2005), who posits a structured lexical array used in the derivation, and by O'Grady (2005), who posits a pushdown storage for elements in the working memory used during syntactic computations). The unselective operation Remerge will then remerge elements from the subbuffer one element at a time, in a top-to-bottom ordering. Under this formulation of Remerge, if the objects in (31) have a Focus feature that cannot be checked by the verb and, as a result, these object must undergo Remerge, they must remerge in the order of their merger; hence, (31b) will be a successful application of Remerge, while (31c) will not.

Although my reformulation of SURVIVE has changed SURVIVE from a repelling principle to a Remerge principle, it remains a principle that accounts for the dislocation principle in terms of the local relations between heads and SOs. The apparent long-distance movement of an SO is the consequence of a sequence of local remergers necessitated by the continued survival of some feature of the SO that must be checked.

To see how SURVIVE works in detail, let us consider the sentence in (33).

(33) Who did Sam tell how to fix what

There are two interesting facts about (33). First, the *wh*-element *what* does not have to move to the Spec of any CP; rather, the *wh*-element can remain within a VP or perhaps vP. Second, the *wh*-element *what* must be interpretively linked to the matrix *wh*-operator *who* and not to the embedded *wh*-operator *how*, as is demonstrated by the fact that (34a), which interpretively connects *who* and *what*, is an acceptable answer to (33), while (34b), which connects *what* with *how*, is not.

(34) a. Sam told me how to fix the bicycle
b. *Sam told me how to fix what

To account for the relationships among the *wh*-elements in (33), we must first identify the features of the *wh*-elements. The two *wh*-elements—*who* and *how*—both must have a feature that is checked in the Spec position of CP and that cannot be checked anywhere else in the derivation (in VP or vP). This feature cannot be the [WH] feature because, as the *wh*–in situ element *what* inside the vP or the VP shows, the [WH] feature is not incompatible with (repelled by) the verb. It must be the case, then, that the *wh*-elements in the Spec of CP have a feature that the *wh*–in situ element does not have. Since *who* and *how* are *wh*-operators, let's assume that they have an [OP] feature that the *wh*–in situ element does not have. Although the two operators in (33) share the [OP] feature, they also have features that set them apart. According to Rizzi (1990) and Cinque (1990), the *wh*-operator *who* is a referential operator, one that can select participants in events, but the *wh*-operator *how* is not referential. To express this difference between *who* and *how*, let us say that *who* has both an [OP] feature and a [REF] feature, while the *wh*-element *how* lacks the latter [REF] feature, possessing only an [OP] feature. Finally, the *wh*–in situ element, though lacking the [OP] feature, does have a [REF] feature (see Stroik 1995 for arguments in support of the [REF] feature). However, as (34a) indicates, a *wh*–in situ element is paired with a *wh*-operator and the referentiality of the *wh*–in situ element is dependent on the referentiality of the operator with which it is paired. That is, the *wh*–in situ element forms a referential ordered pair with a referential *wh*-operator (Stroik (1995) and Zubizarreta (1998) propose paired analyses of multiple-*wh* constructions in English). This suggests that the [REF] feature of the *wh*–in situ element must encode its referential dependence on another *wh*-element; I will express the complex referentiality of a *wh*–in situ element as a [REF/WH] feature.[7] (Note: Adger and Ramchand (2005) also argue that features can involve dependency relations; they use such features to explain the distribution of complementizers in Scottish Gaelic.)

The above features play a significant role in the derivation of (33), repeated below.

(33) Who did Sam tell how to fix what

The derivation of (33) begins with the verb merging with an object argument (the object will be selected-out of the lexical items placed in the Numeration from the lexicon; see Collins 1997 and Chomsky 2000a for discussions of lexical selection). This Merger will yield (35a).

(35) a. [fix what]

The verb will be able to check the lexicosemantic features of the DP *what* to ensure a semantic (theta) compatibility with the DP object. The verb may also be able to check the Case and object-agreement features of the object (as Epstein et al. (1998) propose), though, according to Chomsky (2000a), these features may have to be checked by the light verb v, rather than by the main verb. If the verb has checked the Case and agreement features of the DP, then these features will be deactivated in the Numeration and they will eventually be spelled out in the VP where they are checked (i.e., the word *what* would show up morphophonetically in its merged position). On the other hand, if the morphophonetic features of the DP (Case and agreement features) are not checked by the verb, these features will survive in the WorkBench and the DP will undergo Remerge until these features are checked. Although the verb might be able to check the Case and agreement features of the DP *what*, it will not be able to check the [REF/WH] feature of the DP. Since the DP will have an unchecked feature, as will its copy in the Numeration, SURVIVE will require that a copy of the DP in the Numeration be available for Remerge. However, before this remerging can take place, the verb *fix* will merge with the adverb *how* (the adverb could also merge later in a functional category, as Cinque (1999) argues; since the merger site is not relevant to my discussion, I will assume, for ease of exposition, that the adverb merges with the verb). The Merger of the adverb will derive (35b).

(35) b. $[_{VP}$ how [fix what]]

The verb in (35b) will be able to check the lexicosemantic features of the adverb, but not its [OP] feature. Therefore, according to the SURVIVE Principle, the adverb will remain in the Numeration with its [OP] feature still active. The VP now merges with the light verb v, which attracts a subject argument PRO and triggers Remerge of the previously merged elements. The derived vP will be (35c).

(35) c. $[_{vP}$ how [what [PRO [v [how [fix what]]]]]]

(In my discussion, I will be charting only the multiple (re)mergings of the *wh*-elements and will not comment on verb remergings or on non-*wh* DP remergings.) The light verb will check, among other things, the unchecked features of the remerged elements for feature compatibility. If the Case and agreement features of the DP object have not been checked by the verb (as Chomsky (1995) assumes), then these features will be checked by the light verb (should these morphophonetic features be checked in vP, the DP will show up in the surface form of the sentence in the vP). However, the light verb will not be able to check the [REF/WH] feature of the DP, so the DP *what* will remain active in the lexical buffer of the WorkBench with all of its features deactivated except its [REF/WH] feature. The light verb will also not be able to check the [OP] feature of the adverb *how*; consequently, the adverb will remain active in the lexical buffer. The vP will next merge with T (the head of TP). The T head will remerge with both the *wh*-object and the *wh*-adverb, but it will not be able to check the unchecked features of these *wh*-elements and, as a result, these elements will continue to remain active in the lexical buffer. (Since T does not deactivate any features of what or of how, I will not provide a derivation for TP.) TP then merges with a C head, which will remerge with the elements in the lexical buffer that have remained active in the buffer for remerger. The resulting derivation will be (35d).

(35) d. $[_{CP}$ how [what [C . . . $[_{vP}$ how [what [PRO [v $[_{VP}$ how [fix what]]]]]]]]]

The C head in (35d) will be able to check the [OP] feature of the adverb *how*, leaving the adverb with no other features to check. Hence, the adverb will not remain active in the lexical buffer; it will remain in the Spec of CP, where it will undergo morphophonetic Spell-Out at the sensorimotor interface. Furthermore, to ensure compatibility with its operator, the head C will not have a [REF] feature—should the C have a [OP, REF] feature matrix, it would be incompatible with the [OP] *wh*-adverb, and the adverb would have its [OP] feature SURVIVE in the lexical buffer for Remerge. Given that the head C is not a [REF] head and not able to agree with the *wh*-operator *how*, it then follows that this head cannot check the [REF/WH] feature of the *wh*–in situ element. Possessing an unchecked feature, the *wh*–in situ element must be remain in the lexical buffer of the WorkBench available for Remerge. The derivation continues in similar fashion once the verb *tell* merges with the embedded CP. The verb will be able to check the lexicosemantic features of its object *who*, but it will not be able to checked the [OP] feature of its object, nor the [REF/WH] feature of the *wh*–in situ element. These *wh*-elements, with their unchecked features, will also remain in the WorkBench available for Remerge. They will remerge with each head as it enters the derivation; however, their unchecked features will survive until the *wh*-elements are Remerged in the Spec of the matrix CP (see (35e)).

(35) e. [$_{CP}$ who [what [C . . . [$_{CP}$ how [what [C . . . [$_{vP}$ how [what [PRO [v [$_{VP}$ how [fix what]]]]]]]]]]]]

The matrix C will check the [OP, REF] features of the *wh*-element *who* and will also be able to check the [REF/WH] feature of the *wh*–in situ element. At this point in the derivation, there will be no elements with unchecked features; consequently, the derivation will terminate with all its elements checked for compatibility with interface requirements. The derivation culminating in (35e) will be a well-formed derivation in which the *wh*–in situ element is interpreted morphophonetically within the embedded vP or VP (depending on where its Case feature is checked) and in which the *wh*–in situ element is referentially dependent on the matrix operator.

The SURVIVE Principle, as we can see in the discussion of (35a–e), plays a crucial role in deriving multiple-*wh* constructions such as (33); however, my SURVIVE analysis also extends to the multiple-*wh* constructions in (36) and (37).

(36) a. *Why did Pat buy what
b. *What did Pat buy why

(37) a. Who saw a picture of whom
b. *Who saw the picture of whom

According to Hornstein (1995), the sentences in (36) are both ungrammatical under pair-list readings. I can explain the ungrammaticality of (36a) and (36b) in a natural way under my SURVIVE analysis. Specifically, (36a) will be ungrammatical because the [REF/WH] feature of the *wh*–in situ element *what* cannot be checked by any head—not even the C head, which must have the [OP] feature to ensure compatibility with the nonreferential *wh*-operator *why* in the Spec of CP and therefore cannot have not the [REF] feature necessary to check a [REF/WH] feature. The *wh*–in situ element with its unchecked feature, therefore, will remain in the Numeration at the termination of the derivation, and the derivation will collapse because it will have an element not checked for compatibility at the interface. In a similar vein, (36b) will be ungrammatical because its *wh*–in situ element will be a referentially dependent *wh*-element [REF/WH]; however, this element will lack the [REF] feature required to establish the paired dependency with an operator. The [REF/WH] feature, then, will survive the derivation, and it will force the derivation to stall or crash.

The grammaticality difference between (37a) and (37b) can also be explained under my SURVIVE analysis. In (37a), the [REF/WH] feature of the *wh*-in-situ element survives past its Remerger with the indefinite, nonreferential determiner head of the DP containing the *wh*-element.[8] This feature will survive up to the C head, where it will be appropriately checked by the head possessing [OP, REF] features. On the other hand, (37b) is ungrammatical because the D head of the DP is a [REF] head. The [REF] feature of the *wh*–in situ element is compatible with the D head and will not survive beyond this head. If the satisfaction of the [REF] prevents the entire [REF/WH] feature from surviving, then the [REF/WH] will not be unchecked, and the derivation will stall or crash. If, somehow, the [REF/WH] feature of the *wh*–in situ element does survive the D head, it will eventually Remerge with a C head that requires the [REF] feature necessary for the paired dependency. Lacking the requisite [REF] feature, the *wh*–in situ element will survive the C head with its [/WH] feature active and the derivation will subsequently crash because of this unchecked feature. That the [REF] feature plays as important a role in the derivation of the sentences in (37) as I claim, can be seen if we add a [REF] feature to the indefinite D head in (37a), as in (38).

(38) *Who saw a certain picture of whom

Once the indefinite D head takes on a [REF] feature, it will no longer permit a *wh*–in situ element to pass along features that would license the *wh*-element as a member of a referential ordered pair.

As my discussion of multiple-*wh* constructions in English demonstrates, A SURVIVE approach to syntactic derivation employs only local operations and relations. The operations Merge and Remerge are local operations and the repelling/coping operation SURVIVE is also a local operation. Using only local operations maximally simplifies the computation of a derivation because the derivation does not require any look-back or look-forward computations. Furthermore, a SURVIVE analysis of syntactic derivation does not permit any countercyclic operations: all computations involving a head occur when the head is merged into the syntax and all computations involving nonheads occur at the point of merger or remerger. That is, all computations must occur only in the phrasal domain of the most recently merged head and, therefore, these computations are strictly local. If maximal simplicity is a necessary condition for an "optimal" solution to the design of HL, then the solution offered by a SURVIVE analysis needs to be considered seriously because SURVIVE is a maximally simple design that is also a "good" design—that is, an empirically adequate design.

Conclusion

The empirical arguments I present in this chapter confirm the conceptual arguments that I muster in the first chapter: they demonstrate that (i) an optimal theory of grammar will not include any operations other than strictly local ones and (ii) syntactic operations will not move constituents. Nonlocal operations and any variety of Move or Attract operations not only violate Brody's (2002) conditions on syntactic representations (as I discuss in chapter 1), but they also fail to explain the empirical data I discuss in this chapter. In addition, any nonlocal operation will be so powerful that it must be constrained by some sort of minimality condition(s); hence nonlocal operations necessarily expand the ontology of a grammar. What I show in chapters 1 and 2 is that a theory of grammar will simplify its conceptual and ontological commitments, while extending its ability to account for empirical data, if its operations are restricted to two copying operations—Merge and Remerge—that necessarily apply locally and that map SOs from the WorkBench to the derivation.

Under my analysis, a syntactic derivation is a continuous, bottom-up, structure-building derivation. Both Merge and Remerge (which is triggered by the SURVIVE Principle) accrete structure without ever going back to alter any of the structure already built. As a result, these operations will generate a single end-product derivation that becomes the representation submitted to the sensorimotor and the conceptual-intentional interfaces for interpretation (thereby satisfying Brody's (2002) conditions on representations and Epstein et al.'s (1998), Frampton and Gutmann's (2002), and Fitzpatrick's (2002) conditions on derivations).

3 Some *Wh* Puzzles

Introduction

Wh-constructions pose a special challenge for any theory of syntax, in part because of their rich morphophonetic variety. There are languages, such as Bulgarian and Serbo-Croatian, that allow multiple *wh*-elements to be spelled out in fronted positions, as in (1).

(1) *Bulgarian (from Rudin 1988)*
   a. Koj kogo vizda?
      who whom sees
      'Who sees whom?'
   b. *Serbo-Croatian (from Bošković 1997c)*
      Ko je koga vidjeo
      who AUX whom seen
      'Who saw whom?'

There are also languages, such as Chinese, that spell out all *wh*-elements in their Merged position—that is, none of these *wh*-elements are morphophonetically spelled out in fronted positions. This is shown in (2).

(2) *Chinese (from Huang 1982)*
   Ni xiang-zhidao shei mai-le sheme?
   you wonder who bought what
   'What do you wonder who bought?'

And there are language, such as English, that front some *wh*-elements but not others, as is illustrated in (3).

(3) Who did you tell to do what

This variety certainly complicates the explanatory challenges confronting any theory. However, an even more daunting challenge comes from the fact that theorists radically disagree about the *wh*-data themselves. For example, Huang (1982) and Fiengo et al. (1988) claim that (4) is an example of a well-formed multiple-*wh* construction; Hornstein (1995) and Stroik (1992, 2000) disagree.

(4) Why did you buy what

Furthermore, Richards (2001) and Fiengo et al. (1988) judge multiple-*wh* constructions with *wh*–in situ contained within islands to be well formed (see (5a–c)) and interpretively indistinguishable from the *wh*-construction in (5d), which does not have a *wh*-element contained within an island; on the other hand, Dayal (2002) finds the examples in (6a–b) to be grammatical, but she contends that they must have only a single-pair reading, and not the pair-list reading available for the *wh*-construction in (6c) (Stroik (2000) makes a similar point). And while Kayne (1983) finds that the Superiority Effect in (6d) is ameliorated by adding a third *wh*-element (as in (6e)), Clifton, Fanselow, and Frazier (2006) find that their processing experiments suggest that this is not the case.

(5) a. Who persuaded [the man who bought which car] to sell the hubcaps (Richards)
b. Who likes [books that criticize who] (Fiengo et al.)
c. Who got jealous [because I spoke to who] (Fiengo et al.)
d. Who bought what (Fiengo et al.)

(6) a. Which student read [the book that which professor wrote]
b. Which student got a headache [after she read which book]
c. Which philosopher likes which linguist
d. *What did who buy there
e. (?)What did who buy where

The fact that *wh*-constructions come in diverse forms both within and across languages offers one set of explanatory challenges; the fact that judgments about the *wh*-data are widely disparate and often contradictory offers another.

In this chapter, I will show how my Merge and Remerge analysis of syntax can explain the diverse forms of *wh*-constructions and how it sheds valuable light on our understanding of *wh*-data. Although my discussion will focus on multiple-*wh* constructions in English and on *that*-trace effects, I will also analyze *wh*-constructions in languages other than English.

Let me begin my discussion of *wh*-constructions by showing how my minimalist syntax analyzes the (maximally) simple *wh*-construction in (7).

(7) Who snores

The derivation of (7) goes as follows, with a couple of simplifications—I will not, for example, include the light verb v in my derivation and I will not discuss how the T(ense) head comes to check the Tense feature on the verb. Since the verb *snores* does not have a complement feature and has no modifiers, the verb will merge with its Subject argument *who* after these two lexical items are placed in a Numeration.

(8) a. Merge ⟨who, snores⟩ → who snores

The verb will check the thematic feature of the DP *who*; however, it will not be able to check the agreement features (Person, Number, and Case) or the [WH] feature of the *wh*-element. These unchecked features then survive on the copy of *who* in the Numeration. The merged constituent will next merge with T (if it did not do so, the Tense feature on the verb could not be checked).

(8) b. Merge ⟨T, ⟨who, snores⟩⟩ → T who snores

At this point, the SURVIVE Principle will require the DP *who*, which has surviving features, to remerge with the T head.

(8) c. Remerge ⟨who, ⟨T, ⟨who, snores⟩⟩⟩ → who T who snores

The T head will check the agreement features of the DP *who*, but not its [WH] feature, which continues to survive in the Numeration. Because (7) is a question, it will have a mood operator—a C head with a [WH] feature—merged into the derivation, as in (8d).

(8) d. Merge ⟨C, ⟨who, ⟨T, ⟨who, snores⟩⟩⟩⟩ → C who T who snores

And finally, in accordance with the SURVIVE Principle, the DP *who* must automatically remerge with the C head; in the process of this remerging, the DP will have its last unchecked feature checked by the C [WH] head.

(8) e. Remerge ⟨who, ⟨C, ⟨who, ⟨T, ⟨who, snores⟩⟩⟩⟩⟩ → who C who T who snores

As the end product of the derivation, (8e) will be a well-formed representation that will be sent by Chomsky's (2002) Transfer operation to be interpreted by both the sensorimotor interface and the conceptual-intentional interface. The sensorimotor interface will interpret only overt morphophonetic features and it will interpret these features where their concatenation relations are checked.[1] Since the only overt morphophonetic feature of the DP is its [WH] feature, the DP will be spelled out where this feature has been checked in the SpecCP position. This means that the sensorimotor interface will interpret the highest copy of the DP and none of the other copies. In fact, this interface will interpret (8e) in the following way.

(8) f. ⟨who, ⟨C, ⟨who, ⟨T, ⟨who, snores⟩⟩⟩⟩⟩ → who (C who T who) snores

The material in parentheses has no morphophonetic interpretation because the features checked for/by these elements do not involve morphophonetic features.

As with the sensorimotor interface, the conceptual-intentional interface will interpret only the relevant logicosemantic features in (8e) that have been appropriately checked. Even though the Numeration has only a single copy of the DP *who*, this DP will be interpreted at the conceptual-intentional interface in each of the three structural positions it appears in because a logicosemantic feature has been checked in each of these concatenations: its lowest copy will receive a thematic interpretation, its middle copy will receive a number interpretation, and the highest copy will receive a *wh*-operator (and *wh*-scope) interpretation. All the links of the ⟨*who*, *who*, *who*⟩ chain in (8e), then, contribute to the semantic interpretation of the DP in (7).

Bear in mind that once a feature is checked, it is deactivated and cannot be checked again. As a result, all features of a lexical item can be checked in only one place and, therefore, can be interpreted in only one place. Should a derivation create a copy chain at a given interface, such as the ⟨*who*, *who*, *who*⟩ chain discussed above, each link in the chain must contribute nonoverlapping content to the interpretation because each link must check a unique feature. This consequence of feature-checking prevents the creation of chains with multiple Case features or multiple thematic roles.

Wh-Features

Before we can analyze *wh*-constructions that are significantly more complicated than the one in (7), we must undertake something rarely attempted by theorists—we must closely examine the [WH] feature. Most theorists have simply assumed that all *wh*-elements have the same [WH] feature (an operator feature). (We can see this in many sources, including Huang 1982; Hornstein 1984; May 1985; Aoun 1985; Lasnik and Saito 1989, 1992; Aoun and Li 1993; Chomsky 1995, 2001; and

Collins 1997.) This assumption has led these theorists to believe that all multiple-*wh* constructions have the same range of readings. However, as Dayal (2002) observes, (9a) and (10a) do not in fact have the same sorts of readings. Sentence (9a) can take a pair-list reading, admitting paired answers such as (9b); on the other hand, (10a) does not permit a pair-list reading, allowing only singleton answers as in (10b) but not paired answers as in (10c)

(9) a. Which philosopher likes which linguist
 b. Professor Smith likes Professor Brown and Professor King likes Professor Matthew

(10) a. Which student believes that Mary read which book
 b. Smith believes that Mary read Chomsky's last book
 c. Smith believes that Mary read Chomsky's last book and Jones believes that Mary read Lasnik's last book

What the data in (9) and (10) suggest is that not all *wh*-elements act as pure logical operators, or else (9a) and (10a) should both permit pair-list readings. That is, it must be the case the two *wh*-elements cannot both be independent operators. Zubizarreta (1998) comes to a similar conclusion. Noting that sentences can have one and only one focus and that *wh*-constituents carry focus, she maintains that multiple-*wh* constructions such as (9a) and (10a) cannot treat each separate *wh*-element as an independent (focused) operator along the lines advanced by Huang, May, Aoun, and the other theorists mentioned above; rather, the *wh*-elements must be interpreted as a (focused and linked) pair.[2]

Let us assume, following Zubizarreta, that the *wh*-elements in (9a) and (10a) must be linked. But how? Hornstein (1995) argues that this sort of linking is a form of pronominal binding. For Hornstein, in a multiple-*wh* construction such as (11a), one of the *wh*-elements is an operator and the other (the *wh*–in situ element) is a complex constituent with an implicit pronoun (see (11b)) that is interpreted functionally by being operator-bound.

(11) a. who bought what
 b. [who$_i$ [t$_i$ bought [pro$_i$ N]]]

Although Hornstein does offer a mechanism that links the *wh*-elements in a multiple-*wh construction* together, there are significant problems with his analysis, of which I will discuss one particularly distressing problem here (however, see Stroik 2000 and Clifton, Fanselow, and Frazier 2006 for extended presentations of these problems). To see this problem, let's consider how Hornstein's analysis would apply to (9a) and (10a). Under his analysis, (9a) and (10a) would have (12) and (13), respectively, as their logical representations.

(12) [which philosopher$_i$ [t$_i$ likes [pro$_i$ linguist]]]

(13) [which student$_i$ [t$_i$ believes [that Mary read [pro$_i$ book]]]]

Notice that in both (12) and (13) the *wh*-operators can bind pronominal elements. This means that the linkage between the operators and pronouns are the same and that, consequently, the operators and the pronouns in (12) and (13) should engage in the same interpretations. Hence, Hornstein's analysis predicts that (9a) and (10a) should permit the same readings. This, however, is not the case according to Dayal: for her, (9a) permits a pair-list reading, while (10a) does not. It would seem then that Hornstein's analysis fails to account for the *wh*-linkages required to disentangle the readings given to (9a) and (10a). Of course, we could attempt to rescue Hornstein's analysis by stipulating some type of locality condition on the interpretation of pronominal binding that might give the distal binding in (13) a different set of interpretations than the set given to local binding. Any such proposals would quickly run afoul of the fact that pronouns bound by *wh*-operators do not exhibit local-versus-distal differences in interpretation, as the examples in (14) illustrate.

(14) a. Who$_i$ likes his$_i$ mother
  b. Who$_i$ believes that Mary likes his$_i$ mother

Unless Hornstein's analysis can be revised to explain the interpretive differences between (9a) and (10a), it must be rejected on empirical grounds.

An alternative analysis of *wh*-linking is presented in Stroik 1995, 2000. In this work, I argue that the *wh*-elements in sentences such as (9a) are not linked via the operator binding of a pronominal element; instead, they are interpretively linked as ordered pairs in which the value of the in situ element is referentially dependent on the value assigned to the fronted *wh*-operator. Under my analysis, the *wh*-elements in (15) are interpreted at the conceptual-intentional interface as an ordered pair ⟨who, what⟩ and in this pair, the operator *who* can freely take any referential value, while the reference value assigned to the in situ element *what* can be determined only after a value for *who* has been selected.

(15) Who read what

The relationship between the *wh*-operator and the *wh*–in situ element, then, is the same as the ordered-pair relationship between x and y in the mathematical function $y = F(x)$. That is, as the range value of y is fixed by the domain value assigned to x, so the semantic (referential) value assigned to a *wh*–in situ element is fixed by the value assigned to a *wh*-operator. Support for my ordered-pair analysis of multiple-*wh* constructions comes, in part, from the fact that if no reference value is assigned to a *wh*-operator, then no reference value can be can be assigned to a *wh*–in situ element either. We can see this if we look closely at the answers to (15) that are given in (16).

(16) a. Everyone read nothing/the Bible
b. #No one read everything/the Bible

Notice that (16a) is an acceptable response to (15) because the *wh*-operator has been assigned a reference value ("everyone") and this domain value allows the *wh*–in situ element to be assigned any range value (including the null value "anything"); on the other hand, (16b) is not an acceptable response to (15) because the *wh*-in situ element has been assigned a (nonnull) reference value when the *wh*-operator has not. Similarly, notice that if one of the *wh*-elements in a multiple-*wh* construction is nonreferential as in (17) and (18) (Cinque (1990) and Rizzi (1990) claim that the *wh*-adjuncts *why* and *how* are nonreferential), then it is impossible to establish a referential dependence between a *wh*-operator and a *wh*–in situ element—that is, the *wh*-elements will not be linked as they must be according to Zubizarreta (1998).

(17) a. *What did Chris read why
b. *What did Chris read how

(18) a. *Why did Chris read what
b. *How did Chris read what

(For additional discussion of my ordered-pair analysis, see Stroik 1995, 2000.)

Importantly, my analysis attributes the grammatical properties of multiple-*wh* constructions to lexical features—much in the spirit of minimalist analysis. The operator *who* forms a licit ordered pair with *what* in (15), while the operators *why* and *how* in (18) do not, because the operator *who* has a reference feature [REF] that *why* and *how* lack. The latter operators have only an operator feature [OP]. Furthermore, the *wh*–in situ element *what* in (15) differs from the operator *who* in at least two ways: it lacks an operator feature and it has a referential dependency feature [REF/] (dependency features are also proposed in Adger and Ramchand 2005). Since the *wh*–in situ element is a *wh*-element dependent on a *wh*-operator, its referential-dependency feature is fully expressed as [REF/WH]. It would seem, then, that the distribution of *wh*-elements has much to do with their features.

Strangely enough, the referential-dependency feature carried by the *wh*–in situ element *what* in (15) brings us back to Hornstein's (1995) analysis. Recall that Hornstein differentiated operators from *wh*–in situ elements by assigning the latter elements pronominal features. Given that *wh*-words have traditionally been identified as (interrogative and relative) pronouns, there might be some wisdom in finding pronominal properties in *wh*-elements or perhaps finding parallels between pronouns and *wh*-elements. Pronouns come in at least three varieties: R-expression pronouns such as *the bastard*; structure-dependent anaphors, such as *herself* and *each other*, which must be locally linked to structurally expressed antecedents; and structure-independent pronouns, such as *she* and *they*, which cannot have local antecedents, but can have discourse-linked antecedents. *Wh*-elements, interestingly, also come in three equivalent varieties. They show up as *wh*-operators (R-expressions); as structure-dependent *wh*–in situ elements that must be (locally) linked to referential operators; and as discourse-linked *wh*-elements. We can observe the last *wh*-elements in echo constructions (see (19)).

(19) a. Chris read what
b. Chris believes that you read a book about what
c. Chris got jealous because you spoke to whom

Echoic *wh*-elements, much like structure-independent pronouns, need not be linked to other *wh*-elements and are assigned a value within a discourse; these *wh*-elements will have a discourse-dependence feature [DISC] similar to the one possessed by the pronoun *she*.

If *wh*-elements are indeed pronouns, then we should expect that they should have distributional features that parallel the distributional features of other pronouns. I claim above that *wh*-elements do have such distributional features. In particular, the [WH] feature on a *wh*-element such as *who* can have three subfeatures—an [OP] feature, a [REF/WH] feature, and a [DISC] feature—that are responsible for the distributional and logical properties of *wh*-constructions.

SURVIVE and *Wh*-Constructions

In this section, I will give a detailed analysis of how the features of *wh*-elements determine the properties of *wh*-constructions. My analysis will show these properties are a natural consequence of applying Merge, SURVIVE, and Remerge to *wh*-features.

Let's begin by looking at *wh*-echo constructions. These constructions are particularly interesting because *wh*-echo elements are permitted to be in every imaginable grammatical context. They can show up as subjects of tensed clauses (20a), subjects of infinitives (20b), objects of verbs (20c), and objects of prepositions (20d).

(20) a. Chris believes that who will win
b. Chris wants who to win the race
c. Chris expects to win what
d. Chris will read to who

They can also show up in relative clauses (21a), adjunct clauses (21b), coordinate constructions (21c), and focused constructions (21d).

(21) a. Pat likes books that criticize who
b. Pat was happy after Mary fired who
c. Pat likes Bob and who
d. Pat will read only to who

They can show up a singletons, as in (20) and (21), or they can show up multiply, as in (22).

(22) a. Sam told Pat that who read what
b. Sam convinced whom to read what to whom

Wh-echo elements are so licentious (and ubiquitous) that little has ever been said about them. They are largely taken for granted. But they are *wh*-elements and one would suspect that, as such, they have [WH] features that must be checked. Needless to say, this begs very important questions: What could these [WH] features possibly be, if they exist? How might they be checked, if they exist?

Before we answer the questions just posed, let's make a few additional observations about *wh*-echo elements. All the *wh*-echo elements in (20)–(22), it is important to note, have the same properties: they do not exhibit any displacement (i.e., they are spelled out in their merged positions); they are all D(iscourse)-linked in Pesetsky's (1987) sense, unable to tolerate *wh*-the-hell variants (see (23)); and when they show up multiply, they do not get a paired interpretation—we can observe this, in part, in (24), where the responses to (22) permit the more deeply embedded *wh*-elements to have a reference value despite the fact that the higher *wh*-element has not been assigned a reference value.

(23) a. *Chris expects to win what the hell
b. *Pat will read only to whom the hell
c. *Sam told Pat that who the hell read what

(24) a. Sam told Pat that no one read the Bible
b. Sam convinced no one to read the Bible to Mary

If we are to offer a compelling account of *wh*-constructions, we obviously need to explain the properties of *wh*-echo elements, including their seemingly unconstrained distribution.

Although the data surrounding *wh*-echo constructions are admittedly vast, we can successfully explain these data if we pay close attention to the features of *wh*-echo elements. As I discuss above, *wh*-echo elements are *wh*-elements that have [WH] features. The [WH] features they have are [DISC] features, features that must be interpreted within a discourse in much the same way that the [DISC] features of pronouns such as *her* are interpreted. If we compare the pronouns *her* and *whom* in (25a,b) with the pronoun *whom* in (26), we will get a clear sense of how the [DISC] feature is interpreted.

(25) a. Chris likes her
b. Chris likes whom

(26) Whom does Chris like?

The pronouns in (25a) and (25b) differ from the pronoun in (26) in that the former pronouns take their antecedents (or referents) from the backgrounded discourse, whereas the latter pronoun can take a discourse-free antecedent (or referent). The salient fact about [DISC] features, then, is that they are discourse features, not concatenation (i.e., structural) features. Since [DISC] features are not concatenation features, they do not have to be checked in the course of a syntactic derivation, as I argued in the previous chapter. Consequently, a [DISC] feature is not a feature that can be said to survive a syntactic operation, and so it is not a feature that will trigger the Remerge operation. What this means is that the [DISC] feature will remain inert throughout the syntactic derivation. We can see this if we look at the derivation for (27a).

(27) a. Chris likes whom
b. Chris likes Pat

The derivation for (27a) proceeds in exactly the same way as does a derivation for (27b). It begins by merging the verb *likes* with its object *whom* in (28a)—the verb will check the thematic feature of *whom* and, following

Epstein, Groat, Kawashima, and Kitahara (1998), it will also check the agreement features of *whom* (though it also possible, as Chomsky (1995) argues, that these features are checked in vP). At this point in the derivation, all the features of *whom*, except for its [DISC] feature, will be appropriately checked. Since the [DISC] feature, as a nonconcatenation feature, need not be checked, the SO *whom* will have no features that survive in the derivation and, as a result, *whom* will not remerge in the course of the derivation. After *whom* merges with the verb in (28a), the light verb **v** will merge (see (28b)); the subject will merge in (28c), checking the thematic feature of *Chris* but not its agreement features, which survive; the tense head *T* will merge in (28d); and finally the subject *Chris* will remerge to have its agreement features checked (see (28e)).

(28) a. Merge ⟨likes, whom⟩ → likes whom
b. Merge ⟨v, ⟨likes, whom⟩⟩ → v likes whom
c. Merge ⟨Chris, ⟨v, ⟨likes, whom⟩⟩⟩ → Chris v likes whom
d. Merge ⟨T, ⟨Chris, ⟨v, ⟨likes, whom⟩⟩⟩⟩ → T Chris v likes whom
e. Remerge ⟨Chris, ⟨T, ⟨Chris, ⟨v, ⟨likes,whom⟩⟩⟩⟩⟩ → Chris T Chris v likes whom

In the course of the derivation terminating in (28e), all the concatenating features will be checked. Therefore, (28e) will not stall or abort; rather, it will serve as the representational input that is interpreted at the interfaces. At the sensorimotor interface, *whom* will be spelled out in its merged position and *Chris* will be spelled out in its remerged position (i.e., the positions where the relevant morphophonetic features are checked in the derivation). And at the conceptual-intentional interface, both *whom* and *Chris* will have their thematic features interpreted in their checked (merged) positions; in addition, *whom* will have its [DISC] feature interpreted by assigning it a discourse referent generally taken from the discourse preceding the "Chris likes whom" utterance. Under the foregoing analysis, the [DISC] feature, though interpreted at the conceptual-intentional interface, is exempted from syntactic checking. Importantly, not having to be checked permits constituents with a [DISC] feature to distribute freely in the syntax; hence, *wh*-echo constituents can appear widely, as the examples in (20) and (21) illustrate.

In addition to the *wh*–in situ elements that appear in (20)–(23), there are also *wh*–in situ elements that pair with *wh*-operators, and these pairs can be given pair-list interpretations. We can see such *wh*–in situ elements in (29).

(29) a. Who read what
b. What did Pat give to whom
c. Who expects whom to win
d. Who knows how to do what
e. Who knows what Pat read to whom

A possible pair-list interpretation (actually, a set of ordered-pair responses) for (29a) is given in (30).

(30) Pat read a Poe novel; Chris read the *Koran*; and Sam read Shakespeare's *Much Ado About Nothing*

Importantly, although, as the data in (29) suggest, the *wh*–in situ elements that form ordered pairs with *wh*-operators can appear in a wide range of constructions—both in clausemate constructions with *wh*-operators (29a–b) and in nonclausemate constructions (29d–e)—these *wh*-elements differ significantly from *wh*-echo elements in their distribution. Unlike *wh*-echo elements that can show up in every conceivable grammatical construction, paired *wh*–in situ elements have a restricted distribution. For one, these elements must be paired with referential *wh*-operators: they obviously cannot appear unpaired, as in (31a) and they cannot be paired with nonreferential operators [OP], as in (31b).

(31) a. *Pat read what? (ungrammatical on a nonecho reading)
b. *Pat knows how to do what? (ungrammatical on the nonecho reading)

Of course, (31a) and (31b) are acceptable as echo questions. In these cases, *what* has a [DISC] feature and is analyzed as previously discussed. A more interesting situation arises in (32).

(32) Who knows what had happened to whom

In (32), the *wh*–in situ element *whom* could be paired with either the *wh*-operator *who* or the *wh*-operator *what*. Of note, the operator *what* could be a referential operator [OP, REF], having both an operator feature and a referential feature, or it could be a nonreferential operator [OP]. If it is an [OP, REF] operator, then (32) could be answered as in (33a); if it is an [OP] operator, then (32) could be answered as in (33b).

(33) a. Pat knows something ominous had happened to Sam
b. Pat knows what had happened to Sam

Whenever *what* is a nonreferential [OP] operator, it cannot be paired with the *wh*–in situ element *whom,* as the response given to (32) in (34) demonstrates, because it cannot form an ordered referential pair with the *wh*-in situ element.

(34) #Pat knows what had happened to whom

The fact that (31a) is ungrammatical and that (34) is an unacceptable response to (32) suggests that *wh*–in situ elements will pair with other *wh*-elements if and only if these elements are referential *wh*-operators [OP, REF]. No such distributional constraint applies to *wh*-echo elements—hence the permissibility of an echoic reading for (31b).

Another distributional difference between *wh*-echo elements and paired *wh*–in situ elements is that the former can appear in "movement" islands, as in (21a–c), while the latter cannot, as the examples in (35) demonstrate.

(21) a. Pat likes books that criticize who
b. Pat was happy after Mary fired who
c. Pat likes Bob and who

(35) a. *Who likes books that criticize who
b. *Who was happy after Mary fired who
c. *Who likes Bob and who

I have marked the examples in (35) as ungrammatical under a pair-list reading. Needless to say, the examples in (35) can be grammatical under a single-pair reading—Dayal (2002) makes a similar observation. Such single-pair readings arise, according to Dayal, only if the *wh*–in situ elements are nonoperators that are interpreted contextually, perhaps via Reinhart's (1998) choice function. In other words, the *wh*–in situ elements in (35) will participate in single-pair interpretations only if their [WH] features are not [OP] features but are features that will be interpreted contextually at the conceptual-intentional interface. These features are so hauntingly reminiscent of the [DISC] features of *wh*-echo elements that it is difficult not to attribute the single-pair readings of the sentences in (35) to the presence of the [DISC] feature on the *wh*–in situ elements. Although it might seem to be a leap to explain both the *wh*-echo sentences in (21) and the single-pair interpretations for the sentences in (135) along similar lines—by assuming the presence of the [DISC] feature—there is one exceedingly strong reason for doing so: this explanation will provide a natural account of the similarly licentious distributions of *wh*-echo sentences and of multiple-*wh* sentences with single-pair interpretations.

The fact that the sentences in (35) can be given single-pair interpretations is important because, as we have just argued, it provides additional support for the [DISC] feature required to explain *wh*-echo constructions; however, the fact that these sentences cannot be given pair-list interpretations is equally important because it separates some multiple-*wh* constructions, such as those in (35), from other multiple-*wh* constructions, such as those in (29), thereby raising the possibility that not all *wh*–in situ elements have exactly the same interpretive features. It is certain, at the very least, that not all *wh*-in situ elements have a contextually interpreted [DISC] feature that prevents the in situ elements from being (interpretively) bound by any (*wh*) operator—for if all *wh*–in situ elements did have a [DISC] feature, then all multiple-*wh* constructions would necessarily have only single-pair interpretations linking a single operator value to a single contextual value assigned to the *wh*–in situ element. Since multiple-*wh* constructions such as those in (29) can be given pair-list interpretations, this suggests that the *wh*–in situ elements in these constructions do not carry a [DISC] feature.

That *wh*–in situ elements in multiple-*wh* constructions cannot be interpreted contextually if they participate in pair-list readings, squares with my previous observation that these in situ elements are interpretively dependent on another *wh*-element in the construction. I have, in fact, argued that these *wh*–in situ elements must be interpreted as the dependent-variable part of ordered pairs they form with referential *wh*-operators: they are *wh*-variables whose referential values are dependent on the referential values assigned to the *wh*-operators. If this is correct, then the interpretation of such *wh*–in situ elements cannot be determined by extrastructural, contextual (or discourse) features; rather, the interpretation of *wh*–in situ elements must be determined by the structural relations that the *wh*-in situ elements have with their paired *wh*-operators. We can see the importance of the structural relations established between *wh*–in situ elements and the *wh*-operators on which they depend for their reference in (36).

(36) Which woman knows what which man bought

Notice that although the *wh*–in situ element *which man* in (36) is structurally related to two *wh*-operators (*which woman* and *what*), it can be interpretively dependent only on one of the operators (*which woman*). What this demonstrates is that *wh*–in situ elements are referentially dependent on some, but not all, of the *wh*-operators in a sentence. If we are to explain *wh*–in situ elements, we must explain how their interpretive dependency on *wh*-operators is structurally shaped.

We can get some insight into the structural determinants responsible for the interpretive dependencies of *wh*–in situ elements if we look at another interpretively dependent element: the reflexive. As with *wh*–in situ elements, reflexives such as *herself*, *himself*, and *myself* can be interpreted either contextually (logophorically) or structurally. The former interpretation shows up in (37a), the latter in (37b).

(37) a. As for myself, spaghetti would be fine
b. Chris likes herself

Since I am investigating the relationship between structure and interpretation, I will consider examples such as (37b), but will not comment on examples with logophoric reflexives such as (37a). In (37b), the reflexive *herself* is interpretively linked to, and dependent on, its antecedent *Chris*. The interpretive dependence that the reflexive *herself* has on its antecedent, importantly, is mediated by Agreement features. That is, a reflexive can be linked to an antecedent only if the reflexive and its antecedent have the same Person, Number, and Gender features. Without such appropriate agreement, a reflexive cannot be licensed for interpretation, as we can observe in (38).

(38) That woman likes
*himself/*itself/*myself/*yourself/*themselves/*ourselves/
*yourselves

In (38), none of the reflexives shares the Person, Number, and Gender features of the DP *that woman*; hence none of the reflexives can take this DP as their antecedent, thereby leaving every reflexive in (38) without any available antecedent. The reflexives in (38), consequently, are not licensed for interpretation. Given that the reflexives in (37b) and (38) are all intrinsically marked with Person, Number, and Gender features that can be readily interpreted at the appropriate conceptual-intentional and sensorimotor interfaces, the licensing of reflexives for interpretation must involve something beyond the mere syntactic presence of these features, something that marks reflexives as being agreement-dependent elements [AGR/]. Let's suppose that reflexives have the [AGR/] feature. If a reflexive (excluding logophorically interpreted reflexives, which arguably possess a [DISC] feature) has an [AGR/] feature, it will have to have this feature checked by a head H with an AGR feature, and its Person, Number, and Gender features will have to agree with those of the antecedent DP in SpecHP that checks the agreement features of H. In other words, the reflexive must end up in an HP that includes an [AGR] head H and a DP that both checks the agreement features of H and serves as an antecedent for the reflexive, as in (39). It is important to note that reflexives, as agreement-dependent elements, cannot check the agreement features of

H. This accords with Rizzi's (1990), Woolford's (1999), and Haegeman's (2004) claims that anaphors cannot check agreement features; rather, reflexives must, to be properly interpreted, (dependently) agree with the features of H, which are (independently) checked by another element (the antecedent of the reflexives).[3]

(39) [DP [Reflexive H . . .]]

To see how this works, let's consider the sentences in (40).

(40) a. Chris likes herself
b. *Herself likes Chris
c. *Mary believes that Bill likes herself
d. *Mary believes that herself likes Bill

Sentence (40a) will be derived as follows. First the reflexive *herself* will merge with the verb *likes*; the verb will check the thematic role of the reflexive, but not its Case or [AGR/] features, which will SURVIVE. Since the reflexive has surviving features, the reflexive will have to automatically remerge once the light verb v is merged into the derivation (after the reflexive remerges, the subject argument *Chris* will merge and its thematic role will be checked by v). The light verb will be able to check the Case feature of *herself*, but not its [AGR/] feature—because the light verb in English lacks an [AGR] feature, as is suggested by the fact that objects of verbs do not have their Person, Number, and Gender features checked in English.[4] The [AGR/] feature on the reflexive, then, continues to SURVIVE. Consequently, the reflexive must remerge after the next head (the Tense head) is merged into the derivation. The T head has an [AGR] feature, so it can check the [AGR/] feature of the reflexive; furthermore, the subject argument *Chris* will remerge in SpecTP, checking the agreement features of T and licensing the agreement features of the reflexive for interface interpretation in the structure derived in (41).

(41) [Chris [herself T . . .]]

Whereas it is possible to derive a licit representation for (40a), it is impossible to do so for (40b), (40c), and (40d). In (40b), the object argument *Chris* will have its thematic feature checked by the verb *likes* and its Case feature checked by the light verb v. Meanwhile, the reflexive *herself* will have its thematic feature checked by the light verb, and its [AGR/] feature checked by T. Unfortunately, since the Person, Number, and Gender features of T cannot be checked by the [AGR/] reflexive and since the agreement features of the reflexives cannot be licensed by T, the final derivation for (40b)—see (42)—cannot help but crash.

(42) [herself [T . . .]]

The derivation of (40c) will also stall or crash, but for different reasons than above. Its derivation will reach a point where the embedded sentence will look as follows:

(43) [Bill [herself T . . .]]

In the embedded TP, the [AGR/] feature of the reflexive *herself* is not incompatible with the [AGR] feature of the head T and as a result, it is deactivated and will not SURVIVE. This means that none of the concatenation features of the reflexives—the thematic features, Case feature, and [AGR/] feature—will SURVIVE the embedded TP; consequently the reflexive will not have any surviving features to require subsequent Remerge. The reflexive then will be stranded within the embedded TP, where it cannot have its Person, Number, and Gender features licensed for interpretation since these features do not agree with the agreement features of T (or of the available DP antecedent *Bill*). Example (40d) has a similar story. As with (40b) and (40c), the reflexive in (40d) will remerge in the embedded TP. The T head will check the [AGR/] feature of the reflexive; however, the head will not be able to license the agreement features of the reflexive for interpretation. Given that all the concatenation features of the reflexive will have been checked by various heads within the embedded TP, the reflexive will not have any surviving features that can motivate Remerge. The reflexive must remain within the embedded TP, and its agreement features will remain unlicensed for interpretation.

The analysis of reflexives and agreement sketched above predicts that [AGR/] feature of reflexives will not be able to SURVIVE any head H that carries an [AGR] feature itself because the two [AGR]-type features are not incompatible. What this suggests is that languages in which verbs have object agreement will not permit object reflexives to find subject antecedents. Such reflexive objects will have their thematic, Case, and [AGR/] features checked/deactivated within vP (or perhaps AGROP); not having any surviving features to compel Remerge, the reflexive will be stranded within the vP, where it cannot have its Person, Number, and Gender features licensed by any distal antecedent in the subject position. My prediction, importantly, is borne out. According to Woolford (1999), languages with verb-object agreement will not allow object reflexives, as is illustrated in (44), an example from Inuit that Woolford has taken from Bok-Bennema (1991).[5]

(44) *Hansiup immi asap-puq
Hansi (ERG) himself (ABS) wash-IND.3SG.3SG
'Hansi washed himself.'

Woolford finds similar examples in other object agreement languages, such as Swahili and Nez Perce. Needless to say, the fact that verb-object agreement prevents subjects from serving as antecedents for object reflexives provides significant support for my analysis.

I have undertaken the foregoing lengthy discussion of the agreement-dependent features of reflexives to provide a model for how to explain feature-dependent elements. As we recall, the *wh*–in situ element in a multiple-*wh* construction with a pair-list interpretation is reference-dependent on some *wh*-operator. If we adopt the reflexive model to account for *wh*–in situ elements in sentences such as in (45), we can offer the following explanation (note: this discussion builds on the analysis of multiple-*wh* constructions I presented in chapter 2).

(45) Who read what

Being referentially dependent on a *wh*-operator, the *wh*–in situ element *what* in (45) will have a reference-dependent feature [REF/], perhaps even a reference-dependent-on-*wh* feature [REF/WH]. In the derivation for (45), the *wh*–in situ element *what* will first merge with the verb *read*, which will check the thematic feature of *what*. However, the verb will not be able to check the Case feature or the [REF/WH] feature, which will SURVIVE. Consequently, the *wh*–in situ element must remerge once the light verb v merges into the syntax. The light verb will check the Case feature of the *wh*–in situ element, but not its [REF/WH]—the light verb will also check the thematic feature of the *wh*-operator *who* (see (46a) for this syntactic derivation).

(46) a. [who [what [v [read what]]]]

Since both *what* and *who* will have unchecked concatenation features, these SOs will have to remerge after the Tense head is merged into the derivation, remerging in the order specified in (46b).

(46) b. [who [what [T [who [what [v [read what]]]]]]]

The Tense head will check only the Case and agreement features of *who*, leaving the [REF/WH] feature of *what* and the [WH] feature of *who* unchecked. This will require both *wh*-elements to remerge once the C[WH] head is merged (see (46c)).

(46) c. [who [what [C [who [what [T [who [what [v [read what]]]]]]]]]]

In (46c), the C head will not only check the [WH] and [REF] features of the referential operator, but also deactivate the [REF/WH] feature of the *wh*–in situ element, because this C head will not be incompatible with this [REF/] feature. All the concatenation features of the SOs in (46c) will be checked/deactivated; as a result, (46c) will be a licit representation that will be interpreted at the conceptual-intentional and sensorimotor interfaces. The [REF/WH] feature of the *wh*–in situ element will require this *wh*-element to be interpreted as the dependent part of an ordered pair with a referential *wh*-operator. Hence, *who* and *what* will be interpreted as ⟨who, what⟩.

Notice, then, that reflexives and *wh*-in situ elements have their dependency features checked/deactivated in similar structures (47a–b) and that they receive similarly dependent interpretations.

(47) a. [DP [reflexive [H[+AGR] . . .]]]
     b. [*Wh*-operator [*wh*–in situ [C[+WH,+Ref] . . .]]]

Both reflexives and *wh*–in situ elements are immediately c-commanded by the antecedent elements on which they depend for their interpretations, and they both have their dependency features deactivated by heads that check the relevant features of the antecedents.

Now if my feature-based analysis of *wh*–in situ elements is on the right track, then I would make the following predictions about multiple-*wh* constructions. First, I predict that *wh*–in situ elements in pair-list multiple-*wh* constructions must be referential—that is, nonreferential *wh*-elements such as *why* and *how* will not be able to be *wh*–in situ elements in such constructions. The examples in (48) corroborate this prediction.

(48) a. *Who is sleeping why
     b. *Who fixed the bike how

Second, I predict that the *wh*-operators in pair-list multiple-*wh* constructions must also be referential elements, or else the [REF/WH] feature of the *wh*–in situ element will not be checked. In other words, the *wh*-operators in pair-list multiple-*wh* constructions will not be able to be nonreferential operators—*why* or *how*. We can see that this prediction is confirmed by the data in (49).

(49) a. *Why did Pat eat what
     b. *How did Chris chase who

Third, I predict that the [REF/WH] feature of a *wh*-in situ element can be deactivated by any head H that has a nonincompatible [REF] feature; this could leave the *wh*–in situ element stranded in HP and possibly not able to satisfy its operator dependency. Since DPs and CPs are the only maximal projections that might bear a [REF] feature, I predict that when these projections are referential, they will deactivate the [REF/WH] feature and prohibit *wh*–in situ elements from having their dependency feature appropriately checked. The data in (50) test this prediction for DPs.

(50) a. Who took a picture of whom
b. *Who took the picture of whom
c. *Who took a certain picture of whom

In (50a), the DP is not referential; therefore, the *wh*–in situ element will not have its [REF/] feature deactivated by the head D. This feature will SURVIVE until it reaches the matrix CP, where it will be checked by the C[WH,REF] head. In (50b) and (50c), the DPs are headed by D[REF] heads that will deactivate the [REF/] features of the *wh*–in situ elements. The *wh*–in situ elements will, consequently, not have any features that SURVIVE the DP and these elements will be stranded in the DPs. Being so stranded, the *wh*–in situ elements will not have their [REF/WH] features appropriately checked, and any derivation for (50b) and (50c) must crash or stall. Similar results emerge for CPs. Referential CPs—for example, declarative CPs headed by *that*—do indeed prevent a *wh*–in situ element from linking appropriately with a *wh*-operator. We can observe this in (51). (The grammaticality judgments in (51) reflect Dayal's (2002) judgments of permissible pair-list readings.)

(51) a. *Who believes that who left
b. *Who believes that Chris read what

In (51a) and (51b), the [REF/] features of the *wh*-in situ elements will not SURVIVE the embedded CP; therefore these elements will remained stranded in the embedded CPs, without any way to have their [REF/] features checked. Although a referential CP will strand a *wh*–in situ element, a nonreferential one will not. Neither CPs with nonreferential operators in SpecCP (see (52a) and (53a)) nor infinitival CPs, which lack referential Tense (see (53a)), will strand *wh*–in situ elements.

(52) a. Who knows how Sam fixed what
b. *Who know that Sam fixed what
c. *Who said that Pat knows how Sam fixed what

(53) a. Who wants Pat to read what
b. Who told Sam why Pat expects to read what

It is important to note that the *wh*–in situ element *what* in (52b) cannot be part of a pair-list interpretation when it is contained in an embedded CP headed by *that*; on the other hand, the *wh*–in situ element in (52a) can participate in a pair-list reading when it is contained in an embedded CP with a nonreferential *wh*-operator. Interestingly, as (52c) demonstrates, if the *wh*–in situ element is separated from a referential operator by a nonreferential CP and a referential CP, it will not be able to participate in a pair-list reading with the *wh*-operator. Further, the *wh*–in situ element in (53a) can participate in a pair-list interpretation even though it is contained in an infinitival CP, and if a *wh*-in situ element is separated from a *wh*-operator by two nonreferential CPs (as in (53b)), it will still be able to participate in a pair-list reading. That my feature-based analysis of the *wh*–in situ elements can account for all the data in (50)–(53) offers valuable support for my analysis. The final prediction I make about *wh*–in situ elements is that they cannot participate in pair-list readings from within relative clauses—a prediction corroborated by the data in (54), taken from Dayal 2002.

(54) *Which student read the book that which professor wrote

Since relative clauses are referential, as we can see from the fact that nonreferential DPs cannot be the antecedent for relative pronouns (this is shown in (55)), any *wh*–in situ element within a relative clause will be stranded in the relative clause and will not be able to be linked to a *wh*-operator outside the relative clause.

(55) *No one, who I like a great deal, left

As the evidence in (48)–(54) illustrates, the [REF/] feature of a *wh*–in situ element must be checked by a head H that is both [WH] and [REF], but it can be deactivated by any head that is [REF].

At this point in my analysis of *wh*-elements, I have proposed that echo-type *wh*–in situ elements have a [DISC] feature and that pair-list-type *wh*–in situ elements have a [REF/] feature. The one *wh*-element we must still analyze is the *wh*-operator. Quite simply, all *wh*-operators have some sort of operator feature ([OP] or perhaps [*WH*-OP]), which must be checked by a head with an equivalent feature. In (56), the *wh*-operator *who* has a thematic feature, a Case feature, and an [OP] feature.

(56) Who did Pat hire

The thematic feature of *who* will be checked by the verb *hire* and its Case feature will be checked by the light verb v; the [OP] feature, however, will

SURVIVE each and every head until the C head. Hence, the *wh*-operator *who* will merge and/or remerge with every head in its derivation. When it remerges with the C head, the [OP] feature of *who* will finally be appropriately checked. The [OP] feature can SURVIVE an indefinite number of nonoperator heads. For this reason, a *wh*-operator can exhibit long-distance displacement, as (57) illustrates.

(57) Who did Pat tell you that Chris expects me to talk to tomorrow

In (57), the *wh*-operator *who* first merges as an argument of the verb *talk*, but it eventually appears in the SpecCP position of the matrix sentence, where its [OP] feature will finally be checked. Since there are no operator heads between the merged position of *who* and its position in the matrix sentence, the [OP] feature of *who* will not be stranded before it can be checked in the matrix CP. However, should a *wh*-operator remerge with any head with a nonincompatible [OP] feature, the *wh*-operator could be stranded and not appropriately checked. Such a situation arises in the examples in (58).[6]

(58) a. *What did Chris tell Bill who was reading
b. *What does Chris know who Bill was reading to
c. *What does Chris wonder how Bill read
d. *What did Chris tell Bill why to read

The *wh*-operator *what* in all the sentences in (58) has the same derivational history. It first merges into the syntax as an argument of the embedded verb *read*, where the verb can check some of the features of *what*, but not its [OP] features. As a result, *what* must remerge in the syntax to have its [OP] feature checked. In (58a–d), the *wh*-operator *what* attempts to undergo serial remerger until it reaches the matrix CP and has its [OP] feature checked by the [OP] feature of the head C of the CP. Now it is certainly the case that a *wh*-operator can iteratively remerge from an embedded position until it makes its way to a matrix CP; examples of this can be seen in (59).

(59) a. What did Chris tell Bill Sam was reading
b. What does Chris know Bill was reading to Sam

Although the *wh*-operator *what* can licitly wend its way to the matrix CP in (59), it cannot in (58) because the embedded CP has an [OP] feature that does not show up in the embedded CP in (59), and this [OP] feature is not incompatible with the [OP] feature of the *wh*-operator *what*. The fact that the embedded CP has an operator head in (58a–d) means that no *wh*-operator within this CP will have its [OP] feature SURVIVE beyond the CP. Since all of the examples in (58) have embedded CPs that contain two *wh*-operators, both of these *wh*-operators must eventually remerge in the embedded CP, where the C head can check the [OP] feature of one of these operators, though not the [OP] feature of the second operator (see Bošković 1999 for a discussion of the single-checking property of C heads in English). The [OP] feature of the second operator in (58a–d), then, cannot be checked in the embedded CP, nor can it SURVIVE the CP because it is not incompatible with the operator feature of the C head. Consequently, the second operator must be stranded in the embedded CP and its [OP] feature will remain unchecked. Having an uncheckable [OP] feature in the embedded CP will force derivations for (58a–d) to stall once the embedded CP is completed and to abort at that point in the derivation. Absent a completed derivation, none of the sentences in (58) can have an LF or a PF representation—that is, the sentences simply cannot be generated.

Similar results emerge in relative clauses. These clauses are CPs with C heads that carry [OP] features, which allows the C heads in relative clauses to check the [OP] features of relative pronouns, as in (60).

(60) a. Chris met a politician [whom she likes a great deal]
b. They found a place [where they could be happy]

The relative pronouns *whom* in (60a) and *where* in (60b) have their [OP] features checked by the C heads in the CPs of the bracketed relative clauses. Given that the C head in CP relative clauses has an [OP] feature, I predict that this head will prevent any *wh*-operator within the clause from remerging beyond the CP—that is, I predict that *wh*-operators will not be permitted to escape relative clauses. The data in (61) test this prediction.

(61) a. *What did Chris see a woman who was reading
b. *Whom does Chris like the book that Sam was reading to
c. *What do you know the best place to eat in

As I have predicted, the *wh*-operators *what* and *whom* in (61a–c) cannot remerge outside the relative clauses that they are merged in.

Interestingly, and importantly, there are other heads H that have [OP] features, and these heads all disallow *wh*-operators from escaping their HPs (similar intervention effects are discussed in Beck 1996, Pesetsky 2000, and Soh 2005). One such head is the logical operator *only*. This logical operator has an [OP] feature that is not incompatible with the *wh*-operator feature; however, although *only* has an [OP] feature, it is not a [OP-WH] feature and therefore cannot check a *wh*-operator feature. Hence, the [OP] feature of *only* will deactivate a *wh*-operator feature, thereby preventing it from surviving, but it will not check the *wh*-operator feature. As a result, no *wh*-operator will be able to escape an XP headed by *only*, as is demonstrated in (62).

(62) a. *Whom did Pat write [only to]
cp Whom did Pat write to
b. *What was Chris reading [only about]
cp What was Chris reading about
c. *Who does Sam have [only pictures of]
cp Who does Sam have pictures of

The presence of the logical operator *only* prohibits *wh*-operators from escaping their Merge sites within the *only*-Phrases. Operator heads like *whether* and *if* also strand *wh*-operators merged within their phrases and they do so for the same reasons that the operator *only* does—they have [OP] features that are not only incompatible with *wh*-operator features, but are also incapable of checking *wh*-operator features. The examples in (63) show that *wh*-operators cannot, in fact, escape phrases headed by *if* or by *whether*.

(63) a. *Who will Chris tell Pat [if Sam hires]
b. *What does Chris wonder [whether Pat will buy]

Likewise, other heads with [OP] features strand *wh*-operators merged in their phrases. These operators include temporal operators such as *before*, *after*, *while*, and *when* among others (see (64)), logical operators such as *and* and *or* (see (65)), and most other subordinating operators such as *unless* and *because* (see (66)).

(64) a. *What did Chris see Sam [before/after/while/when he read]
b. *Who did Chris hire Sam [before/after/while/when speaking with]

(65) a. *Who does Chris like [Sam and/or]
b. *Who does [Chris like Sam and/or Pat dislikes]

(66) a. *What will Chris fired Sam [unless Pat reads]
b. *Who does Chris admire Sam [because he fired]

It appears generally that heads H with operator features will not allow *wh*-operators to escape their HPs because the [OP] features of *wh*-operators are not incompatible with the [OP] features of these heads.

There does, however, seem to be one operator head that does not uphold the above generalization: the NEG(ation) head. As is well known, the NEG operator does not necessarily strand *wh*-operators (see (67)).

(67) a. What won't you do for me
b. Who will Chris not hire
c. Who doesn't John have a picture of

Although NEG does have an operator feature, it has a special variety of the [OP] feature, one that checks only nonreferential operators [OP, −REF]. Support of my claim that NEG has an operator feature that is incompatible with referential *wh*-operators (conversely, it is compatible only with nonreferential *wh*-operators) comes from three sources. First, a negated DP such as *no one* cannot be the antecedent for a nonrestrictive relative pronoun, as we can observe in (68).

(68) a. *No on, whom Chris likes, left
b. *Chris will hire no one, who Sam just hired

As the data in (68) suggest, the referentiality of the nonrestricted relative pronoun is not compatible with the NEG element. Second, despite the fact that a NEG head cannot strand referential *wh*-operators, as in (67), it can strand the nonreferential *wh*-operators *how* and *why* (see Cinque 1990 and Rizzi 1990 for arguments that *how* and *why* are nonreferential). I provide relevant examples in (69).[7]

(69) a. *How will Chris not read the book
cp How will Chris read the book
b. *This is why I don't believe that Chris was fired (why)
cp This is why I believe that Chris was fired (why)

Similarly, as Soh (2005) observes, in Mandarin Chinese a NEG head strands the nonreferential *wh*-operator *weishenme* ('why'), but not the referential operator *shenme* ('what'), as is illustrated in (70).

(70) a. *Ni bu renwei Lisi weishenme kan zhentan-xiaoshuo
you not think Lisi why read detective-novel
'Why don't you think Lisi reads detective novels?'
b. Ta bu mai shenme
he not sell what
'What didn't he sell?'

And third, *wh*–in situ elements, which are also nonreferential operators, are stranded by NEG heads. We can see this in both English (see (71)) and in German (see (72), taken from Beck 1996).

(71) a. Who the hell is going where
b. *Who the hell isn't going where

(72) a. Wen hat Luise wo gesehen
Whom has Luise where seen
'Where did Luise see whom?'
b. *Wen hat niemand wo gesehn
whom has nobody where seen
'Where did nobody see whom?'

Let's assume, in the face of the foregoing discussion, that the NEG head has [OP, −REF] features that are incompatible with other [OP] features. This incompatibility will explain why the referential *wh*-operators in (67) are not stranded by NEGPs and why the *wh*-elements in (69), (70a), (71b), and (72b) are stranded.

What we have seen in this section is that *wh*-elements can have three different types of *wh*-features—a [DISC] feature, a [REF/] feature, or an [OP] feature—and that these features play significant roles in the interpretive and distributional properties of *wh*-constructions.

Superiority Effects

Since Kuno and Robinson (1972, 474) first observed that "a wh-word cannot be preposed, crossing over another wh (element)," there has been a spate of analyses that have attempted to account for this so-called superiority effect. The Superiority Effects illustrated in (73b) and (74b) show that in multiple-*wh* constructions, *wh*-objects cannot cross over their structurally *wh*-superior subjects.

(73) a. Who saw what
b. *What did who see

(74) a. Chris knows who bought what
b. *Chris knows what who bought

In the aftermath of May's (1977) arguments for a syntactic theory of logical form that permitted the LF movement of *wh*–in situ elements, data such as those in (73) and (74) seemed to suggest that superiority could be reduced to subject/object asymmetries on LF movement. Assuming—as did Huang (1982), Pesetsky (1982), May (1985), Lasnik and Saito (1984), Aoun (1985, 1986), and Rizzi (1990), among others—that the sentences in (73a) and (73b) have logical representations (75a) and (75b) respectively, the aforementioned theorists posited various Path Containment conditions, Empty Category Principle conditions, and Binding Principle conditions to rule out the LF movement of the *wh*-object *what* in (75b).

(75) a. [what$_i$ who$_j$ [t$_j$ saw t$_j$]]
b. *[who$_i$ what$_j$ [t$_i$ saw t$_j$]]

Under these analyses, the LF movement of *what* in (75a) is licit because it leaves behind a trace t_i that is lexically licensed by its lexical governing head *saw*, thereby satisfying the government conditions of the Empty Category Principle (ECP) and thereby permitting the operator *what* to properly bind its variable trace (see Huang, Lasnik and Saito, and Rizzi for the details of this sort of analysis). Or the LF movement of *what* is licit because the paths connecting the operators and their variable satisfy the Path Containment Condition (PCC) by not having overlapping operator-trace chains (see Pesetsky and May for this sort of analysis). On the other hand, the LF movement of *who* in (75b) is illicit because its trace t_i is neither lexically licensed by a lexical governing head nor antecedent governed (the intervening *wh*-operator *what* prevents the *wh*-operator *who* from being a local antecedent governor for the trace), which leaves the ECP unsatisfied and the *wh*-operator *who* without a trace variable to bind; or because the operator-trace chains overlap in (75b) in violation of the PCC; or because the intervening *wh*-operator *what* keeps the *wh*-operator *who* from binding its trace (see Aoun), thereby not satisfying conditions on Generalized Binding.

Although the LF-movement analyses of superiority discussed above can offer explanations for the data in (73) and (74), they have significant problems accounting for the examples in (76)–(78).

(76) a. Who did you persuade to buy what
b. *What did you persuade who to buy

(77) Who knows what who bought

(78) Who knows what who told Sam to buy for whom

As Hendrick and Rochemont (1982) point out, the sentences in (76) present problems for ECP-style analyses of superiority. Given that the *wh*-operators in (76a) and (76b) are both *wh*-objects, ECP analyses should predict that (76a) and (76b) should be equally grammatical since the LF movements in these examples will leave behind traces that satisfy the ECP (these traces will be lexically governed). That there is a grammaticality difference between (76a) and (76b), then, jeopardizes ECP analyses of

Superiority Effects. An even graver problem for LF-movement analyses of superiority comes from sentences such as (77) and (78). If (74b) is ungrammatical because at LF the *wh*-subject *who* crosses over the fronted *wh*-object *what*, thereby violating the ECP, the PCC, and Generalized Binding, then (77) and (78) should involve an even more egregious violation of grammaticality because at LF the embedded *wh*-subject *who* in these examples has to cross over the fronted *wh*-object *what* and the matrix *wh*-subject *who* (in order to account for the fact that the embedded *who* must have matrix scope in (77) and (78)). Needless to say, the grammaticality of (77) and (78) seriously undermines the validity of the various LF-movement analyses of superiority that I have discussed.

Since Chomsky's (1995, 2000b) Minimalist Program no longer separates S-structure movement from LF movement and since this Minimalist Program no longer assumes the ECP, it would appear to be poised to offer an analysis of superiority that is not hamstrung by LF movement. Unfortunately, current minimalist analyses of superiority fare no better than have LF-movement analyses, as we will see. Most minimalist explanations of superiority (see Chomsky 1995, Richards 1997, 2001, Pesetsky 2000, Radford 2004, and others) argue that the sentences in (73b) and (74b) are ungrammatical because they violate the Attract Closest Principle (79), taken from Radford (2004, 162).

(79) *Attract Closest Principle*
A head that attracts a given kind of constituent attracts the closest constituent of the relevant kind.

According to the Attract Closest Principle, the C [WH] head in (73) and (74) must attract the closest *wh*-element to fill the matrix SpecCP position. In both (73) and (74), the closest *wh*-element to the C head is the *wh*-subject; hence, the *wh*-subject can fill the SpecCP position (as in (73a) and (74a)), but the more distant *wh*-object cannot (as in (73b) and (74b)). Importantly, not only can the Attract Closest Principle account for the data in (73) and (74), it can also explain the grammaticality differences in (76). Example (76a) is grammatical, while (76b) is not, because the matrix *wh*-object is closer to the matrix C [WH] head than is the embedded *wh*-object and, consequently, only the former can be attracted to the matrix SpecCP position. Successful as the Attract Closest Principle is in explaining (76), it cannot explain data such as (77). In (77), the embedded *wh*-object *what* is attracted to the embedded SpecCP position over the closer *wh*-subject *who*; this should be a violation of the Attract Closest Principle. In fact, the Attract Closest Principle should allow (80), while disallowing (77). The grammaticality of both (77) and (80) compromises an Attract Closest Principle analysis of superiority.

(80) Who knows who bought what

My sense is that the Attract Closest Principle fails to explain (77) for the same reason that LF-movement analyses do: they all reduce Superiority Effects to movement phenomena, and there is simply no way to appeal to movement to discriminate the well-formed embedded clause in (77) from the ill-formed (74b). The failure of movement theories to account for Superiority Effects, however, does not guarantee the success of non-movement analyses to account for these effects. Two notable examples are Aoun and Li's (2003) and Hornstein's (1995) analyses. After arguing against restricting Superiority Effects to instances of movement, Aoun and Li propose a movement-independent explanation for superiority. They contend that superiority phenomena can best be captured by a closest-feature principle that they call the "Minimal Match Condition" (MMC), stated in (81).

(81) *Minimal Match Condition*
A *wh*-operator must form a chain with the closest XP with a [WH] feature that it c-commands.

What the MMC requires, in essence, is that a *wh*-operator not have any other *wh*-element (which includes *wh*-elements, *wh*-traces, and *wh*-resumptive pronouns) intervening between it and its variable. For Aoun and Li, (73b) and (74b) are both ill-formed because they derive (82), a structure in which the *wh*-operator *what* must form a chain with the *wh*– in situ element *who*, rather than with its trace variable. This leaves the trace variable uninterpretable and, under the Chomsky's (1995) Principle of Full Interpretation, any sentence with an uninterpreted variable will be ungrammatical.

(82) [what [who . . . t]

Aoun and Li's MMC, unfortunately, is as incapable of explaining (77) as have been all the analyses I have previously considered. The reason that the MMC cannot account for (77) is that its derivation will necessarily include structure (82) in its representation of the embedded clause. Consequently, the MMC (mis)predicts that (77) should be as ungrammatical as are (73b) and (74b). Where Aoun and Li's analysis fails to account for (77), Hornstein's (1995) analysis succeeds, somewhat. Hornstein proposes a nonmovement, binding analysis of superiority. For Hornstein, the sentences in (73) have the logical representations given in (83).

(73) a. Who saw what
b. *What did who see

(83) a. [who$_i$ [t$_i$ saw [pro$_i$ N]]
b. *[what$_i$ [. . . [pro$_i$ N] see t$_i$]]

Logical representation (83a) is well formed because (i) the *wh*-operator *who* is a quantificational set generator, (ii) the *wh*–in situ element *what* has an implicit pronoun (pro) that can interpreted as a variable pronoun bound by the *wh*-operator, and (iii) the binding relations established in (83a) do not yield a weak crossover violation.[8] On the other hand, (83b) is ill-formed, even though the *wh*-operator *what* is a set generator capable of binding the implicit pronoun in the *wh*–in situ element *who*, because the binding relations established in (83b) produce a weak crossover violation. Hornstein's analysis, interestingly enough, can account for (77). To see this, consider (84), a possible logical representation for (77).

(84) [who$_i$ [t$_i$ [knows [what$_j$ [[pro$_i$ N] bought t$_j$]]]]

In (84), the implicit pronoun is appropriately bound by the matrix *wh*-operator (this will explain why the *wh*–in situ element looks as if it takes matrix scope) without incurring a weak crossover violation involving either the matrix operator *who* or the embedded operator *what*. (Note that if the implicit pronoun were bound by the embedded operator, a weak crossover violation would emerge, which accounts for why the *wh*-in situ element cannot take embedded scope.) So far, so good. If we push Hornstein's analysis, however, we will see that it has substantial flaws in it. Under Hornstein's assumptions, sentences (85a) and (85b) will have well-formed logical representations (86a) and (86b), respectively.

(85) a. Who believes that who bought a book
b. Who believes that Chris bought what

(86) a. [who [t . . . [[pro N] bought a book]]]
b. [who [t . . . [Chris bought [pro N]]]]

Given the well-formedness of (86a) and (86b), Hornstein should predict the sentences in (85) should be in line for the same range of interpretations that (77) is. That is, (85a) and (85b) should permit a pair-list interpretation, just like (77) does. This prediction, following Dayal's (2002) judgments, does not hold because (85a) and (85b) simply cannot receive a pair-list interpretation. The fact that Hornstein's binding analysis will mispredict the interpretations of such sentences presents an enormous problem for this analysis.

I have examined thus far what sorts of explanations cannot account for Superiority Effects—those that appeal to movement, binding, and/or minimal feature match. What I will show next is how we can account for Superiority Effects, in terms of local feature checking, SURVIVE, and Remerge. But before I proceed with my arguments, let me pause to make a few remarks about sentences such as (73b).

(73) b. What did who see

Sentence (73b) is not ungrammatical. It can in fact receive a perfectly well-formed interpretation in which the *wh*-element *who* is interpreted echoically—in which case *who* has a [DISC] *wh*-feature. Sentence (73b) will receive, then, a single-pair interpretation akin to the interpretations available for (86a) and (86b).[9] Ungrammaticality judgments for (73b) are appropriate only in contexts where one seeks to give (73b) a pair-list interpretation.

With these comments in mind, let's look at (73a) and (73b) from the perspective of the *wh*-features we identified in the last section.

(73) a. Who saw what
  b. What did who see

In (73a), the *wh*-element *who* will have an [OP] feature that will check the [OP] feature of the C head. The *wh*–in situ element *what* cannot have an [OP] feature, since a C head in English can only phonetically spell out one *wh*-operator; however, it can have a [DISC] feature or a [REF/] feature. If it has the former feature, it will be interpreted contextually (perhaps echoically) and the entire sentence will receive a single-pair interpretation. If it has the latter feature [REF/]—a feature that cannot be checked or deactivated when the *wh*–in situ *what* merges with the verb *saw* and, therefore, a feature that will SURVIVE merger with the verb—the *wh*-element *what* will remerge in vP, in TP, and again in CP, where the dependent feature [REF/WH] can be checked. In the meantime, the *wh*-operator *who* will be merged in vP, where its thematic feature can be checked. Since the [OP], Case, and Agreement features of *who* cannot be checked in vP, these features will SURVIVE and *who* will remerge in TP. The Case and Agreement features of *who* will be checked in TP, though its [OP] feature will not be. Having a surviving [OP] feature, *who* will remerge in CP. Once *who* remerges in CP, (73a) will have (87) as its derivation (I am not registering any of the verb movements in (87)).

(87) [who [what [C [who [what [T [who [what [v [saw what]]]]]]]]]

Notice the following about (87). First, the *wh*–in situ element *what* is thematically interpreted in its merged position; it is morphophonetically interpreted in vP (where its Case and Agreement features are checked); and it is logically interpreted in CP, where its [REF/] feature is checked. Second, the *wh*-operator *who* is also in various positions: thematically in vP, Case- and Agreement-wise in TP, and as an operator in CP. Finally, and of special note, the Remerge operation requires the *wh*-elements *what* and *who* to appear in a fixed order that reflects the order of their merging. This ordering is of crucial importance to the logical interpretation of *what*. Recall that *what* has a referential-dependency feature [REF/], which must, as must all dependency features, be immediately c-commanded by the constituent on which it depends. That is, since *what* is referentially dependent on a *wh*-operator, it must be immediately c-commanded by a *wh*-operator. In (87), *what* has its [REF/] feature checked in the CP and *what* is appropriately c-commanded by the *wh*-operator *who*. Given that all the concatenation features of the SOs in (87) have been checked, (87) will become the representation for (73a) that is submitted to the interfaces for interpretation. The sensorimotor interface will spell out the morphophonetic features where they are checked, and the conceptual-intentional interface will spell out the meaning relations of (73a), including the ordered pair relation ⟨operator, dependent⟩ established for *who* and *what*. The Merge- and Remerge-driven derivation for (73b) will develop along similar lines. In fact, a version of (87)—call it (87′)—will also be the derivation for (73b), although this version will differ from (87) in one significant detail: in (87′) *who* will have the [REF/] feature and *what* will have the [OP] feature.

(87′) [who [what . . .]]

The fact that the referentially dependent *wh*-element *who* in (87′) is not immediately c-commanded by an operator in an operator-checking domain (the CP), leaves this [REF/] feature unchecked. Not having all concatenation features checked in (87′) means that this derivation cannot be sent to the interfaces for interpretation. Consequently, there is no sensorimotor or conceptual-intentional interpretation available for (73b) and, as a result, (73b) cannot be grammatically computed.

The same type of analyses apply to the sentences in (74) and (76), so I will not provide any extended discussion of these examples. Suffice it to say, that (74b) and (76b) are ungrammatical because, as with (73b), the [REF/] features of their *wh*–in situ elements cannot be appropriately checked. I do, however, want to discuss (77) at length given how terribly this example vexes other analyses of Superiority Effects.

(77) Who knows what who bought

The derivation of the embedded sentence in (77) goes in exactly the same way that the derivation for (73b) proceeds. That is, the embedded sentence in (77) will have (87′) as its derivation, with the *wh*–in situ element *who* remerged into the embedded CP, but remerged in such a way that *who* immediately c-commands the *wh*-operator *what*. As I have already discussed, (87′) is not in itself a well-formed derivation since the [REF/] feature of *who* cannot be checked in (87′). And if the embedded *wh*-operator in (77) were the only *wh*-operator available to check [REF/] feature of *who*, as it is in (88), there would be no way to check this [REF/] feature and, as a result, no way to derive a representation for (77).

(88) *Chris knows what who bought

Once the derivation for (77) reaches (87′), two facts drive the remainder of the derivation. First, the embedded *wh*-operator is not a referential operator. We know this because (89) is an acceptable answer to (77) and this answer does not require the embedded operator to take a referent.

(89) Chris knows what Sam bought

Furthermore, we know this because equivalent sentences can have non-referential operators as the embedded operator; see (90).

(90) Who knows why who left

If, as I have argued, the embedded C head does not have a referential feature (it only has an [OP] feature), then the [REF/] feature of the *wh*–in situ element *who* in (87′) will SURVIVE and *who* will have to remerge iteratively until this feature can be deactivated. In its quest to have its [REF/] feature checked, *who* will continue to remerge up to the matrix CP. Here the second fact about (77) takes on import. There is a second *wh*-operator in (77), the matrix *wh*-subject *who*. This operator will merge in the matrix vP and subsequently remerge in TP (to have its Case and Agreement features checked) and in CP (to have its [OP] feature checked). The derivation for (77) will eventually reach (91), a derivation in which the referential *wh*-operator *who* immediately c-commands the *wh*–in situ element *who*.

(91) [who [who . . .]]

In (91), the *wh*-operator will have its [OP] feature checked and the *wh*–in situ element will have its [REF/] appropriately checked against the referential feature of the operator. All the concatenation features of the SOs in (91) then will be appropriately checked, so (91) will be a well-formed representation that can be sent by Chomsky's (2002) Transfer operation to the interfaces for interpretation.

Some support for my foregoing analysis comes from the fact that in addition to explaining the data in (73), (74), and (76)–(78), my analysis can explain (92) and (93). (The judgments in (92) and (93) reflect the ability of these examples to allow pair-list interpretations.)

(92) a. What did Chris put where
b. *Where did Chris put what

(93) a. Who told you what Chris had put where
b. Who told you where Chris had put what

The asymmetry in (92) can be explained in the same way the asymmetries in (73) and (74) can be explained: in (92a) the *wh*–in situ element *where* can have its [REF/] features checked in the matrix CP against the referential value of a c-commanding operator *what*; in (92b), on the other hand, the *wh*–in situ element *what* cannot have its [REF/] feature checked because in the matrix CP the *wh*–in situ element will c-command the *wh*-operator *where*. Meanwhile, in both (93a) and (93b), the [REF/] features of the *wh*–in situ elements can SURVIVE the embedded CPs since the C head does not have a [REF] feature and the *wh*–in situ elements can eventually remerge in the matrix CPs, where they can be checked against the referential values of their c-commanding *wh*-operators.

Even stronger support for my analysis comes from my reanalysis of some surprising data first discussed by Kayne (1983) and then revisited by Richards (1997). Kayne observes that although the sentences with two *wh*-elements such as (94a) exhibit Superiority Effects, these effects disappear with the addition of a third *wh*-element (see (94b)).

(94) a. *What did who give to Mary
b. What did who give to whom

That (94a) has a detectable Superiority Effect (i.e., it cannot receive a pair-list interpretation) is uncontroversial. The question of how (94b) can or cannot be interpreted, on the other hand, is much more unsettled. My sense of (94b) is that it has a possible partial pair-list interpretation involving the *wh*-operator *what* and the *wh*–in situ element *whom*, but the *wh*–in situ element *who* does not participate in this interpretation, and that a Superiority Effect between *what* and *who* does exist. I suspect this to be the case for two reasons. First, *wh*-the-hell constructions are marginally acceptable in multiple-*wh* constructions without Superiority Effects (see (95a)), but quite bad in multiple-*wh* constructions with detectable Superiority Effects (see (95b)).

(95) a. Who the hell is engaged to whom
b. *Who the hell is who engaged to

Using *wh*-the-hell constructions as a test for superiority, we can observe from example (96) that (94b) fails the *wh*-the hell test, suggesting that it still has a Superiority Effect in it.[10]

(96) *What the hell did who give to whom
cp Who the hell gave what to whom

Second, if adding *wh*-elements could remediate constructions exhibiting Superiority Effects, then (97a) and (98a) should be remediated by adding the *wh*–in situ elements in (97b) and (98b).

(97) a. *Whom did Chris read what to
b. *Whom did who read what to

(98) a. *Whom did who read the Koran to
b. *Whom did who read what to

The fact that (97b) and (98b) are as ill-formed as (97a) and (98a) merely complicates any analysis of (94b). Importantly, my analysis of superiority can explain the data in (94b), (97), and (98). Under my analysis of (94b), the *wh*–in situ subject *who* cannot have a [REF/] feature because if it did and if the *wh*–in situ element *whom* also had a [REF/] feature, then the matrix CP would have to have the following structure.

(99) [who [what [whom . . .]]]

In (99), the [REF/] feature of *whom* can be checked against the referential features of its c-commanding operator *what*, but the [REF/] feature of *who* cannot because the referentially dependent *who* c-commands its referential anchor, the operator *what*. This means that *what* and *whom* can receive an ordered pair (pair-list) interpretation; it also means that the *wh*-element *who* will not be able to participate in a pair-list interpretation—which is tantamount to stating that the *wh*-operator *what* has a superiority relation with the *wh*–in situ element *who* (hence the ungrammaticality of (96)). The data in (97) and (98) also follow naturally under my analysis. The Superiority Effects in the (a) examples are straightforward, so I will not comment on them. In (97b) and (98b), neither of the *wh*–in situ elements can have [REF/] features. If they did, the matrix CP would have structure (100), a structure in which the *wh*–in situ elements *who* and *what* c-command the *wh*-operator *whom* and, therefore, cannot have their reference-dependent features [REF/] checked against a c-commanding *wh*-operator.

(100) [who [what [whom . . .]]]

There is simply no way for the [REF/] features of *who* and *what* to be checked in (100). Consequently, there is no derivation for (97b) or (98b) that would license a pair-list interpretation with either *wh*–in situ element—that is, the *wh*-operator *whom* necessarily has superiority relations with the *wh*-in situ elements.

The data discussed by Richards (1997), and rediscussed by Aoun and Li (2003), offer another challenge to my analysis of Superiority Effects. According to Richards and to Aoun and Li, the sentence in (101) has no detectable Superiority Effect.

(101) What did who persuade whom to buy

For me, (101) has the same grammaticality that (73b) has. It seems to me that both (73b) and (101) have interpretations where the *wh*–in situ elements receive a discourse (or echoic) valuation; however, neither (73b) nor (101) allows a pair-list interpretation involving a *wh*-operator and a *wh*–in situ element. That is, I sense that (101) does in fact have a Superiority Effect in it. Support for my "sense" comes from the *wh*-the-hell example in (102).[11]

(102) *What the hell did who persuade whom to buy
cp Who the hell persuaded whom to buy what

Example (102) is quite awful, suggestive of the fact that it involves a Superiority Effect. My analysis predicts the Superiority Effect in (101). The only way (101) can avoid having a Superiority Effect is if it can license a pair-list reading. To license such a reading, one of the *wh*–in situ elements would have to have a [REF/] feature that can be checked against the features of a c-commanding *wh*-operator. Unfortunately, should the *wh*–in situ elements in (101) possess a [REF/] feature, then the application of various Merge and Remerge operations will eventually derive a CP with structure (103).

(103) [who [whom [what . . .]]

In this structure, the *wh*-operator *what* cannot c-command either *wh*–in situ element inside the CP, so it cannot be involved in checking any [REF/] features for these in situ elements. Given this analysis, I (correctly) predict that (101) does not permit a pair-list interpretation and that it does exhibit a Superiority Effect, despite the claims by Richards and Aoun and Li.

Parasitic Gaps

The different types of features that *wh*-elements can carry not only affect *wh*-extraction and superiority relations, but they also play an important role in licensing parasitic gaps (PG) constructions; see (104).

(104) Which article did Pat file [after reading PG]

The hallmark property of PG constructions is that they permit a *wh*-element to be linked interpretively to a seemingly extraneous gap (the PG). Interestingly, although some *wh*-elements can license PGs, not all of them can do so (an observation first made by Engdahl (1983)). For example, *wh*–in situ elements and *wh*-echoic elements cannot license PGs, as is shown in (105).

(105) a. *Who filed which article [after reading PG]
b. *Chris filed which article [after reading PG]

Data such as those in (104) and (105) have led many linguists to assume that PGs can be licensed only by pure *wh*-operators. This appears to be the case; however, it is important to note that not all pure *wh*-operators can license PGs in English—only referential operators can. If we compare the ability of the referential *wh*-operator *where* in (106a) to license a PG with the ability of the nonreferential operator *why* in (106b), we will notice that only the former can license parasitic gaps.

(106) a. Where did Chris meet Sam [before meeting Pat PG]
b. *Why did Chris meet Sam [before meeting Pat PG]

Being a pure *wh*-operator, then, is not sufficient to license PGs; rather, these operators must have other features as well. In English, PG licensers must have a [*WH*-OP] feature and a [REF] feature.

The fact that an operator feature alone cannot license PGs not only explains the data in (106), but it also helps account for some observations that Lin (2005) makes about PGs in Chinese. Lin notes that not all *wh*-operators in Chinese can license PGs, as (107) demonstrates.

(107) *Laowang [zai huijian PG zhiqian] jiu kaichu-le shei
Laowang at meet PG before already fire-PERF who
'Who did Laowang fire before meeting?'

The *wh*-operator in (107) cannot license the PG despite being a pure referential operator. However, if the *wh*-operator has a TOP(icalization) feature (which forces the *wh*-operator to remerge in a TopP to have its topicalization feature checked), then it can license a PG (see (108)).

(108) Shei Laowang [zai huijian PG zhiqian] jiu kaichu-le
Who Laowang at meet PG before already fire-PERF
'Who did Laowang fire before meeting?'

As we can see, Chinese, like English, allows *wh*-operators to license PGs, but only if these operators have other features that come into play, too (though these additional features can vary from language to language).

The arguments made in this section are much like the arguments made in the previous two sections: they demonstrate that *wh*-constituents can possess sets of features that determine the interpretive and distributional properties of *wh*-related phenomena.

That-Trace Effects

Another *wh*-puzzle that has intrigued generative linguists concerns the *that*-trace data discussed in Chomsky and Lasnik (1977) as well as in Chomsky (1981); see (109).

(109) a. *Who does John think [$_{CP}$ t that [t left]]
b. Who does John think [$_{CP}$ t that [Bill saw t]]
c. Who does John think [$_{CP}$ t e [t left]]
d. Who does John think [$_{CP}$ t e [Bill saw t]]

What is puzzling about the data in (109) is why *wh*-objects can be extracted from embedded CPs regardless of whether the CP has a lexical *that*-complementizer or an empty *e*-complementizer, but *wh*-subjects can be extracted from embedded CPs only if the CP has an empty *e*-complementizer. The curious ungrammaticality of (109a) requires an explanation. Chomsky and Lasnik (1977) propose that (109a) is ungrammatical because its derivation will include the ill-formed structure stated in (110).

(110) *That*-trace filter. *[that t]

Needless to say, this account is not at all explanatory; all it does is label the existence of *that*-trace effects. It does not, as it should to be adequately explanatory, provide an account of why the *that*-trace filter should exist at all.

Chomsky (1986) offers another explanation for (109a), proposing that (109a) is ungrammatical because it does not satisfy the Empty Category Principle (ECP), which requires nonpronominal empty categories to be lexically governed or antecedent governed. In (109a), the *wh*-trace must satisfy the ECP. However, it cannot do so because it cannot be lexically governed by either of the two heads that govern it—the Inflectional head of the sentence or the complementizer C—since these functional categories are not lexical categories and, therefore, cannot lexically govern the trace. Nor can it be antecedent governed by the coindexed intermediate trace in the Spec of CP since the C head intervenes within the binding relationship between the two traces and blocks antecedent government. For Chomsky, given that the *wh*-subject trace cannot satisfy the ECP, sentence (109a) must be ill formed. On the other hand, the sentences in (109b–c) are all grammatical, in part, because their *wh*-variable traces satisfy the ECP. The subject trace in (109b), though not lexically governed, can be antecedent governed by the intermediate trace, since the empty complementizer cannot intervene in any binding relationship, and the object traces in (109c–d) are both lexically governed by their verbs. Hence, all the *wh*-traces in (109b–d) are properly governed.

What is important about Chomsky's ECP analysis of *that*-trace effects is that he defines his solution to the *that*-trace puzzle in terms of the (lexical) properties of the complementizers *that* and *e*. This property-based analysis of *that*-trace effects is later refined by Rizzi (1990), who brings Agreement features into the mix. Noting that Chomsky's analysis prohibits an intermediate trace in a CP headed by *that* from antecedent governing a *wh*-trace, Rizzi observes that Chomsky mispredicts the grammaticality of (111) since the *wh*-adjunct trace will be neither lexically governed nor antecedent governed under Chomsky's assumptions.

(111) How do you think [$_{CP}$ t that [Bill solved the problem t]]

With the grammaticality of (111) in mind, Rizzi argues that if (111) is to satisfy the ECP, then the *wh*-adjunct trace must be antecedent governed by the intermediate trace in CP. But this means that the *wh*-subject trace in (109a) must also be antecedent governed, despite Chomsky's claims to the contrary. If so, Chomsky's ECP analysis of the ungrammaticality of (109a) fails. Committed to the theoretical necessity of the ECP, Rizzi proposes a way to salvage an ECP analysis of (109). Rizzi maintains that the ECP has two conditions that must be met: (i) a nonpronominal empty category must be properly head governed (canonically governed by a head with lexical features) and (ii) it must be antecedent governed or theta governed. The ungrammaticality of (109a) arises, for Rizzi, because the *wh*-subject trace cannot be properly head governed by the Infl element (which governs the trace, but not in the canonical direction) or by the C head *that* (which lacks the features necessary for proper government). Although Rizzi has a ready analysis for the ungrammaticality of (109a), his analysis of (109b) requires a few additional theoretical refinements. Rizzi contends that an explanation for (109b) is grounded in the features available for complementizers in English. The complementizer features available, according to Rizzi, are not the [+overt]/[−overt] features identified by Chomsky. Rather, complementizers have a [+Agreement]/[−Agreement] feature visible in other languages such as French, which has a complementizer with an agreement feature *qui* and one without agreement *que*. Importantly, complementizers with agreement features can properly head govern the subject position, whereas those without agreement will not be able to do so; this predicts that subject extraction will be licit (i.e., will satisfy the ECP) if and only if this extraction is from an embedded CP with a C [+AGR]. The data in (112) support this prediction.

(112) L'homme que je crois [t qui/*que [t viendra]]
The man who I think that will come
'The man who I think that will come'

In (112), the complementizer with agreement *qui* licenses subject extraction, while the complementizer without agreement *que* does not, just as Rizzi predicts. According to Rizzi, the agreement distinction between *qui* and *que* in French also appears in English. The complementizer *that* for Rizzi lacks an [Agreement] feature, but the empty complementizer (identified by Rizzi as *Agr*) has [Agreement]. Under these assumptions, (109b) is grammatical because the empty complementizer head *Agr* will properly head govern the subject trace (satisfying the first condition of the ECP), and the intermediate trace will antecedent govern the subject trace (satisfying the second condition of the ECP). By supplementing his version of the ECP with assumptions about complementizer features, Rizzi can successfully account for the difference between (109a) and (109b).

Over the last decade, minimalist analysis has veered away from stipulative mechanisms such as the ECP. Because of this theoretical shift, Rizzi's analysis of (109a) and (109b) has in large measure been abandoned—at least the ECP aspects of it have been. Some of the spirit of Rizzi's analysis, however, still remains: in particular, his assumption that the features of the complementizer play a crucial role in explaining the grammatical difference between (109a) and (109b). Sasaki (2000), for instance, argues that the Case features of complementizers account for the French data in examples such as (112). The complementizer *que* has a [−Case] feature and the complementizer *qui* has a [+Case] feature. In other words, *que* will not be able to tolerate an XP [+Case] in its Spec position, whereas *qui* will require such an XP. In (112), the [Case] feature of the *wh*-subject in the most embedded clause, though checked in the TP of the CP, will remain active in its interpretable phase (according to Chomsky 2001, interpretable phases include v*P and CP). Since the *wh*-subject has a [Case] feature that remains active in the most embedded CP, it will be able to escape this CP only if it lands in a SpecCP position that accepts its [Case feature]. That is, the *wh*-subject will escape the CP if it is headed by *qui* (which has a [Case] feature) but not if it is headed by *que* (which lacks a [Case] feature). The data in (112) conform with this analysis. Sasaki proposes a similar analysis for (109a) and (109b), but with one twist in it. The one twist is that, for Sasaki, the complementizer *that* in English has a [−Case] feature and the empty complementizer is unmarked for a [Case] feature. Given the above assumptions, we can account for the data in (109a) and (109b) as follows. In (109a), when the *wh*-subject moves into the SpecCP position, its [+Case] feature will remain active throughout its CP phase, in accordance with Chomsky's (2000a) assumptions about the phase domain of features. This movement, though, creates an ill-formed structure in the embedded CP because the complementizer *that* cannot take a [+Case] constituent in its Spec. On the other hand, the movement of the *wh*-subject in (109b) will not produce an ungrammatical structure, because the [+Case] feature of *wh*-element has no effect on the empty complementizer, which is unmarked for a [Case] feature. (Note: Sasaki's analysis also has a straightforward explanation for (109c) and (109d): the [Case] feature of the *wh*-object is no longer active once the *wh*-object moves out of its v*P phase; hence, the *wh*-object will move into the embedded SpecCP without a [Case] feature and will not have any effect on either the complementizer *that* or the empty complementizer.)

Although Sasaki's analysis of the data in (109) seems a viable successor to Rizzi's analysis, there are two reasons not to accept it: one of them is empirical, the other conceptual. On the empirical side, Sasaki's analysis simply cannot be extended to cover the adverb effect—discussed in Culicover 1993, Browning 1996, Sobin 2002, and Haegeman 2003—in which fronted adjuncts ameliorate the *that*-trace effect, as is illustrated in (113).

(113) a. This is the man who I think [t that *(next year) [t will buy your house]]
b. Robin met the man OP Leslie said [t that *(for all intents and purposes) [t was the mayor of the city]]
c. Who do you think [t that *(over the last couple of years) [t has improved his golf game most]]

Aware that examples such as those in (113) cannot be explained by his analysis (because the *wh*-subject will move into the bracketed CP with an active [+Case] feature, which should force the complementizer *that* [−Case] to reject this movement), Sasaki tries to dismiss this evidence in a footnote. But this rather ineffective ploy cannot change the fact that native speakers of English widely accept the sentences in (113). It is not the data in (113) that are in question, it is Sasaki's theory. In addition to the empirical problem besetting Sasaki's analysis, there is also a significant conceptual problem. To account for the data in (109) and (112), Sasaki assumes that the complementizers in English and French are sensitive to the [Case] features of XPs in specifiers. However, Sasaki fails to offer any arguments for why complementizers should be sensitive to [Case] features or for why French has complementizers with a [+Case]/[−Case] sensitivity, while English has complementizers with a [−Case]/[unmarked for Case] sensitivity (the strangeness of the [Case] sensitivity in English begs for an extraordinary explanation). By not providing any justification for these [Case] features, Sasaki's analysis appears excessively stipulative.

Still, despite some problems, Rizzi's and Sasaki's analyses, which attribute *that*-trace effects to the agreement features of complementizers, have sufficient explanatory power to make it worthwhile to explore this line of analysis further. That complementizers can have features is uncontroversial. As discussed in Hoekstra and Maracz 1989, Haegemen 1992, Zwart 1997, 2001, Ackema and Neeleman 2000 and Carstens 2003, among others, complementizers can be marked overtly for agreement, as we can see in (114), a West Flemish example taken from Haegeman 1992.

(114) a. Kpeinzen dan-k (ik) morgen goan
I-think that-I (I) tomorrow go
'I think that I'll go tomorrow.'
b. Kpeinzen da-j (gie) morgen goat
I-think that-you (you) tomorrow go
'I think that you'll go tomorrow.'

In (114a) and (114b), the complementizers are inflected for the same phi-features that the subjects carry. (It is worth noting that West Flemish has two types of complementizers, one marked for agreement *da*-forms and one not marked for agreement *die*. See Rizzi 1990 for a discussion of *da/die*.) Although Hoekstra and Maracz (1989), as well as Zwart (1997, 2001) and Watanabe (2000), argue that complementizer agreement (CA) results from head agreement between a complementizer C and another head (T, I, or AGRS), Carstens (2003) convincingly argues that CA involves C agreement with the subject. Let's accept Carstens's analysis of CA, but let's also expand her analysis by assuming that languages not exhibiting overt morphological agreement on complementizers can still have CA—this latter assumption accords with Rizzi's (1990) assumption that complementizers can have agreement. Adding these assumptions to Rizzi's (1990) and Anderson and Lightfoot's (2002) assumption that the complementizer alternations that show up in languages—such as *da/die* in West Flemish, *que/qui* in French, and *that/e* in English—are defined in terms of agreement and nonagreement, will lead us to conclude that one of the paired complementizers just listed has subject agreement features, and one does not. However, let's deviate from Rizzi and from Anderson and Lightfoot, who assume that *qui* in French and *e* in English are marked with agreement features and that *que* and *that* are not, by assuming that the reverse is true. There are two reasons for reversing Rizzi's assumptions about which complementizers take agreement. First, as Bošković (1999) and Lasnik (1999) observe, some dialects of French permit an overt interrogative complementizer; see (115).

(115) Qui que tu as vu
Whom C you have seen
'Whom did you see?'

In (115), the complementizer *que* will have to have, at the very least, an [OP] agreement feature to check the [OP] feature of the *wh*-operator. This demonstrates that *que* is able to carry agreement features. Second, the complementizer *qui* in French and the empty complementizer *e* in English have more limited distributions than do *que* and *that*. This is shown in (116), where a canonical embedded clause takes the complementizer *que* but not *qui* (example (116a) is taken from Pesetsky 1982). A similar situation arises in West Flemish, where a canonical embedded clause must have the complementizer *da*, not *die* (see (116b)).

(116) a. Je crois que/*qui Pierre a faim
I believe that Pierre has hunger
'I believe that Pierre is hungry.'
b. Kpeinzen da/*die Valere gisteren dienen boek gelezen eet
I-think that Valere yesterday that book read has
'I think that Valere has read that book yesterday.'

If the complementizers with the most limited distribution are marked for agreement, then it must be the case that CA is unusual (i.e., more marked) than non-CA. But the data in (114) illustrate that canonical embedded clauses in West Flemish take complementizer agreement, suggesting that CA is the unmarked case. Relatedly, Ackema and Neeleman (2000) demonstrate that, in Hellendoorn, a marked (noncanonical) embedded clause in which an adjunct is fronted requires non-CA rather than CA.

(117) dat/*datte op den warmsten dag van 't joar wiej tegen
that/that-PL on the warmest day of the year we against
oonze wil erwarkt hebt
our will worked have
'That on the warmest day of the year we have worked against our will.'

The fact that noncanonical embedded clause in (117) must have a complementizer without agreement would seem to indicate that nonagreeing complementizers are the marked form of complementizer. If this is correct, then the unmarked complementizers *da/que/that* take agreement features and the unmarked ones *die/qui/e* do not have agreement features.

Let's add one other wrinkle to our analysis of complementizers. In addition to following Carstens's assumption that complementizers with agreement share the phi-features of the subject, let's assume that these complementizers share other subject features as well. Motivation for this assumption comes from Ackema and Neeleman's (2000) observation that in Hellendoorn subject-C agreement is different from subject-T agreement, which involves phi-feature agreement. (We can see this difference in (118)—where the verb in the C position (118a) takes different agreement morphology than does the verb in the T position (118b).)

(118) a. Volgens miej lop**e** wiej noar 't park
according-to me walk-PL we to the park
'According to me we are walking to the park.'
b. Wiej loop**t** noar 't park
we walk-PL to the park
'We are walking to the park.'

That the subject-complementizer agreement in (118a) differs from the subject-tense phi-agreement in (118b) suggests the former agreement cannot be limited to phi-feature agreement. Other types of agreement features must be involved in subject-complementizer agreement. So let's assume that this agreement is not a partial agreement of subject features, as is subject-tense agreement; rather it is, as Carstens describes it, subject agreement [+SUBJ AGR]. That is, in subject-complementizer agreement, the complementizer features will agree with the subject features. Importantly, this agreement will include phi-features as well as other features such as [WH] (or [OP]) features, should a subject have the latter features.[12]

With our revised analysis of complementizer agreement, we are now in position to revisit the data in (109).

(109) a. *Who does John think [$_{CP}$ t that [t left]]
b. Who does John think [$_{CP}$ t that [Bill saw t]]
c. Who does John think [$_{CP}$ t e [t left]]
d. Who does John think [$_{CP}$ t e [Bill saw t]]

Notice that (109a–b) and (109c–d) differ in that the former sentences have the complementizer *that* and the latter, the empty complementizer *e*. This difference plays a crucial role in explaining the grammaticality judgments expressed in (109). Let's turn first to sentence (109a). In (109a), the complementizer *that* has a [+SUBJ AGR] feature and, therefore, it will have the same features that the subject *who* does. The shared features will include the [WH] feature. We know from previous discussions in this chapter that the [WH] feature of the subject *who* cannot be checked by the embedded T head. Consequently, the [WH] feature will SURVIVE and the subject will have to remerge in the embedded CP. Since the C head *that* has all the features that the subject *who* has, none of the features of *who* will be incompatible with the head *that*—that is, none of them will SURVIVE the head. Not having any surviving features, the subject *who* cannot remerge beyond the embedded CP. The subject, then, is stalled in the embedded CP. Unfortunately, the head *that* cannot check operator features without violating the subcategorization requirements of the verb *think*, which cannot take a CP [+OP]. As a result, the subject *who* is stranded in a CP where its *wh*-operator feature cannot be checked. The derivation for (109a) must be abandoned at this point and a *that*-trace effect arises. In (109b), the *wh*-object *who* will also eventually have to be remerged in the embedded CP, because its [WH] feature continues to SURVIVE each head H it (re)merges with prior to the C head. When it reaches the embedded CP, it will find that its [WH] feature is not compatible with the [+SUBJ AGR] features of the complementizer *that*. Because its [WH] feature still survives, the *wh*-object will remerge, and continue to do so, until it reaches the matrix CP. There its [WH] will be appropriately checked and no *that*-trace effect will emerge. Sentences (109c) and (109d) also have unproblematic derivations. The

[WH] features of the *wh*-operators in (109c) and (109d) will bring each of these operators to an embedded CP. Since the CP will be headed by a complementizer *e* that has no agreement features, the [WH] feature of the *wh*-operator will SURVIVE and the operator will have to remerge iteratively until it has its [WH] feature checked in the matrix CP. Complementizer agreement, then, affects *wh*-displacement (*wh*-SURVIVAL) in the following ways: (i) subject-agreement complementizers block subject remerger, but not object remerger, and (ii) complementizers without agreement are porous, not blocking any remerger. And the reason that complementizer agreement has the effects listed above has everything to do with SURVIVE.

Although we can comfortably account for the facts in (109), explaining the equivalent French facts (see (119)) poses some challenges.

(119) a. *l'homme que je crois [OP que [OP est intelligent]]
The man who I believe that is intelligent
'The man who I believe that is intelligent.'
b. l'homme que je crois [OP que [Jean connait OP]]
the man who I believe that Jean knows
'The man that I believe that Jean knows.'
c. l'homme que je crois [OP qui [OP est intelligent]]
the man that I believe that is intelligent
'The man that I believe that is intelligent.'
d. *l'homme que je crois [OP qui [Jean connait OP]]
the man who I believe that Jean knows
'The man that I believe that Jean knows.'

That a *wh*-subject cannot escape a CP headed by a complemetizer with agreement *que*, as in (119a), but can escape a complementizer without agreement *qui*, as in (119c), follows in straightforward fashion under our analysis, as does the fact that a *wh*-object can escape a CP headed by a complementizer with agreement, as in (119b). What is problematic for our analysis, however, is the fact that a *wh*-operator cannot escape the [−SUBJ AGR] complementizer *qui* in (119d). We predict that (119d) should be well formed, when it is not. To account for (119d), we need to look more carefully at the differences between English complementizers and French complementizers. In English, as we can see in (120a), complementizer agreement is generally optional, though a few verbs such as the bridge verb *quip* in (120b) can require it.

(120) a. Chris believes that/e Sam is intelligent
b. Chris quipped that/*e Sam is intelligent

Complementizer agreement in French, on the other hand, is not optional, as (121) shows.

(121) Je crois que/*qui Pierre a faim
I believe that Pierre has hunger
'I believe that Pierre is hungry.'

French complementizers must have agreement if they can—that is, unless the agreement will cause a derivation to stall. Under this assumption, (119b) is ungrammatical because there is no reason to have the [−SUBJ AGR] complementizer *qui* since *que* does not stall the derivation. If this analysis is on the right track, then we would expect languages that do not have optional agreement to behave more like French in (119), than like English in (109). As we recall from (116b), West Flemish is like French in that it requires complementizer agreement.

(116) b. Kpeinzen da/*die Valere gistern dienen boek gelezen eet
I-think that Valere yesterday that book read has
'I think that Valere has read that book yesterday.'

Given (116b), we predict that West Flemish should have complementizer patterns similar to French. According to Sasaki's (2000) data, the West Flemish example in (122) behaves like the French example in (119), allowing the [−SUBJ AGR] complementizer *die* only where it is necessary to prevent a stalled derivation.

(122) a. *den vent da Pol peinst [OP da [OP gekommen ist]]
the man that Pol thinks that come is
'The man that Pol thinks that has come.'
b. den vent da Pol peinst [OP da [Marie OP getrokken heet]]
the man that Pol thinks that Marie photographed has
'The man that Pol thinks that Marie has photographed.'
c. den vent da Pol peinst [OP die [OP gekommen ist]]
the man that Pol thinks that come is
'The man that Pol thinks that has come.'
d. *den vent da Pol peinst [OP die [Marie OP getrokken heet]]
the man that Pol thinks that Marie photographed has
'The man that Pol thinks that Marie has photographed.'

We can see in (122d) that the complementizer *die* is not permitted in a derivation that could proceed apace if it had the [+SUBJ AGR] complementizer *da* instead, as in (122b). The West Flemish data, then, would appear to corroborate our analysis of the French data.

One other fact about *that*-trace phenomenon that we need to discuss concerns the adjunct intervention effect exhibited in (113′).

(113′) a. This is the man who I think [who that *(next year) [who will buy your house]]
'This is the man who I think that next year will buy your house.'

(Notice that in this version of (113a) I have replaced *t* with *who* to conform with the way my SURVIVE analysis computes derivations.) In (113a), the presence of an adjunct between the complementizer *that* and the *wh*-subject position obviates the *that*-trace effect. Here, again, French does not behave like English: it does not have any intervention effects similar to the one in (113a), as we can observe in (123).

(123) *l'homme que je crois [OP que [tous les ans OP viendra]]
the man who I believe that every year will come
'The man who I believe that every year will come.'

The sentence in (123) is ungrammatical whether or not there is any adjunct intervening between the complementizer *que* and the subject position. The absence of interventions effects in (123), however, can be explained by the absence of permissible intervention structures in French. As (124) demonstrates, French does not allow adjunction intervention between complementizers and subjects.

(124) *Je crois que tous les jours Jean mange du chocolat.
I believe that every day Jean eats chocolate
'I believe that every day Jean eats chocolate.'

We now can explain (123): French does not have intervention effects akin to (113a) simply because French does not permit intervention (this fact deserves a principled explanation, which I leave for future investigation). Since English does allow adjuncts to intervene between complementizers and subjects, we must offer an explanation for the intervention effect in (113a). A plausible explanation comes from Carstens (2003), who proposes that a complementizer C can agree with α only if C closest c-commands α and α is nominative. Applying Carstens's proposal to (5a), we will see that the mostly deeply embedded complementizer *that* closest c-commands the adjunct, which happens not to be nominative; therefore, the C *that* cannot agree with any element α, including the subject. Hence, even though the complementizer *that* is a potential [+SUBJ AGR] complementizer, it is not able to have subject agreement in sentence (113a)'s structural configuration and, therefore, it will not have the features able to deactivate the *wh*-subject [WH] feature. Having a surviving [WH] feature, the *wh*-subject will escape the CP in the same way that the *wh*-subject does in (109c).

The foregoing analysis of (113) crucially hinges on Carstens's notion of complementizer agreement, a notion that, as stated, is stipulative. In hopes of strengthening my analysis, let me offer a few remarks about Carstens's notion of agreement. We can schematize Carstens's conditions on agreement as in (125).

(125) ... C $[_{XP}$ YP [+NOM]...

The closest c-command condition on complementizer agreement identifies YP—which is the specifier of XP most adjacent to C—as the locus of complementizer agreement. And the nominative Case condition requires that the YP specifier have a particular morphological Case that can be checked only in a Spec position of TP. This suggests that XP in (125) is TP and that Carstens's conditions are conditions on the outermost specifier of the complementizer's subcategorized TP argument. Now XPs can include two sets of features: head features and specifier-argument features. An example of an XP with both head features and specifier-argument features is the DP *whose mother*, a DP with a feature matrix that has both D features and a [WH] specifier feature. Though heads generally select only the head features of their XP-complement arguments (as the verb *wonder* selects a CP complement with an [OP] head), it is in principle possible for a head to select all the features of its XP complement argument—that is, both the head features and the specifier-argument features. Putting this relationship in the bottom-up computation of a minimalist syntax, we can formulate the above claim as: once an XP is computed, it can merge with a head Z that agrees with all XP's features, including its head and its specifier-argument features. Carstens's conditions on complementizer agreement can be met under the Merge operation just describe if a C head merges with a TP and agrees with the head and specifier features of TP. However, there is still one issue we must address. Carstens insists that the outermost specifier YP in (125) have a nominative Case feature. This ensures that YP is a specifier argument of T, an argument that has features (phi-features and the nominative Case feature) checked by T. But what happens if the YP specifier does not take a nominative Case feature (and does not have its phi-features checked by T), as in (113a) where the outermost specifier YP is an adjunct, not an argument? In the course of the derivation of a TP, the T head merges (perhaps remerges) with various XPs projecting only T features until it merges or remerges with the final specifier and then projects a TP. It is only in the creation of TP that head features and specifier features can amalgamate. If the final specifier YP is not a specifier argument of T (i.e., it does not share agreement features with T), its features will not be able to amalgamate with the head features—thereby forming a feature matrix for TP that includes head and specifier features. The reason is that the non-T-agreeing features of YP are not intersective with the constitutive/defining features of TP; in other words, the features of YP are extraneous to TP and cannot be part of the feature matrix that defines TP. So if the final specifier YP in TP is not a feature-interrelated argument of T, TP will simply not have specifier-argument features for the C head to agree with. This is what happens in (113a).

(113a) This is the man who I think [who that [*(next year) [who will buy your house]]]

In (113a), if an adjunct intervenes between the complementizer *that* and the *who*-subject, the most embedded TP will not project the subject features that trigger complementizer agreement. Without such agreement, as we have seen several times before, the complementizer will not deactivate the [WH] feature of the *wh*-subject; consequently, the *wh*-subject will continue to remerge until its [WH] feature is checked. That is, adjunction intervention prevents complementizer agreement and circumvents potential *that*-trace effects.

Let me make one final observation about *that*-trace effects. Since these effects, as seen in (109a), result from complementizer-subject agreement and since this agreement, like agreement generally, is a local phenomenon, we would predict that there should be no long-distance *that*-trace effects. In other words, all [+SUBJ AGR] complementizers not immediately c-commanding a *wh*-subject should not participate in *that*-trace effects with the displaced *wh*-subject. The data in (126) and (127) corroborate this prediction.

(126) l'homme que je pense [OP que Jean croit [OP qui [OP viendra]]
The man who I think that Jean believes that will come
'The man that I think that Jean believes that will come.'

(127) Who does Chris believe [OP that Sam told you [OP e [OP would leave]]]

In both (126) and (127), the *wh*-subjects not only can escape the CPs headed by [−SUBJ AGR] complementizers *qui* and *e*, they can also escape the higher CPs headed by [+SUBJ AGR] complementizers *que* and *that* because the latter complementizers, which agree with the features of their local subjects *Jean* and *Sam* respectively, do not share the features of the *wh*-subjects and have no way to deactivate the [WH] features of these subjects. The *wh*-subjects, then, will continue to remerge in the derivation until they reach CPs with heads carrying a [WH] feature.

In this section, I have discussed how a SURVIVE analysis can explain *that*-trace effects. Building on Carstens's analysis of complementizer agreement, I have demonstrated that *that*-trace effects arise when a *wh*-subject remerges in its local CP and this CP has a complementizer with subject-agreement features. In such a case, none of the features of the *wh*-subject can SURVIVE the CP since the complementizer is not incompatible with these features and, therefore, will deactivate (though not check) them. As a result, the *wh*-subject will be stranded in the CP with an unchecked [WH] feature, and the derivation will be forced to abort.

Wh-Elements in English Infinitives

The previous section examined *wh*-phenomena in tensed embedded clauses; this section will look at *wh*-phenomena in nontensed (infinitival) embedded clauses. Infinitival clauses in English are generally CPs that can show up in two ways: either with an empty complementizer *e*, together with the empty pronoun PRO, as in (128), or with a lexical complementizer *for*, as in (129).

(128) a. Chris wants [e [PRO to hired Pat]]
b. Chris went to the store [e [PRO to buy some milk]]

(129) a. I would prefer very much [for [you to hire Pat]]
b. [For [Chris to win the election]] would be a big surprise

Infinitival clauses, however, can also be TPs, but only in exceptional cases where a lexical head especially selects a TP argument, as the verbs *want* and *expect* do in (130).

(130) a. Chris wants [$_{TP}$ Sam to hire Pat]
b. Chris expects [$_{TP}$ Sam to read the Koran]

An indicator that a head has selected an infinitival TP argument is that the TP will have a lexical subject.

Infinitival TPs are interesting, in part, because they cannot strand *wh*-elements. Since the TP head *to* does not possess a [WH] feature (nor do the heads VP or vP), any *wh*-element in an infinitival TP will have its [WH] feature SURVIVE the TP and the *wh*-element will, therefore, necessarily remerge beyond the TP. In other words, any *wh*-elements in an infinitival TP can escape the TP. We can observe this in (131).

(131) a. Who does Chris want [who to hire Pat]
 'Who does Chris want to hire Pat?'
 b. Who does Chris want [Sam to hire who]
 'Who does Chris want Sam to hire?'
 c. How does Chris want [Sam to fix the car how]
 'How does Chris want Sam to fix the car?'

In (131a), a *wh*-subject escapes the TP; in (131b), a *wh*-object escapes; and in (131c), a *wh*-adjunct escapes. Importantly, we would expect not only *wh*-operators to escape TP infinitives, but *wh*–in situ elements to do so, too. The fact that the T head of an infinitival TP clauses lacks a referential tense feature would lead us to predict that if an infinitival TP contains a *wh*–in situ element with a [REF/] feature, this *wh*-element will be able to escape the nonreferential TP and eventually remerge in a CP where it can have its [REF/] *wh*-feature checked. Such *wh*-elements, then, should be able to participate in pair-list interpretations. The examples in (132) support our prediction.[13]

(132) a. Who expects [Sam to read what]
 a′. Who the hell expects Sam to read what
 b. Who wants [whom to go where]
 b′. Who the hell wants whom to go where
 c. What does Chris expect Sam to read to whom
 c′. What the hell does Chris expect Sam to read to whom

As we can see in (132a–c), the *wh*-in situ elements do participate in pair-list interpretations—this is emphasized by the well-formedness of (132a′–c′). TP infinitives, then, are porous for both *wh*-operators and *wh*–in situ elements. This porousness is expected under our analysis because T heads lack the [WH] features and reference features necessary to deactivate the [WH] features of *wh*-elements. As a result, *wh*-elements with concatenative [WH] features cannot be stranded in infinitival TPs.

CP infinitives headed by empty complementizers (see (128)) behave like TP infinitives with respect to *wh*-phenomena. To see why, consider the structure of these CPs, given in (133).

(133) ... [$_{CP}$ e [$_{TP}$ PRO to vP]]

In (133), the infinitival TP cannot strand elements with [WH] features, for reasons discussed above. But then neither can the CPs. This is so because an empty complementizer in the CP, like an empty complementizer in a tensed embedded clause, does not have agreement features. Consequently, any *wh*-element that reaches the SpecCP position will not have its [WH] feature deactivated by the empty complementizer *e* and the *wh*-element will remerge outside the CP. If our analysis of (133) is correct, both *wh*-operators and *wh*–in situ elements should pass through the CP as freely as they do through the infinitival TPs in (131) and (132). The data in (134) and (135) confirm this.

(134) a. What does Chris want [what e [PRO to do what]]
'What does Chris want to do?'
b. What did Chris go to the store [what e [PRO to buy what]]
'What did Chris go to the store to buy?'

(135) a. Who wants [e [PRO to buy what]]
'Who wants to buy what?'
b. Who went to the store [e [PRO to buy what]]
'Who went to the store to buy what?'

As we can see in (134), a *wh*-operator can remerge outside the infinitival CP, regardless of whether the CP is an argument (134a) or an adjunct (134b). And in (135a–b), the *wh*–in situ element *what* can iteratively remerge, eventually having its [REF/] feature checked in the matrix CP, thereby licensing the *wh*–in situ element for a pair-list interpretation. (Of special note, the CP adjuncts in (134b) and (135b) are porous with respect to *wh*-displacement.)

CP infinitives headed by the complementizer *for* exhibit one *wh*-property not shared by the other infinitives: they will not allow a *wh*-subject to escape. That is, they have what looks like a *for*-trace effect, as (136) illustrates. (I use traces in (136), rather than copies of *who*, to make the possible *for*-trace effect register more clearly.)

(136) a. *Who would Chris prefer quite sincerely [t for [t to hire Pat]]
cp Chris would prefer quite sincerely for Sam to hire Pat
b. *Who would it be easy [t for [t to hire Pat]]
cp It would be easy for Sam to hire Pat

If the complementizer *for* is a complementizer with [+SUBJ AGR] features, then the ungrammaticality of (136a) and (136b) will have the same explanation I have offered for *that*-trace effects. Importantly, there is a reason to assume that the complementizer *for* has [+SUBJ AGR] features. As Haegeman (1991) and many others have maintained, the complementizer *for* is responsible for the Case feature of the infinitival subject in (137).

(137) For him to attack would be surprising

What this means is that the Case feature of the complementizer *for* must agree with the Case feature of the infinitival subject. Hence, *for* has [+SUBJ AGR] features. Having subject-agreement features, the complementizer will share all the subject's features, including the [WH] feature, and it will not be incompatible with any of the subject's features. When the *wh*-subject remerges in the Spec of the infinitival CP, it will not have any incompatible features to trigger further remerge—that is, it will be stranded in the embedded CP, where its [WH] feature cannot be checked. The stranded, unchecked feature in the CP will force the derivation to abort. Interestingly, even though *that*-trace effects can be ameliorated by adjunct intervention (because this intervention prevents complementizer agreement), no such amelioration is possible in infinitives, as we can observe in (138).

(138) *Who would it be easy [t for [next week [t to hire Pat]]

But why shouldn't adjunct intervention work in (138) when it works in *that*-trace constructions? The reason is that if an adjunct intervenes and blocks complementizer-subject agreement, then the complementizer will not be able to share its Case feature with the subject and the Case feature of the subject will remain unchecked. Support for this analysis comes from the fact that adjuncts cannot intervene even when the subject is not a *wh*-subject, as in (139).

(139) *It would be easy [for [next week [Sam to hire Pat]]]

By intervening between the complementizer and the subject, the adjunct makes it impossible for the subject to have its Case feature checked.

Wh-Constructions in Slavic

Wh-constructions in Slavic languages provide an interesting challenge to our analysis of *wh*-constructions because they differ from *wh*-

constructions in English in several significant ways. First, in Slavic, as opposed to English, *wh*-phrases cannot be left in situ; as Bošković (2002) observes, not even echo *wh*-phrases can be left in situ (see (140)).

(140) a. ?*Ivan kupuje šta? (Serbo-Croatian)
Ivan buys what
b. ?*Ivan e kupil kakvo? (Bulgarian)
Ivan is bought what
c. ?*Ivan kupil cto? (Russian)
Ivan bought what

Second, unlike in English, all *wh*-words in multiple-*wh* constructions in Slavic appear in fronted positions, according to Rudin (1988), Bošković (1999, 2002), and Richards (2001), among others. Examples of this fronting are given in (141).

(141) a. Ko koga voli (Serbo-Croatian)
who whom loves
'who loves whom'
b. Koj kogo e vidjal (Bulgarian)
who whom AUX loves
'who loves whom'
c. Kto kogo ljubit (Russian)
who whom loves
'who loves whom'

Third, multiple-*wh* constructions can involve nonreferential *wh*-elements in Slavic languages, though they do not in English. Bošković (1997a, 1997b, 1999) provides relevant examples, such as those in (142).

(142) a. Kogo kak e tselunal Ivan (Bulgarian)
whom how is kissed Ivan
'How did Ivan kiss whom'
b. koj kogo kak e tselunal (Bulgarian)
who whom how is kissed
'Who kissed whom how'

The data in (140)–(142) make it clear that the [WH] features in English—[DISC], [REF/], and [OP]—need not be the [WH] features selected for Slavic languages. Take, for instance, the [DISC] *wh*-feature in English. As I discussed previously in this chapter, the [DISC] feature is not a concatenation feature; hence it is not checked syntactically and it plays no role in where its host *wh*-element is spelled out morphophonetically. Slavic languages do not have such a feature. All [WH] features in

Slavic languages do play a role in morphophonetic Spell-Out: as the evidence in (141) suggests, they compel *wh*-elements to front. As with the [DISC] feature, the [REF/] feature in English lacks phonetic realization. Although this feature is checked syntactically and can force its host element to remerge, as in (143), it is not a morphophonetic feature and plays no role in where its host has phonetic visibility in a sentence.

(143) [who [whom . . . likes whom]]
'Who likes whom'

In (143), the [REF/] feature is checked in the matrix CP, but the morphophonetic Case and Agreement features of the ⟨whom, whom⟩ chain are spelled out on the lower copy. Again, Slavic languages do not have a [REF/] *wh*-feature. This is obvious in that all *wh*-elements in Slavic do have morphophonetic effects and in that the nonreferential *wh*-elements in (142) participate in multiple-*wh* constructions, which indicates that referential dependency is not a determinative feature of Slavic multiple-*wh* constructions.

Slavic languages may not have [DISC] and [REF/] *wh*-features; however, they do share one *wh*-feature with English: the [OP] feature. In English, the [OP] feature is the one *wh*-feature that has morphophonetic visibility. It is a concatenation feature that must be checked by a head capable of giving phonetic visibility to the feature. The *wh*-operator *who* in (143) iteratively remerges in the derivation until it gets to the CP, where the head C can check its [OP] feature, and when derivation (143) proceeds to the interfaces, the [OP] feature will be interpreted both semantically and morphophonetically. In Slavic languages, all *wh*-elements also have some sort of [OP] feature—one that must be checked syntactically and one that is interpreted both in the sensorimotor interface and the conceptual-intentional interface. The fact that all *wh*-elements in Slavic have an [OP] is responsible, then, for the inability of *wh*-elements to remain in situ (see (140)) and for the mandatory fronting of all *wh*-elements, as illustrated in (141).

Were the telling properties of *wh*-constructions in Slavic languages limited to those exhibited in (140)–(142), our analysis would be at an end. We are not done, however, because *wh*-constructions in Slavic languages have other properties that require explanations. Here I will discuss what I consider to be one of the most interesting of these properties—the fact that Superiority Effects are not uniform across Slavic languages. According to Rudin (1988) and Bošković (1999, 2002), some Slavic languages exhibit Superiority Effects and some do not. Bulgarian is one of the Slavic languages that has Superiority Effects. It will not permit a *wh*-object to cross over a *wh*-subject, as in (144), nor a *wh*-adjunct to cross over a *wh*-object, as in (145).

(144) a. Koj kakvo e napisal (Bulgarian)
who what AUX wrote
'Who wrote what'
b. *Kakvo koj e napisal
what who AUX wrote
'What did who write'

(145) a. Kogo kak e tselunal Ivan
whom how AUX kiss Ivan
'Whom did Ivan kiss how'
b. *Kak kogo e tselunal Ivan
how whom AUX kiss Ivan
'How did Ivan kiss whom'

Bošković (2002) points out that in Bulgarian equivalent Superiority Effects show up in all contexts, save those like (146), where a *wh*-adjunct can cross over a *wh*-object in constructions that include a third *wh*-element.

(146) a. Koj kogo kak e tselunal
who whom how AUX kissed
'Who kissed whom how'
b. Koj kak kogo e tselunal
who how whom AUX kissed
'Who kissed whom how'

Notice that even though *kak* cannot cross over *kogo* in (145b), it can in (146b). Now, we can account for all the above data if we build on the analyses of Rudin (1988), Richards (2001), and Bošković (1999, 2002). Let's follow Bošković in assuming that (i) in Slavic languages, only one *wh*-element can have a *wh*-operator [OP] feature checked in CP, and (ii) all other *wh*-elements in a multiple-*wh* construction have a focus-operator feature [FOC] that is checked in a Focus Phrase FP and that is interpreted in the sensorimotor interface (i.e., the [FOC] feature has morphophonetic visibility). Let's also follow Rudin and Richards in assuming that *wh*-features in Bulgarian are checked outside the TP. And let's follow Rudin in allowing for the possibility that a *wh*-focus feature in Slavic languages can be checked in more than one site—that is, there can be multiple FP sites, which we can see in Serbo-Croatian examples (147) taken from Rudin. Notice that in (147a) and (147b) the *wh*-element *koga* can appear in two different structural positions.

(147) a. Ko je koga prvi udario
who has whom first hit
'Who hit whom first'
b. Ko je prvi koga udario
who has first whom hit
'Who hit whom first'

And finally, let's take Bošković's observation that multiple-*wh* constructions in Bulgarian must take pair-list interpretations, not single-pair interpretations, to mean that *wh*-elements in Bulgarian with the [FOC] feature also have an operator-dependence [OP/] feature that must be checked in the CP in the same way that the [REF/] feature in English is checked and with similar results: the formation of ordered-pair interpretations at the conceptual-intentional interface.

Armed with the above assumptions, we can explain the Bulgarian data in (144)–(146) in the following way. If, as we have assumed, multiple-*wh* constructions in Bulgarian have only one *wh*-element with an [OP] feature (because the C [OP] head can check only one [OP] feature) and all other *wh*-elements have [FOC, OP/] features, then the *wh*-element with the [OP] feature will have to be superior to (i.e., merged after) the other [OP/] *wh*-elements to ensure that when all the *wh*-elements remerge in the matrix CP, the [OP] element will c-command any operator dependent [OP/] element, as in (148). (From (148), we can see that all multiple-*wh* constructions in Bulgarian must have a *wh*-OP element to license all the [FOC, OP/] elements.)

(148) [*wh*-OP [*wh*-OP/ . . .]]

This is exactly what happens in (144a), but not in (144b). In (144a), we can tell from word-order effects that the *wh*-object *kakvo* 'what' has a [FOC] feature, which will be checked in a FP, and that the *wh*-subject *koj* 'who' has the [OP] feature, which will be checked in a CP above the FP. Importantly, as we derive (144a), the *wh*-object *kakvo*, which has an [OP/] feature in addition to its [FOC] feature, is merged before the *wh*-subject *koj* 'who'; therefore, in all subsequent remergings of these two elements—including their remerging in the matrix CP—the *wh*-subject will closest c-command the *wh*-object. In other words, the [OP] element will appropriately c-command the [OP/] element, thereby allowing the [OP/] feature to be checked. As a result, we will have a licit derivation for (144a). The derivation of (144b), on the other hand, has a very different outcome. In (144b), as the word-order effects suggest, the *wh*-object *kakvo* has the [OP] feature and the *wh*-subject *koj* has the [FOC, OP/] features. When these two *wh*-elements remerge in the matrix CP, the *wh*-object will be c-commanded by the *wh*-subject. This configuration, unfortunately, has the operator-dependent [OP/] element c-commanding the operator [OP] element. The consequence is that the [OP/] feature cannot be appropriately checked, so the derivation will have to abort and will not proceed to the interfaces for interpretation. A similar story holds for the examples in (145).

But why doesn't a Superiority Effect involving *kogo* 'who' and *kak* 'how' emerge in (146b), when one does in (145b)? The reason is straightforward. As our discussion of (144) makes obvious, Superiority Effects in Bulgarian arise from the relationship between an [OP] element and an [OP/] element. The Superiority Effect we witness in (145b) results from the fact that the *wh*-adjunct *kak* 'how' has an [OP] feature and that it is merged into the derivation prior to the *wh*-object *kogo* 'whom'—an [OP/] element. Unfortunately, in the matrix CP, the [OP/] element *kogo* will c-command the [OP] element *kak*, which will leave the [OP/] element unchecked. No such problem shows up in (146b) because neither *kak* nor *kogo* in (146b) has an [OP] feature. In fact, they both have [FOC, OP/] features. Since Superiority Effects involve the relationship between [OP] and [OP/] elements, it is simply not possible for a Superiority Effect to emerge between the two [OP/] elements *kak* and *kogo* in (146b). In fact, given that *kak* and *kogo* are [OP/] elements in both (146a) and (146b), we would expect that both of these sentences would resist Superiority Effects equally, and for the same reason. This conclusion is all the more warranted under our analysis since the Remerge operation will derive exactly the same matrix CPs for (146a) and (146b); see (149).

(149) [koj [kogo [kak . . .]]]

If we are correct, the word-order differences in (146a) and (146b) have nothing to do with Superiority Effects since they have nothing to do with the [OP/] features of *kak* and *kogo*. Rather, they emerge due to the [FOC] features of these two *wh*-elements. The [FOC] features of *kak* and *kogo* must be checked in different FPs for their degree of focus—for example, primary or secondary. The degree of focus will determine in which FP the [FOC] features for *kak* and *kogo* are checked and where they are given morphophonetic visibility.

Whereas Bulgarian exhibits Superiority Effects, Serbo-Croatian (see (150)) and Russian (see (151)) do not.

(150) a. Ko koga voli
who whom loves
'Who loves whom'
b. Koga ko voli
whom who loves
'Whom does who love'

(151) a. Kto kogo ljubit
who whom loves
'Who loves whom'
b. Kogo kto ljubit
whom who loves
'Whom does who love'

The data in (150) and (151) demonstrate that multiple-*wh* constructions in Serbo-Croatian and Russian permit *wh*-elements to order themselves freely, without inducing any Superiority Effects. To explain (150) and (151), let's assume, as we did in our discussion of Bulgarian, that the outermost *wh*-elements are *wh*-operators with [OP] features that are checked in the CP and that all other *wh*-elements have [FOC] features that must be checked in FPs. However, let's also assume that *wh*-elements with [FOC] features in Serbo-Croatian and Russian differ from their Bulgarian counterparts in that they do not also have [OP/] features. I make this latter assumption for two reasons. First, as evidence from Rudin (1988) suggests, the [FOC] feature in Serbo-Croatian and in Russian is structurally divorced from the [OP] feature in CP. The [FOC] feature is a TP internal feature checked more proximately to the verb than to the C [OP] (see (152)); on the other hand, the [FOC] feature in Bulgarian is not within the TP: it has a structurally close relationship with the CP.

(152) Kdo ho kde videl je nejasne (Serbo-Croatian, from Rudin 1988)
who him where saw is unclear
'Who saw him where is unclear.'

Second, according to Bošković, Serbo-Croatian and Russian, unlike Bulgarian, do not require multiple-*wh* constructions to receive pair-list interpretations; they can receive single-pair interpretations. If pair-list interpretations arise, as I have argued throughout this chapter, from the relationship between a *wh*-operator and a *wh*-dependent element—for example, a *wh*-element with a [REF/] or [OP/] feature—then the fact that multiple-*wh* constructions in Serbo-Croatian and Russian can be limited to single-pair interpretations suggests that these constructions do not have a *wh*-dependent [OP/] element. Should this be the case, then we can explain not only the single-pair readings permissible for the constructions in (150) and (151), but also the absence of Superiority Effects in these examples. Since Superiority Effects emerge when a *wh*-dependent element c-commands the *wh*-operator that it depends on (see my discussion of Superiority Effects in Bulgarian), languages without *wh*-dependent elements should not have any mechanism able to induce Superiority Effects. That Serbo-Croatian and Russian differ from Bulgarian in not having [OP/] elements means that Serbo-Croatian and Russian will also differ from Bulgarian in their ability to generate Superiority Effects. To give clarity to this analysis, let's look closely at Serbo-Croatian example (150b), repeated below.

(150) b. Koga ko voli
whom who loves
'Who loves whom?'

In (150b), the *wh*-object *koga* 'whom' has an [OP] feature that will SURVIVE until it is checked in the CP, and the *wh*-subject *ko* 'who' has a [FOC] feature that will be checked in a Focus Phrase within the TP. Given that *ko* lacks any features that could force it to remerge in the CP, there is simply no way for the structural conditions for a Superiority Effect to arise in the derivation of (150b).

As we can see, our discussion of *wh*-constructions in Slavic languages adds valuable support for our SURVIVE analysis of syntactic derivation. We have shown that Slavic languages, which have *wh*-properties quite unlike those exhibited in English, derive these properties in the same way that English derives its *wh*-properties—via operations devoted to feature-checking: Merge, SURVIVE, and Remerge.

Conclusion

What we have observed in this chapter is that *wh*-elements come in a variety of forms and meanings. Some *wh*-elements are in situ elements that are interpreted in a discourse; some are in situ elements that are interpreted as referent-dependent variables; some are fronted, focused dependent variables; some are fronted operators; and others not discussed surely exist. I have argued that the morphophonetic and semantic differences among *wh*-elements are encoded in the feature matrices of these elements, and that the syntactic operations Merge, SURVIVE, and Remerge are involved in checking the differing ways *wh*-features display themselves. Furthermore, my analysis of *wh*-features has provided a theoretical framework capable of explaining *wh*-phenomena such as Superiority Effects and *that*-trace effects that have long challenged, and resisted, a rash of syntactic theories.

The importance of allowing *wh*-elements to have more than one type of *wh*-feature matrix cannot be overestimated; doing so opens up productive new ways of approaching *wh*-constructions. Nearly all previous analyses of *wh*-constructions treated *wh*-elements as if they were the same thing: a *wh*-operator. We can see this in Chomsky 1981, Huang 1982, May 1985, and on and on. Unfortunately, limiting *wh*-elements to a single operator type (a [WH] operator) also limited judgments about the meaning, and the well-formedness, of multiple-*wh* constructions. The single-*wh*-element hypothesis led linguists such as Huang (1982) to confuse pair-list *wh*-constructions (see (153)) with *wh*-constructions that do not allow a pair-list interpretation (see (154)), asserting that constructions like (153) and (154) are equally grammatical.

(153) Who read what

(154) Why did Chris buy what

And, while assuming that all *wh*-elements are operators, Richards (2001) judges (155a) to be ungrammatical and, amazingly, (155b) and (155c) to be grammatical, even though (155b) is quite dreadful and (155c) is limited to the same sort of single-pair reading that (155a) is. (The judgments in (155) are Richards's.)

(155) a. *What did who buy
b. Which violin did you ask which sonata to play on
c. Who persuaded [the man who bought which car] to sell the hubcaps

In both (155a) and (155c), the *wh*-in situ elements *who* and *which car* take contrastive stress and must be given discourse interpretations, so why is (155a) much worse than (155c)? It seems to me that the peculiar judgments in (153)–(155) are theory-bound judgments that work to protect, among other things, the single-*wh*-element hypothesis. In this chapter, by moving away from the single-*wh*-element hypothesis, we have been able to refine our judgments of *wh*-constructions and to expand our ability to account for *wh*-data.

4 Conclusion

Peters and Ritchie (1973) demonstrated that the transformational rules of generative grammar were so powerful that they grossly overgenerate the output of natural language. Having the generative capacity of unrestricted rewrite rules, early transformational rules arguably could produce grammars that would be unlearnable for children. To temper the power of these rules, the generative grammarians of the 1970s and 1980s augmented grammars with output filters such as Emond's Structure Preserving Condition, Chomsky's A-over-A Principle, and many of the subtheories of the Government-Binding model. This left generative grammars with three platforms of rules: local phrase structure rules, nonlocal transformational rules, and conditions/filters on rules. Although the system of rules described above yielded some important theoretical and empirical consequences in that it allowed generative grammars to explain, among other things, the displacement relations seen in passive constructions and *wh*-constructions, the conceptual necessity of having all-too-powerful nonlocal rules whose outputs must be filtered was never justified. That a grammar could generate structures such as those in (1), but have to rule them out because they violated an output condition or a condition on the application of rules, meant that humans could, in principle, be computing and then discarding an enormous amount of structural junk.

(1) a. *[Whom do you wonder [who will hire t]]
    b. *[Who do you believe [that t left]]

Such a grammar makes little biological (cognitive) sense. Having excessively powerful rules at our disposal that will permit us to overgenerate natural language is already problematic because of all the processing effort we could be devoting to computing junk. However, to further tax our processing abilities and processing time by compelling us to submit our overgenerated language structures to some filtering devices that weed out the structures we have just spent considerable time generating is enormously inefficient and would absorb an inordinate amount of our mental energy. A grammar that taxes our processing time in this way hardly seems biologically plausible.

The minimalist framework developed over the last decade putatively parts ways with its generative predecessors, divesting itself of all theoretical commitments save those that can pass some test of conceptual necessity. And yet, most early versions of minimalism proposed in the 1990s have continued to embrace the same platform of rules that generative grammars of the 1980s did. In particular, the early minimalist framework assumed that the system of operations/rules includes local structure-building rules (Merge), nonlocal rules with the derivational power of transformational rules (Move), and conditions on rules (Economy Conditions like Chomsky's (1995) Shortest Move, Richards's (2001) Principle of Minimal Compliance, and Aoun and Li's (2003) Minimal Match Condition). The system of operations assumed in the early versions of the minimalist framework, unfortunately, runs into the same plausibility problems that previous generative grammars did. That is, running the operations of a minimalist syntax would be enormously expensive in terms of processing effort. The Move operation, even when it is constrained by Economy Conditions, is not intrinsically constrained; it is extrinsically constrained by Economy Conditions. Hence, the Move operation will still overgenerate language structures and will compel us to use significant processing effort in doing so, and Economy Conditions will have to filter the structures of Move—another costly processing demand.

Chomsky (2002, 2005) substantially reduces the processing needs of minimalism by delimiting the system of operations to two operations: local External Merge and short-distance Internal Merge. External Merge, a direct descendant of Merge, is an operation that combines a head with an element from the Numeration. Internal Merge, a modified Move operation, combines a head with an element already in the derivation; the combination properties of Internal Merge include the leave-a-copy-behind property. Internal Merge, then, recopies elements in a derivation at sites some distance from their External Merge sites. In this way, Internal Merge assumes the displacement role that Move played in earlier version of minimalism. Although Internal Merge has Movelike properties, it also has properties that keep it from having the excessive generative power that the Move operation has. The foremost among these properties is that Internal Merge has what looks like Economy Conditions already structured into its operation. In particular, Chomsky defines Internal Merge as Agree + Move—that is, as an operation that will not permit the syntactic recopying of elements in a derivation, unless this recopying satisfies an Agreement requirement of a head H. The Internal Merge operation, then, does not licentiously allow recopying (i.e., movement); rather, it permits recopying only when a Head needs to satisfy one of its features. In this case, the Head will probe the derivation looking for an available ZP checker and if it finds one reasonably close, the Head will attract ZP. Thus the computational power of Internal Merge is naturally constrained by Agree (alternatively Attract), which not only requires a probe-H to initiate Internal Merge, but also limits the probe search to the agreement domain specified by the Phase Impenetrability Condition (a condition that permits elements in phases—CP and vP—to be available for the Internal Merge operation only if they are edge/specifier elements of the phases). What this means is that the structural reach of the Internal Merge operation is limited from one phase edge to the next phase edge. As a consequence, all long-distance displacement is actually a series of short-distance phase-to-phase displacements. By replacing the unconstrained Move operation with the tightly constrained Internal Merge operation, Chomsky lessens the processing demands of minimalist derivations.

Although Chomsky's (2002, 2005) minimalist framework makes great strides in simplifying operations and derivations and in reducing the processing demands to compute these derivations, this framework remains riddled with problems. For one, Chomsky's framework is a mixed derivational and representational framework, and as such has all the redundancy problems raised by Brody (2002); see my discussion in chapter 1. As Brody notes, any framework that includes a version of Move will necessarily be a mixed framework: no matter how narrowly Chomsky defines Move (even if Move is circumscribed by Agree), the recopying feature of this operator will cause it to be both derivational and representational, and therefore, necessarily redundant. Beyond the redundancy problem lie other, equally intractable, difficulties. For example, Chomsky's framework would appear to be ill-equipped to explain why both (2a) and (2b) are grammatical sentences.

(2) a. Who knows who said what
    b. Who knows what who said

Since Chomsky's Internal Merge operation is an Attract Closest operation that selects *wh*-subject movement over *wh*-object movement in (3), it is not set up to select both types of movement/recopying in (2).

(3) a. Who said what
    b. *What did who say

Furthermore, to license long-distance displacements like the one in (4), without allowing Internal Move to have the unlimited power—and the problems—of Move, Chomsky breaks long-distance displacement down to a series of short movements/recopyings, each of which must involve Agree and each of which is limited to phase-to-phase structural distances.

(4) [Who does John [believe [that Mary [told you [that she would [fire (who) tomorrow]]]]]]

Under the above assumptions, the *wh*-element *who* in (4) will have to move (be recopied) from bracketed phase to bracketed phase. There must be six such copies (movements). Given that each of these copying operations (movements) must be motivated by Agree (which is required if Internal Merge is to constrain its computational power), Chomsky postulates that the heads of all the CP and vP phases between the merge site of *who* and its eventual target site in the matrix CP have an EPP type of feature OCC (for occurrence of) that has the function of letting elements escape phases to have their interpretative features checked elsewhere. The ad hoc nature of this feature/function is painfully obvious. The OCC feature/function lacks phonetic motivation and semantic motivation—there is, in short, no interface reasons to have such a groundless feature. And yet Chomsky finds it necessary to impose this feature/function on his framework for what appears to be one and only one reason: it ensures that Internal Merge must be constrained by Agree. Should Internal Merge not involve some version of local Agree, it will lapse into the Move operation, with all the attendant processing problems. Hence, OCC must exist to save Internal Merge, which must exist to keep the entire framework from sliding down into the processing abyss. The OCC feature/function hypothesis, it seems to me, does not follow from conceptual necessity; rather, it follows from theory self-preservation: from the need to protect the theory itself.

But let us exercise some charity and follow Chomsky in assuming that OCC (or EPP) does exist despite the stipulative nature of this assumption. Having the OCC creates two types of problems—one empirical, the other processing-related. The empirical problem in having an OCC feature/

function is that it should be available in the embedded CPs for both (5a) and (5b); as a result, the *wh*-subject in (5a) should be as free to escape the embedded CP phase as is the *wh*-object in (5b) and these two sentences should be equally grammatical.

(5) a. *Who does John believe [that (who) hired Sam]
    b. Who does John believe [that Sam hired (who)]

The grammatical differences in (5), then, would seem to be unexpected under Chomsky's analysis. Relatedly, the fact that the OCC feature will allow the *wh*-element in (5b) to escape the embedded CP should lead us to expect that it will also allow the *wh*–in situ element in (6) to escape.

(6) Who believes that [Sam fired who]

That is, under Chomsky's analysis, we should erroneously predict that (6) should be able to have a pair-list reading (see my discussion of pair-list interpretations in chapter 3).

The processing problem with Chomsky's OCC analysis is that the avowedly optional OCC feature can force us to needlessly waste processing time. To see this, let's take a careful look at (4). In (4), the four intermediate phases must all have the optional OCC feature if the *wh*-element is to make its way to the matrix CP. However, any one of these phases P could not have an optional OCC. Should this occur, the derivation will crash because the *wh*-element will not pass through P and will not have its [WH] feature checked appropriately. All of the processing effort we would have devoted to computing the derivation for (4) will have been wasted and we will have to start the derivation again. But the next derivation could crash, too, for the same reasons just mentioned, and if we cannot carry past derivational histories with us as we restart our computation of (4), we might find ourselves endlessly computing partial derivations that culminate in crashes. A similar situation arises in complex sentences without *wh*-elements, such as (7).

(7) John believes that Mary told you that she would fire Sam

Aside from the *wh*-element, (7) is structurally similar to (5), having all the same phases that (5) has. Since each phase can have an OCC feature, or not, the six phases in (7) all have the possibility of having an OCC feature. If any one of these phases P has such a feature, however, the derivation will crash because the OCC will not be checked by an element recopied (moved) by Internal Merge into phase P. And, again, crashes are expensive: they require that one restart the failed derivation, without any appreciable guarantee that the next computation will be successful. Furthermore, the more phases a derivation has, the greater the possibility that some head of a phase will select the wrong option for the OCC feature and end with a crash. Processing sentences with numerous phases, then, could be exceedingly expensive in terms of processing time. Given Frampton and Gutmann's (2002) arguments that derivations should be crash-proof, the OCC hypothesis, with its potential for inducing crashes, would seem ill-conceived. And yet there is something instructive to be learned from the OCC hypothesis about processing syntactic derivations. That is, if we could correctly predict when the OCC feature must show up and when it must not show up, then our processing demands would substantially decrease because we could avoid costly crashes. Making such a prediction, however, would require that we know what features already in a derivation must be passed along. In other words, to avoid expensive crashes, a derivation will have to "know" which features of its elements have to depend on the presence, or absence, of the OCC option. But if the OCC feature merely registers the existence of another feature, why have the OCC at all? Doesn't the feature announce itself? It would appear, then, that the passing of any element through a phase is not about the OCC, but about the features of the displaced element themselves. This suggests that the OCC hypothesis lacks conceptual necessity and should be surrendered, as should the Internal Merge hypothesis, which crucially depends on the OCC hypothesis (and its EPP variant).

Our discussion thus far leads to the following conclusions: the Move operation is too powerful; the Internal Merge operation, an attract-based (or agree-based) version of the Move operation, is stipulative and costly; and, as our analysis of the OCC hypothesis informs us, displacement is not about head attraction, but about the ongoing presence (the survival) of the displaced element's features. These conclusions square with the proposals I make in this book. Throughout the book, I argue that movement-type operations—such as Move or Internal Merge—cannot explain displacement phenomena. As I contend, if we appeal to movement-type operations to account for displacement, we essentially end up with the fatuous and uninformative argument that things get displaced because they move (or get recopied as some distance away). Furthermore, there does not appear to be any acceptable way to undergo movement. To see this, let's consider how the element YP might move after it merges with head H, as in (8).

(8) Merge $\langle H, YP \rangle \rightarrow H\ YP$

Should YP move, this movement will have to be induced by one or more of the following factors: the properties of YP (this is assumed by Chomsky's (1993) Principle of Greed); the properties of H (this is assumed by the Repel Principle I consider in chapter 2); and/or the properties of another head H2 outside the HP projection (this is assumed by all attract-type analyses of movement). As I discuss in chapters 1 and 2, none of the properties listed above can successfully drive syntactic movement. In grossly simplified terms, the reason that the properties of YP and H cannot induce movement is that these movement-inducing properties would be in effect at the time of Merge and these properties should compel the YP to move at that time; however, there would not be anywhere for YP to move to at this point in a derivation except for SpecHP, and then the still-active properties of YP and/or H should continue to drive YP movement, sending it to another SpecHP position, and then to another such position, and so on. If we try to delay YP-induced or H-induced movement until we derive enough structure to provide a landing site for the movement, we not only create the processing problem of continually recycling our computations by having to return to every HP we construct to look for elements with delayed-movement properties, but we also tacitly undermine our assumption that the properties of YP or H induce movement by requiring other factors to trigger the movement. Hence, we cannot attribute movement to the properties of YP or H. What about attributing movement to the properties of H2, another head outside HP, as is assumed in all attract (and agree) analyses of movement? Such a movement has some of the delayed-movement problems mentioned above. In particular, if the properties of H2 reach down in the derivation and induce YP to move, then we will have to recycle our computations, returning to HP to find some YP that must move for feature-checking purposes and then move the YP out of HP. All of this searching and moving (recopying) will force us to recompute HP and all the structure between HP and H2P. In addition, as we observed in our discussion of Chomsky's OCC hypothesis, there is no viable attract/agree mechanism available to break long-distance displacement into a series of short-distance displacements; consequently, H2 will have to be able to have a long enough reach to move the *wh*-element in (9) from its merge site to the matrix CP.

(9) Who does John believe that Mary told you that she would fire (who)

Once we permit this sort of movement, then we are back to not only having a powerful Move operation, but also to requiring Economy/

Minimality Conditions to limit the reach of Move. The processing problems created by a look-back Move operation as powerful as the one just described would be enormous.

These sorts of arguments have led me to conclude that move-type operations cannot be responsible for the displacement property of human language. However, as we inferred from our discussion of the OCC hypothesis, the features of the displaced element are necessarily involved in the displacement phenomena. Nonmovement, feature-based operations, therefore, must be responsible for displacement. The bulk of my book is devoted to identifying and examining such operations. Building on the widely help hypothesis that checked features are derivationally deactivated (see, for example, Collins 1997), I propose that only features that SURVIVE the checking process are involved in displacement; and eschewing movement operations, I propose that elements can undergo displacement—that is, show up in more than one structural site—because they can be Merged from the Numeration more than once. The second Merge, and all subsequent Merges, I call Remerge. I argue in this book that the natural interaction of the SURVIVE and Remerge operations produces the displacement property of HL. Displacement, then, is a "free" property of HL, in accordance with Chomsky's (2002, 2005) most recent view. The syntax of HL, under my analysis, consists of two feature-checking operations—Merge and Remerge. Importantly, these two operations are strictly local operations; as a result, they have neither look-back nor look-ahead properties that could dramatically complicate syntactic processing. Having only local, feature-driven Merge-type operations is, I believe, a radically simple design for the computational system of HL. It is, after all, a design that allows only one type of operation OP: OPs that map elements from the Numeration to the Syntactic Derivation (OP: N $\rightarrow$ SD). But not only is my design maximally simple, it is also a design dynamic enough to explain, in a natural way, a range of syntactic phenomena that have resolutely resisted previous generative theories—these phenomena include *that*-trace effects, Superiority Effects, and interpretation differences available for multiple-*wh* constructions. That my simple design for the computational system of HL is able to account for a vast array of complex displacement phenomena makes it both economical and plausible.

Although my reanalysis of minimalism answers some of the longstanding questions that have challenged syntactic theory, it also raises many other theoretically interesting questions, such as whether my computational design of HL can account for scopal effects, reconstruction effects, binding effects, and so on. I leave these important matters for further investigation. However, I do want to make some brief comments on how my analysis might address two other controversial challenges: syntactic lowering and head movement.

Bošković and Takahashi (1998) and Bošković (2004) argue that syntactic lowering is needed to explain scrambling in Japanese and other languages (Richards (2004) also raises the possibility that lowering might be involved in some *wh*-constructions in Slavic languages). These analyses maintain that the scrambled constituent in (10) *Sono hon-o* is base generated in the matrix IP and later lowered to the position marked as *t* to receive Case and a theta-role.

(10) [$_{IP}$ Sono hon-o [$_{IP}$ John-ga [[Mary-ga [t katta]] to] omotteiru]]
That book-ACC John-NOM Mary-NOM bought that thinks
'That book, John thinks that Mary bought.'

Since my analysis of syntax allows only structure-building operations that lack both look-back and look-ahead properties, any syntactic lowering operations such as the one proposed above should, in principle, be impossible not only because they would require HL to have a mixed (derivational and representational) design—something that Brody (2002) argues against—but also because they would place an enormous burden on processing. To see this, consider sentence (11), in which a base generated constituent *Daremo-ni* is scrambled, but there is no place to lower it.

(11) *[$_{IP}$ Daremo-ni [$_{IP}$ dareka-ga
Everyone someone-NOM
[Mary-ga sono hon-o katta to] omotteiru]]
Mary-NOM that book-ACC bought that thinks
'Everyone, someone thinks that Mary bought that book.'

The problem with (11) for the lowering analysis is that this analysis permits us to compute IP structures such as (11), and it will not allow us to rule out (11) until we recompute (look for) all the possible lowerings, hoping to find one that will Case-license and theta-license the scrambled constituent. Needless to say, all this fruitless computing and recomputing of ill-formed structures is incompatible with O'Grady's (2005, 6) observation that the computational system of HL should be efficient: it should "minimize the burden on the working memory." That syntactic lowering, as seen in (11), maximizes the burden on the working memory, by legitimating the various computations I list above, while not producing a well-formed output, suggests that such operations lack plausibility—a conclusion much in line with my analysis of syntactic derivation.

As with syntactic lowering, head movement is highly controversial. Several theorists such as Chomsky (2000a), Koopman and Szabolcsi (2000), and Mahajan (2000) have argued that head movement should be excluded from the narrow syntax, in part because head movement seems to be have properties quite distinct from phrasal movement. Others such as Kural (2005) and Matushansky (2006) have argued that head movement is a viable and necessary syntactic operation. Matushansky (2006, 71) even claims that "head movement and phrasal movement are triggered by the same factor and are in fact instances of the same phenomenon (feature valuation followed by (Re)merge)." Given that all syntactic objects (SOs)—which includes both heads and phrases—consist of sets of features, each of which must be checked for concatenative integrity, my version of minimalism does not discriminate heads from phrases in terms of the feature-checking operation. That is, under my analysis, both heads and phrases, as SOs, should be liable to undergo Remerge (to check unchecked features); hence, they should equally exhibit displacement (i.e., "movement"). Furthermore, under my analysis, since SOs all Remerge for the same reason and they Remerge in the same way (in the next available HP), I would concur with Matushansky that "head movement" (i.e., head Remerge) and "phrasal movement" (phrase Remerge) are in fact the same phenomena. I also agree with Matushansky that a head will Remerge in the Spec position of the next head, as in (12), where the head Y^0 "moves" into XP.

(12) $[_{XP}$ Y^0 $[X^0$ $[_{YP}$ ZP $[Y^0$ WP]]]]

In other words, Y^0 Remerges in exactly the same position a displaced phrase would. (Of note, the fact that heads must behave like all other Remerged SOs by remerging into the next XP provides a natural explanation for the Travis's (1984) Head Movement Constraint, which requires head movement not to skip intermediate heads.) Despite agreeing with much of Matushansky's analysis, I disagree with it in a fundamental way. For me, head "movement" is not a movement at all, but a Remerge phenomenon. As such, a head with an unchecked feature will locally Remerge with every available newly Merged head until all unchecked features are eventually appropriately checked. If any unchecked feature is not ever checked in the derivation, then the derivation will stall before it reaches the interfaces. We can see how I would explain "head movement" by looking at a derivation for (13).

(13) [$_{CP}$ Trinkst C [$_{TP}$ du trinkst T [$_{vP}$ trinkst v [$_{VP}$ trinkst Bier]]]]
'Drink you beer?' (Do you drink beer?)

In (13), the verb *trinkst* has a [V] category feature, a [VFORM] feature (this verbal feature marks the morphological form of the verb; see Stroik 2001 for a discussion of [VFORM]), a [TENSE] feature, and a [Q] feature). The derivation of (13) proceeds as in (14).

(14) a. Merge ⟨trinkst, V⟩ (the verb must first check its [V] category feature before it can Merge with its category-selected arguments)
b. Merge ⟨⟨trinkst, V⟩, Bier⟩ → trinkst Bier
c. Merge ⟨v, VP⟩ → v trinkst Bier
d. Remerge ⟨trinkst, ⟨v, VP⟩⟩ → trinkst v trinkst Bier (automatic Remerge, which checks the [VFORM] feature of *trinkst*, precedes Merge in step (e))
e. Merge ⟨du, ⟨v, VP⟩⟩ → du trinkst v trinkst Bier
f. Merge ⟨T, vP⟩ → T du trinkst v trinkst Bier
g. Remerge ⟨trinkst, ⟨T, vP⟩⟩ → trinkst T du trinkst v trinkst Bier (Remerge, which checks the [TENSE] of *trinkst*, precedes the next Merge, and the Remerge in (g) also precedes the Remerge in (h) because Remerge is a Top-Down storage operation)
h. Remerge ⟨du, ⟨trinkst, ⟨T, vP⟩⟩⟩ → du trinkst T du trinkst v trinkst Bier
i. Merge ⟨C, TP⟩ → C du trinkst T du trinkst du trinkst v trinkst Bier
j. Remerge ⟨trinkst, ⟨C, TP⟩⟩ → trinkst C du trinkst T du trinkst v trinkst Bier (Remerge checks the [Q] feature of *trinkst*)

In accordance with the approach to head "movement" I outline above, the verb *trinkst* in (14) undergoes four different head mergings ("movements") with four different heads—V, v, T, and C. Importantly, these various mergings occur not because the verb is being attracted to a higher head (as Matushansky asserts), but because the verb has features that continue to SURVIVE the derivation until it is finally Remerged in the matrix CP and its last remaining feature (the [WH feature]) can be checked. There are two interesting consequences of the derivation I give in (14). First, notice that under my analysis head-Remerge will precede phrase-Remerge (we can see this in (14g–h)). This ordering of Remerge is the result of the Top-Down storage. That is, since lexical heads are the first constituents to merge in HP, they will also be the first constituents to Remerge in the next phrase H2P (this explains why the Remerged head shows up after the Remerged phrase in (15)).

(15) Was trinkst du?

Second, the fact that a head will continue to Remerge until its features are checked (or until the derivation exhausts itself) raises the possibility that a head could exhibit "long-distance head movement" effects (in apparent violation of Travis's Head Movement Constraint). The existence of such cases, as observed in Lema and Rivero 1990 and in Embick and Izvorski 1995, offers a little subtle support to my analysis of syntactic computation.

I offer my much too brief remarks on syntactic lowering and on head movement to indicate some of the further directions to which my analysis of minimalism might be taken, and to invite other syntacticians to investigate the analytical possibilities that my SURVIVE version of minimalism might afford.

Notes

Chapter One

1. The existence of PRO is currently under serious debate. Hornstein (1999, 2000, 2003) has argued that PRO does not exist and that structures that in the past posited a PRO constituent should be reanalyzed in terms of the syntactic movement (raising) of a lexicalized constituent. That is, (ia) should be analyzed not as (ib), but as (ic).

(i) a. Chris wants to leave.
 b. Chris wants PRO to leave.
 c. Chris wants (Chris) to leave.

Landau (2003) argues against Hornstein's reanalysis of PRO, noting that this reanalysis incorrectly predicts that sentences such as (iia) should be well formed and that (iib) and (iic) should be equally well formed.

(ii) a. *John was hoped to win the game
 b. One interpreter each was assigned to the visiting diplomats
 c. *One interpreter each tried to be assigned to the visiting diplomats.

Although Boeckx and Hornstein (2004) have responded to Landau's critiques, the existence of PRO has not been conclusively settled. Until the existence of PRO is decided, I will continue to follow the generally accepted assumption that PRO does exist.

In addition to questions about the existence of PRO, there are also questions about the Case features of PRO. Cecchetto and Oniga (2004) argue that PRO does not have a null Case feature; rather, PRO shares the Case of its controller. This may be true in Latin, as Cecchetto and Oniga demonstrate in their analysis of (iii).

(iii) Ego volo [PRO esse bonus]
 I (NOM) want PRO to-be good (NOM)
 'I want to be good.'

However, it is unclear how this might apply to English given that subjects of infinitives cannot tolerate Nominative Case; compare (iva) and (ivb).

(iv) a. Chris wants PRO to leave
b. Chris wants me/*I to leave

I leave the issues raised by Cecchetto and Oniga for further research.

2. Lasnik and Park (2003) come to a similar conclusion—that the EPP is a configurational requirement—in their analysis of subjects in sluicing data. Matsubara (2000) and Sasaki (2000) add to this by showing that the EPP is not only a configurational requirement, but also an optional requirement.

3. There are alternatives to Chomsky and Lasnik's LF-movement analysis of antecedent-anaphor relations. Reinhart and Reuland (1993) and Pollard and Sag (1992) contend that reflexivity is a property of predicates. They argue, in particular, that a predicate will be reflexive if and only if two of its arguments are coindexed and that if a semantic predicate is reflexive, either it must have a morphologically complex SELF anaphor (such as *herself*) or it must be a lexically reflexive predicate (such as *behave*). Under this analysis, antecedents and anaphors are locally related not because of the effect that agreement domains have on them, but because they must be coarguments of a predicate.

Zwart (2002) and Kayne (2002) offer yet another analysis of antecedent-anaphor relations. They propose that an anaphor and its antecedent are merged into the syntax as a unit and that the antecedent subsequently moves out of this unit configuration to have its own features checked. That is, they analyze (ia) in the following way: the verb *likes* merges with the complex antecedent-anaphor unit [Pat [herself]], as in (ib); then the DP *Pat* moves out of the complement position to have its thematic and Case features checked in the subject position, as in (ic).

(i) a. Pat like herself
b. likes [Pat [herself]]
c. Pat likes [(Pat) [herself]]

For Zwart and for Kayne, antecedents and anaphors are locally related not because of the effects that agreement or predicate domains have on them, but because they are merged together as a complex unit. I have opted to discuss Chomsky and Lasnik's analysis instead of Reinhart and Reuland's or Zwart's because the former (agreement-based) analysis can extend to cover the object-agreement effects noted by Woolford (1999), whereas it is unclear how the other analyses could account for the fact that languages with object agreement fail to license object anaphors.

4. If I am correct in asserting that lexical features are inherently interface-compatible, then the distinction that Chomsky (2001) and Epstein and Seely (2002b) make between "valued" (syntactically checked) and "unvalued" (syntactically unchecked) has no basis. In addition to the arguments I have already made for having valued features in the lexicon (and in the Numeration), consider the following subsidiary argument. Let's assume that the difference between the lexical item *she* and the lexical item *her* resides in the values assigned to features (actually, a single feature in this case). Let's also assume that these lexical items have only unvalued features prior to syntactic checking, at which point the fea-

tures are assigned values. Under these two assumptions, in our mental lexicon we will not be able to discriminate *she* from *her* because they will have the same unvalued features (in fact, it is hard to see how we could discriminate *she* from *they* or *them* in our mental lexicon). But this leaves us with a problem. If we cannot discriminate these two lexical items in our mental lexicons, what exactly do we take out of the lexicon and place in a Numeration? Some form that looks like [Person, Number, Gender, Case] without any values assigned to these features? Do we really have such a form (a generic pronoun form) stored in our lexicon—a form that is not a lexical item? This seems unlikely and, more importantly, it seems quite divorced from our minimalist assumption that a Numeration includes only lexical items.

Chapter Two

1. I generally follow Chomsky 1995, Brody 1998, and Lopez 2000 in assuming that the merged category Merge⟨A,B⟩ projects the features of A or of B, but not a union or an intersection of these features. Brody calls this property Uniqueness, which he formulates as (i).

(i) *Uniqueness*
Every phrase is projected by a unique category.

However, in chapter 3, I revisit this issue.

2. Chomsky (1995) argues that it is possible for a feature to be checked but not to deactivate. According to Chomsky, the EPP feature of a DP, which is checked in TP, may be one such feature because a DP can have this feature checked in more than one TP, as in (i).

(i) [$_{TP}$ Sam [was expected [$_{TP}$ [(Sam) to be hired (Sam) soon]]]]

However, in chapter 1, I contend that the EPP feature is not a legitimate feature because it cannot be interpreted at the performance interfaces. Once we rule out the EPP feature as a lexical feature, then there do not seem to be any features that can be checked and later rechecked. That is, there do not seem to be any features that can be checked but not deactivated.

3. I informally introduced the SURVIVE Principle in Stroik 1996a.

4. In (22), I assume, as do Chomsky (1995) and Ura (2000), that an XP can have multiple Spec positions.

5. Although Chomsky (1994, 1995, 2000a) observes that having lexical access is unwieldy, and therefore that it is necessary to have a Numeration (a collection of lexical items gathered before the onset of the derivation) to simplify the computation of the derivation, Collins (1997) argues against this notion of Numeration. Collins's position seem to be the correct one. After all, if Numeration is selected arbitrarily, as Chomsky (2000a) maintains, then there is an extremely high probability that the Numeration will have either too few lexical items for the derivation to yield an EXP, or too many lexical items, having unused lexical items in the Numeration at the end of the derivation. Chomsky is aware of these possibilities and

claims that a derivation will not converge in either of the above cases. However, he appears to underestimate the regularity with which a Numeration will fail to exhaustively map onto an EXP at the interfaces. In fact, if one were to select the lexical items in a Numeration arbitrarily, it would rarely be the case that the Numeration will map onto a well-formed EXP at the interfaces. This would require a language user to devote an excessive amount of effort to computing derivations that will crash. In the face of this argument, it seems to me that minimalist assumptions would lead one to eliminate Chomsky's notion of Numeration from the theory.

In my analysis, I also use the notion of a Numeration. My sense of Numeration, however, differs from Chomsky's. My Numeration is not a collection of randomly collected lexical items selected prior to the onset of a syntactic derivation; rather, it is a gradually accreted collection of items being used in a derivation. In the sentence "Pat loves her," the Numeration will begin with the verb {loves}, which will require an Object; to fulfill this latter requirement an LI {her} will be brought into the Numeration from the lexicon, creating the expanded Numeration {loves, her}, and so on. In this sense, a Numeration is a lexical buffer of the elements selected for a derivation as it develops.

One might wonder why we need a lexical buffer (a Numeration) at all. Why not map elements directly from the lexicon to the derivation itself? There is one very important reason for having a lexical buffer (a Numeration), rather than a direct merging from the lexicon to a derivation. The operation Merge does not apply to lexical items; it applies to (individual) features of lexical items. A Numeration is the place where lexical items (which are terms, or perhaps labels, plus a variety of phonetic, morphological, semantic, discourse, and cultural features) are converted/reduced to sets of features that are interpretable at the interfaces. It is to these features that syntactic operations such as Merge apply. In this way, a Numeration is a work space for interpretable features.

6. Epstein, Groat, Kawashima, and Kitahara (1998) also propose Remerge. Their version of Remerge, however, is not motivated by feature incompatibility, as it is in my SURVIVE Principle. Relatedly, Aoun and Li (2003) propose Demerge (which copies elements back into the Numeration) and the eventual Remerging of Demerged elements. Although these two operations have oblique similarities to SURVIVE, they are not sensitive to feature compatibility and have different applicational properties—for example, Aoun and Li's Remerge is not automatic and does not appear to permit multiple remergings of elements.

Furthermore, my proposal is that the "Remerge" buffer is a subbuffer of the WorkBench. The Remerge buffer could also be an independent buffer not related in any way to the WorkBench. Since my SURVIVE analysis goes through in either case, I will not take up the issue of the independence of the Remerge buffer, save for pointing out that since the Remerge buffer is a lexical buffer of (the features of) LIs and the WorkBench/Numeration includes a lexical buffer of (the features of) LIs, it would appear to simplify the theory by having only a single lexical buffer.

7. Another way to represent the complex feature [REF/WH] would be as a structured feature [REF [WH]] (such feature representations would be akin to feature

matrices in theories such as HPSG). In the structured feature [REF [WH]], the WH-feature would be a subfeature of the referentiality feature.

8. That the DP *a picture of whom* is not referential can be seen in (i).

(i) *Who saw [a picture of whom]$_i$. It$_i$ was beautiful.

According to Cinque (1990), a DP can be linked to a pronoun that it does not c-command if and only if the DP is referential (as in (ii)).

(ii) John bought [a picture]$_i$. It$_i$ was very expensive.

The fact that the DP in (i) cannot be linked to the pronoun *it* suggests that the DP is not referential.

Chapter Three

1. I follow Richards (2001, 105) in assuming that "PF must receive unambiguous instructions about which part of a chain to pronounce." I also follow Richards in assuming that the sensorimotor interface pronounces elements where the relevant morphophonetic features are checked.

2. I come to a similar conclusion in Stroik 1992, 1995, 1996b.

3. It is generally thought that antecedents must c-command anaphors to explain the contrasts in (i).

(i) a. John sold Mary to herself
b. *John sold herself to Mary

This c-command relationship, however, must be "local" in some sense because the distal DP *John* cannot serve as the antecedent for the anaphor *himself* in (ii).

(ii) *John thinks that Mary likes himself

In (39), I follow Chomsky and Lasnik (1995) in defining the local nature of this c-command relationship in terms of immediate (or closest) c-command.

4. Although the light verb *v* in English lacks agreement features for Person, Number, and Gender, it has, at the very least, a Case agreement feature. The light verb, then, has partial agreement features. Despite having some features, the light verb, as I discuss in (40a), cannot deactivate the [AGR/] feature of a reflexive because, in part, it does not have all the agreement features necessary to license the agreement features of the reflexive. However, when the light verb checks the features on another object argument—for example *Bill* in (i)—it arguably has the agreement features of the object-argument available to license the agreement features of a reflexive. This explains why the reflexive in (i) can have its features licensed in vP, thereby permitting the object *Bill* to be the antecedent for the reflexive.

(i) John told Bill about himself

Though the light verb can inherit agreement features from its object, it does not have an inherent [AGR] feature, so the [AGR/] reflexive can also SURVIVE the vP and force the reflexive to remerge in the TP, where its agreement dependency can be satisfied. In this case, the subject *John* will be the antecedent for the reflexive.

5. The fact that object agreement plays a role in the licensing of reflexives is, on the surface, problematic for theories of reflexivity that correlate reflexivity with syntactic and/or semantic predicates—theories such as those by Reinhart and Reuland (1993) and by Pollard and Sag (1992).

About object agreement, Woolford notes that languages that have this agreement can license reflexives if they have a special reflexive morpheme. In Swahili, for example, a (null) reflexive can show up in constructions that have the reflexive object morpheme *ji* (see (i)).

(i) Ahmed a-na-ji-penda
Ahmed 3SUBJ-PRES-REFL-love
'Ahmed loves himself.'

The reflexive morpheme, however, is not marked for agreement features, as are the object agreement morphemes. We can see the unmarked object-agreement features in (ii).

(ii) Ahmed a-na-m-penda Halima
Ahmed 3SUBJ.SG-PRES-3OBJ.SG-love Halima
'Ahmed loves Halima.'

The difference between the object markers in (i) and (ii) would seem to suggest that the vP with the reflexive marker does not have the [AGR] feature that is in nonreflexive constructions. Consequently, the (null) reflexive will not have its [AGR/] feature checked in vP; and the reflexive will then have to remerge in the TP, where it can have its features licensed.

6. Pesetsky (1982) and Richards (2001) argue that the example in (i) shows that a *wh*-operator can cross over another *wh*-operator—that is, that a *wh*-operator can climb over a C head with an [OP] feature. If they are correct, example (i) will be a counterexample to my analysis.

(i) what books do you know who to persuade to read

All my informants, however, find (i) to be unquestionably ungrammatical.

7. Getting a good look at the interaction between the adverb *why* and negation is problematic because, as Rizzi (1990) argues, this adverb can be base generated (i.e., merged) in the SpecCP position, thereby circumventing any direct interaction with the NEG head. The example I give in (69b) tries to force a relationship between *why* and negation, but it can do so only under the reading in which the adverb *why* modifies the embedded predicate. In this reading, the adverb will have to merge in the embedded sentence and eventually make its way through the NEGP as it iteratively remerges until it reaches the matrix CP.

8. Weak crossover violations occur when an operator "moves" over a DP that contains a pronoun bound by the operator. In such a case, the operator binds two different variables, neither of which c-commands the other. The *wh* "movement" in (i) produces a weak crossover structure.

(i) *Who$_i$ does his$_i$ mother like t_i

9. I am not alone in observing that not all multiple-*wh* constructions receive pair-list interpretations. Bošković (2002), Dayal (2002), and Aoun and Li (2003) make similar observations.

10. Another test for the presence of superiority effects is the *wh-else* test I develop in Stroik 2000. Multiple-*wh* constructions that allow pair-list interpretations (and do not exhibit Superiority Effects) can extend a pair-list reading by adding the word *else* to the *wh*-elements. We can see this in (ib), which differs from (ia) in that it overtly asks for an extended list of paired responses.

(i) a. Who read what
b. Who else read what else

On the other hand, multiple-*wh* constructions that do not allow pair-list interpretations (and do exhibit Superiority Effects) will not tolerate any attempt to extend the paired responses by adding the word *else* to *wh*-elements, as the examples in (ii) illustrate.

(ii) a. What did who read
b. #What else did who else read

If we apply this test to sentence (94b), we will see that, despite Kayne's claim to the contrary, sentence (94b) does have a Superiority Effect in it (see (iii)).

(iii) #What else did who else give to whom else
cp Who else gave what else to whom else

Furthermore, we can use this test to isolate the site of the Superiority Effect. Notice that (iva) is well formed, but (ivb) is not.

(iv) a. What else did who give to whom else
b. #What else did who else give to whom

This suggests that there is no Superiority Effect involving *what* and *whom*, but there is one involving *what* and *who*. For other tests relevant to pair-list readings (and Superiority Effect detection), see Stroik 2000.

11. Once again, we can also test for the presence (or absence) of Superiority Effects by using the *wh-else* test for (101), as in (i). (The grammaticality judgment in (101) is Richards's, not mine.)

(101) What did who persuade whom to buy

(i) #What else did who else persuade whom else to buy
cp Who else persuaded whom else to buy what else

The ungrammaticality of (i) strongly suggests that sentence (101) does have a Superiority Effect in it, contrary to the claims made by Richards (2001) and Aoun and Li (2003). Furthermore, as the data in (ii) indicate, the Superiority Effects in (101) involve the *wh*-operator's relation with *who* and its relation with *whom*. That is, there are actually two superiority violations in (101).

(ii) a. #What else did who else persuade whom to buy
b. #What else did who persuade whom else to buy

The *wh-else* test, then, supports the conclusion I have drawn from the *wh*-the-hell test.

12. Collins (1997) also assumes that the complementizer *that* can have a [WH] feature, though for quite different reasons.

13. The sentences in (132) also pass the *wh-else* test; see (i). This adds support for my claim that *wh*–in situ elements in TP infinitives participate in pair-list readings.

(i) a. Who else expects Sam to read what else
  b. Who else wants whom else to go where else
  c. What else does Chris expect Sam to read to whom else

References

Abraham, Werner, Samuel D. Epstein, Höskuldur Thráinsson, and Jan-Wouter Zwart. 1996. *Minimal Ideas*. Amsterdam: John Benjamins.

Ackema, Peter, and Ad Neeleman. 2000. Context-sensitive spell-out and adjacency. Unpublished manuscript, Utrecht University and University College London.

Adger, David. 2003. *Core Syntax*. Oxford: Oxford University Press.

Adger, David, and Gillian Ramchand. 2005. Merge and move: *Wh*-dependencies revisited. *Linguistic Inquiry* 36: 161–193.

Anderson, Stephen, and David Lightfoot. 2002. *The Language Organ*. Cambridge: Cambridge University Press.

Aoun, Joseph. 1985. *A Grammar of Anaphora*. Cambridge, Mass.: MIT Press.

Aoun, Joseph. 1986. *Generalized Binding*. Dordrecht: Foris.

Aoun, Joseph, and Yen-hui Audrey Li. 1993. *Syntax of Scope*. Cambridge, Mass.: MIT Press.

Aoun, Joseph, and Yen-hui Audrey Li. 2003. *Essays on the Representational and Derivational Nature of Grammar.* Cambridge, Mass.: MIT Press.

Baltin, Mark. 1995. Floating quantifiers, PRO and Predication. *Linguistic Inquiry* 26: 199–248.

Beck, Sigrid. 1996. Quantified structures as barriers for LF movement. *Natural Language Semantics* 4: 1–56.

Bobaljik, Jonathan. 1995. *Morphosyntax: The syntax of verbal inflection*. Unpublished doctoral dissertation, MIT, Cambridge, Mass.

Boeckx, Cedric, and Norbert Hornstein. 2004. Movement under control. *Linguistic Inquiry* 35: 431–452.

Bok-Bennema, Reineke. 1991. *Case and Agreement in Inuit*. Dordrecht: Foris.

Bošković, Željko. 1997a. On certain violations of the Superiority Condition, AgrO, and economy of derivation. *Journal of Linguistics* 33: 227–254.

Bošković, Željko. 1997b. Superiority and economy of derivation: Multiple wh-fronting. Paper presented at the Sixteenth West Coast Conference of Formal Linguistics, University of Washington, Seattle.

Bošković, Željko. 1997c. Superiority effects with multiple wh-fronting in Serbo-Croatian. *Lingua* 102: 1–20.

Bošković, Željko. 1999. On multiple feature checking: Multiple wh-fronting and multiple head movement. In Samuel Epstein and Norbert Hornstein, eds., *Working Minimalism*, 159–188. Cambridge, Mass.: MIT Press.

Bošković, Željko. 2002. On multiple-wh fronting. *Linguistic Inquiry* 33: 351–383.

Bošković, Željko. 2004. Topicalization, focalization, lexical insertion, and scrambling. *Linguistic Inquiry* 35: 613–638.

Bošković, Željko, and Daiko Takahashi. 1998. Scrambling and last resort. *Linguistic Inquiry* 29: 347–366.

Brody, Michael. 1995. *Lexico-Logical Form: A Radically Minimalist Theory*. Cambridge, Mass.: MIT Press.

Brody, Michael. 1998. Projection and phrase structure. *Linguistic Inquiry* 29: 367–398.

Brody, Michael. 2000. Mirror Theory: Syntactic representation in perfect syntax. *Linguistic Inquiry* 31: 29–56.

Brody, Michael. 2002. On the status of representations and derivations. In Samuel D. Epstein and T. Daniel Seely, eds., *Derivation and Explanation in the Minimalist Program*, 19–41. Oxford: Blackwell.

Browning, Margaret. 1996. CP recursion and that-t effects. *Linguistic Inquiry* 27: 237–256.

Bruening, Benjamin. 2006. Differences between the *wh*-scope marking and *wh*-copy constructions in Passamaquoddy. *Linguistic Inquiry* 37: 25–49.

Carstens, Vicki. 2003. Rethinking Complementizer Agreement: Agree with a Case-Checked Goal. *Linguistic Inquiry* 34: 393–412.

Cecchetto, Carlo, and Renato Oniga. 2004. A challenge to the null case theory. *Linguistic Inquiry* 35: 141–149.

Chametzky, Robert. 2003. Phrase structure. In Randall Hendrick, ed., *Minimalist Syntax*, 192–225. Oxford: Blackwell.

Chomsky, Noam. 1964. *Current Issues in Linguistic Theory.* The Hague: Mouton.

Chomsky, Noam. 1965. *Aspects of the Theory of Syntax*. Cambridge, Mass.: MIT Press.

Chomsky, Noam. 1977. *Essays on Form and Interpretation*. New York: North-Holland.

Chomsky, Noam. 1980. On binding. *Linguistic Inquiry* 11: 1–46.

Chomsky, Noam. 1981. *Lectures on Government and Binding*. Dordrecht: Foris.

Chomsky, Noam. 1986. *Barriers*. Cambridge, Mass.: MIT Press.

Chomsky, Noam. 1993. A Minimalist Program for linguistic theory. In Kenneth Hale and Samuel Jay Keyser, eds., *The View from Building 20*. Cambridge, Mass.: MIT Press.

Chomsky, Noam. 1994. Bare phrase structure. In Gert Webelhuth, ed., *Government and Binding and the Minimalist Program*. Oxford: Blackwell.

Chomsky, Noam. 1995. *The Minimalist Program*. Cambridge, Mass.: MIT Press.

Chomsky, Noam. 2000a. Minimalist inquiries: The framework. In Roger Martin, David Michaels, and Juan Uriagereka, eds., *Step by Step*, 89–155. Cambridge, Mass.: MIT Press.

Chomsky, Noam. 2000b. *New Horizons in the Study of Language and Mind*. Cambridge: Cambridge University Press.

Chomsky, Noam. 2001. Derivation by phase. In Michael Kenstowicz, ed., *Ken Hale: A Life in Language*, 1–52. Cambridge, Mass.: MIT Press.

Chomsky, Noam. 2002. Beyond explanatory adequacy. Unpublished manuscript, MIT, Cambridge, Mass.

Chomsky, Noam. 2005. Three factors in language design. *Linguistic Inquiry* 36: 1–22.

Chomsky, Noam, and Howard Lasnik. 1977. Filters and control. *Linguistic Inquiry* 8: 425–504.

Chomsky, Noam, and Howard Lasnik. 1995. The theory of principles and parameters. In Noam Chomsky, *The Minimalist Program*, 13–127. Cambridge, Mass.: MIT Press.

Chung, Sandra. 1994. Wh-Agreement and referentiality. *Linguistic Inquiry* 25: 1–44.

Cinque, Guglielmo. 1990. *Types of A-Dependencies*. Cambridge, Mass.: MIT Press.

Cinque, Guglielmo. 1999. *Adverbs and Functional Heads*. Oxford: Oxford University Press.

Clifton, Charles, Gisbert Fanselow, and Lyn Frazier. 2006. Amnestying superiority violations: Processing multiple questions. *Linguistic Inquiry* 37: 51–68.

Collins, Chris. 1997. *Local Economy*. Cambridge, Mass.: MIT Press.

Collins, Chris, and Höskuldur Thráinsson. 1993. Object shift in double object constructions and the theory of Case. In Colin Phillips, ed., *MIT Working Papers in Linguistics* 19: 131–174.

Culicover, Peter. 1993. The adverb effect: Evidence against ECP accounts of the that-t effect. In A. Schafer, ed., *NELS 23*, 97–110. Amherst, Mass.: GLSA, University of Massachusetts.

Dayal, Veneeta. 2002. Single-pair versus multiple-pair answers: Wh-in-situ and scope. *Linguistic Inquiry* 33: 512–520.

Embick, David, and Roumyana Izvorski. 1995. On long head movement in Bulgarian. In Janet M. Fuller, Ho Han, and David Parkinson, eds., *ESCOL '94*, 104–115. Ithaca, N.Y.: CLC Publications, Cornell University.

Engdahl, Elisabet. 1983. Parasitic gaps. *Linguistics and Philosophy* 6: 5–34.

Epstein, Samuel, Erich Groat, Ruriko Kawashima, and Hisatsugu Kitahara. 1998. *A Derivational Approach to Syntactic Relations*. Oxford: Oxford University Press.

Epstein, Samuel, and T. Daniel Seely. 2002a. Introduction: On the quest for explanation. In Samuel D. Epstein and T. Daniel Seely, eds., *Derivation and Explanation in the Minimalist Program*, 1–18. Oxford: Blackwell.

Epstein, Samuel, and T. Daniel Seely. 2002b. Rule applications as cycles in a level-free syntax. In Samuel D. Epstein and T. D. Seely, eds., *Derivation and Explanation in the Minimalist Program*, 65–89. Oxford: Blackwell.

Epstein, Samuel, Höskuldur Thráinsson, and Jan-Wouter Zwart. 1996. Introduction. In Werner Abraham, Samuel Epstein, Höskuldur Thráinsson, and Jan-Wouter Zwart, eds., *Minimal Ideas*, Amsterdam: John Benjamins.

Fanselow, Gisbert, and Damir Ćavar. 2001. Remarks on the economy of pronunciation. In Gereon Müller and Wolfgang Sternefeld, eds., *Competition in Syntax*, 107–150. Berlin: Mouton de Gruyter.

Felser, Claudia. 2004. *Wh*-copying, phases, and successive cyclicity. *Lingua* 114: 543–574.

Fiengo, Robert, C.-T. James Hunag, Howard Lasnik, and Tanya Rreinhart. 1988. The syntax of wh-in-situ. In Hagit Borer, ed., *WCCFL 7*. Stanford, CA: CSLI.

Fitzpatrick, Justin. 2002. On minimalist approaches to the locality of movement. *Linguistic Inquiry* 33: 443–463.

Fox, Danny. 2000. *Economy and Semantic Interpretation*. Cambridge, Mass.: MIT Press.

Frampton, John. 1997. Expletive insertion. In Chris Wilder, Hans-Martin Gärtner, and Manfred Bierwisch, eds., *The Role of Economy in Linguistic Theory*. Berlin: Akademie Verlag.

Frampton, John, and Sam Gutmann. 2002. Crash-proof syntax. In Samuel D. Epstein and T. Daniel Seely, eds., *Derivation and Explanation in the Minimalist Program*, 90–105. Oxford: Blackwell.

Freiden Robert. 1999. Cyclicity and minimalism. In Samuel D. Epstein and Norbert Hornstein, eds., *Working Minimalism*, 95–126. Cambridge, Mass.: MIT Press.

Gazdar, Gerald, Ewan Klein, Geoffrey Pullum, and Ivan Sag. 1985. *Generalized Phrase Structure Grammar*. Cambridge, Mass.: Harvard University Press.

Groat, Erich. 1999. Raising the Case of Expletives. In Samuel D. Epstein and Norbert Hornstein, eds., *Working Minimalism*. Cambridge, Mass.: MIT Press, 27–44.

Groat, Erich, and John O'Neil. 1996. Spell-out at the LF interface. In Werner Abraham, Samuel Epstein, Höskuldur Thráinsson, and Jan-Wouter Zwart, eds., *Minimal Ideas*, 113–139. Amsterdam: John Benjamins.

Haegeman, Liliane. 1991. *Introduction to Government and Binding Theory*. Oxford: Blackwell.

Haegeman, Liliane. 1992. Some speculations on argument shift, clitics and crossing in West Flemish. Unpublished manuscript, University of Geneva.

Haegeman, Liliane. 2003. Notes on long adverbial fronting in English and the left periphery. *Linguistic Inquiry* 34: 640–649.

Haegeman, Liliane. 2004. A DP-internal anaphor agreement effect. *Linguistic Inquiry* 35: 704–712.

Harris, Zellig. 1946. From morpheme to utterance. *Language* 22: 161–183.

Haugen, Einar. 1951. Directions in modern linguistics. *Language* 27: 211–222.

Hazout, Ilan. 2004. Long-distance agreement and the syntax of *for-to* infinitives. *Linguistic Inquiry* 35: 338–343.

Hendrick, Randall, and Michael Rochemont. 1982. Complementation, multiple wh and echo questions. Unpublished manuscript, University of North Carolina and University of California, Irvine.

Hoekstra, Jarisch, and Lazlo Maracz. 1989. On the position of inflection in West Germanic. *Working Papers in Scandinavian Syntax* 44: 75–88.

Hornstein, Norbert. 1984. *Logic as Grammar*. Cambridge, Mass.: MIT Press.

Hornstein, Norbert. 1995. *Logical Form*. Oxford: Blackwell.

Hornstein, Norbert. 1999. Movement and control. *Linguistic Inquiry* 30: 69–96.

Hornstein, Norbert. 2000. *Move! A Minimalist Theory of Construal*. Oxford: Blackwell.

Hornstein, Norbert. 2003. On control. In Randall Hendrick, ed., *Minimalist Syntax*, 1–81. Oxford: Blackwell.

Huang, C.-T. James. 1982. Logical relations in Chinese and the theory of grammar. Unpublished doctoral dissertation, MIT, Cambridge, Mass.

Jackendoff, Ray. 1977. *X-bar Syntax*. Cambridge, Mass.: MIT Press.

Jackendoff, Ray. 2002. *Foundations of Language*. Oxford: Oxford University Press.

Johnson, David, and Shalom Lappin. 1999. *Local Constraints vs. Economy*. Stanford: CSLI.

Kayne, Richard. 1983. Connectedness. *Linguistic Inquiry* 14: 223–249.

Kayne, Richard. 1994. *The Antisymmetry of Syntax*. Cambridge, Mass.: MIT Press.

Kayne, Richard. 2002. Pronouns and their antecedents. In Samuel D. Epstein and T. Daniel Seely, eds., *Derivation and Explanation in the Minimalist Program*, 133–166. Oxford: Blackwell.

Kitahara, Hisatsugu. 1997. *Elementary Operations and Optimal Derivations*. Cambridge, Mass.: MIT Press.

Koopman, Hilda, and Anna Szabolcsi. 2000. *Verbal Complexes*. Cambridge, Mass.: MIT Press.

Kuno, Susumu, and Jane Robinson. 1972. Multiple wh-questions. *Linguistic Inquiry* 3: 463–487.

Kural, Murat. 2005. Tree traversal and word order. *Linguistic Inquiry* 36: 367–387.

Landau, Idan. 2003. Movement out of control. *Linguistic Inquiry* 34: 471–498.

Lasnik, Howard. 1995. Case and expletives revisited: On greed and other human failings. *Linguistic Inquiry* 26: 615–633.

Lasnik, Howard. 1999a. On feature strength: Three minimalist approaches to overt movement. *Linguistic Inquiry* 30: 197–217.

Lasnik, Howard. 1999b. *Minimalist Analysis*. Oxford: Blackwell.

Lasnik, Howard. 2001. A note on the EPP. *Linguistic Inquiry* 32: 356–361.

Lasnik, Howard, and Mamoru Saito. 1984. On the nature of proper government. *Linguistic Inquiry* 15: 235–289.

Lasnik, Howard, and Mamoru Saito. 1992. *Move α: Conditions on Its Application and Output*. Cambridge, Mass.: MIT Press.

Lasnik, Howard, and Myung-Kwan Park. 2003. The EPP and the Subject Condition under sluicing. *Linguistic Inquiry* 34: 649–660.

Lema, José, and Maria-Luisa Rivero. 1990. Long head-movement: ECP vs. HMC. In Juli Carter, Rose-Marie Déchaine, Bill Philip, and Tim Sherer, eds., *NELS 20*, 337–347. Amherst, Mass.: GLSA, University of Massachusetts.

Lin, Jonah. 2005. Does *wh*-in-situ license parasitic gaps? *Linguistic Inquiry* 36: 298–302.

Lopez, Luis. 2000. Head of a projection. Unpublished manuscript, University of Illinois at Chicago.

Mahajan, Anoop. 2000. Eliminating head movement. *GLOW Newsletter* 44: 44–45.

Martin, Roger. 1999. Case, the Extended Projection Principle, and Minimalism. In Samuel D. Epstein and Norbert Hornstein, eds., *Working Minimalism*, 1–26. Cambridge, Mass.: MIT Press.

Martin, Roger, and Juan Uriagereka. 2000. Introduction: Some possible foundations of the Minimalist Program. In Roger Martin, David Michaels, and Juan Uriagereka, eds., *Step by Step*, 1–29. Cambridge, Mass.: MIT Press.

Matsubara, Fuminori. 2000. p*P Phases. *Linguistic Analysis* 30: 127–161.

Matushansky, Ora. 2006. Head movement in linguistic theory. *Linguistic Inquiry* 37: 69–109.

May, Robert. 1977. The grammar of quantification. Unpublished doctoral dissertation, MIT, Cambridge, Mass.

May, Robert. 1985. *Logical Form*. Cambridge, Mass.: MIT Press.

McCloskey, James. 2000. Quantifier floating and wh-movement in an Irish English. *Linguistic Inquiry* 31: 57–84.

McCloskey, James. 2002. Resumption, successive cyclicity, and the locality of operations. In Samuel D. Epstein and T. Daniel Seely, eds., *Derivation and Explanation in the Minimalist Program*, 184–226. Oxford: Blackwell.

Nunes, Jairo. 2001. Sideward movement. *Linguistic Inquiry* 32: 303–344.

O'Grady, William. 2005. *Syntactic Carpentry*. Mahwah, NJ: Erlbaum.

Pesetsky, David. 1982. Paths and categories. Unpublished doctoral dissertation, MIT, Cambridge, Mass.

Pesetsky, David. 1987. Wh-in-situ: Movement and unselective binding. In Eric Reuland and Alice ter Meulen, eds., *The Representation of (In)definiteness*, 98–129. Cambridge, Mass.: MIT Press.

Pesetsky, David. 1998. Some optimality principles of sentence pronunciation. In Pilar Barbosa, et al., eds., *Is the Best Good Enough?*, 337–384. Cambridge, Mass.: MIT Press.

Pesetsky, David. 2000. *Phrasal Movement and Its Kin*. Cambridge, Mass.: MIT Press.

Peters, Stanley, and Ronald Ritchie. 1973. On the generative power of transformational grammars. *Information Sciences* 6: 49–83.

Phillips, Colin. 1996. Order and constituency. Unpublished doctoral dissertation, MIT, Cambridge, Mass.

Phillips, Colin. 2003. Linear order and constituency. *Linguistic Inquiry* 34: 37–90.

Pollard, Carl, and Ivan Sag. 1992. Anaphors in English and the scope of binding theory. *Linguistic Inquiry* 23: 261–303.

Radford, Andrew. 1997. *Syntactic Theory and the Structure of English*. Cambridge: Cambridge University Press.

Radford, Andrew. 2004. *English Syntax*. Cambridge: Cambridge University Press.

Reinhart, Tanya. 1998. Wh-in-situ in the framework of the minimalist program. *Natural Language Semantics* 6: 29–56.

Reinhart, Tanya, and Eric Reuland. 1993. Reflexivity. *Linguistic Inquiry* 24: 657–720.

Richards, Norvin. 1997. What moves where when in which language? Unpublished doctoral dissertation, MIT, Cambridge, Mass.

Richards, Norvin. 1999. Featural cyclicity and the ordering of multiple specifiers. In Samuel D. Epstein and Norbert Hornstein, eds., *Working minimalism*. Cambridge, Mass.: MIT Press.

Richards, Norvin. 2001. *Movement in Language*. Oxford: Oxford University Press.

Richards, Norvin. 2004. Against bans on lowering. *Linguistic Inquiry* 35: 453–463.

Rizzi, Luigi. 1990. *Relativized Minimality*. Cambridge, Mass.: MIT Press.

Rudin, Catherine. 1988. On multiple questions and multiple wh-fronting. *Natural Language and Linguistic Theory* 6: 445–501.

Sasaki, Jun. 2000. A minimalist account of complementizer alternations. *Linguistic Analysis* 30: 162–176.

Sobin, Nicholas. 2002. The Comp-trace effect, the adverb effect and minimal CP. *Journal of Linguistics* 38: 527–560.

Soh, Hooi Ling. 2005. *Wh*-in-situ in Mandarin Chinese. *Linguistic Inquiry* 36: 143–155.

Sportiche, Dominique. 1988. A theory of floating quantifiers and its corollaries for constituent structure. *Linguistic Inquiry* 19: 425–450.

Stroik, Thomas. 1992. English wh-in-situ constructions. *Linguistic Analysis* 22: 133–153.

Stroik, Thomas. 1995. Some remarks on superiority effects. *Lingua* 95: 239–258.

Stroik, Thomas. 1996a. DP raising and inner island effects. *Linguistic Analysis* 26: 139–158.

Stroik, Thomas. 1996b. *Minimalism, Scope, and VP Structure*. Thousand Oaks, CA: Sage.

Stroik, Thomas. 1999. The SURVIVE Principle. *Linguistic Analysis* 29: 278–303.

Stroik, Thomas. 2000. *Syntactic Controversies*. Munich: Lincom Europa.

Stroik, Thomas. 2001. On the light verb hypothesis. *Linguistic Inquiry* 32: 362–369.

Stroik, Thomas, and Michael Putnam. 2005. The lexicon at the interfaces. Paper presented at the LASSO 2005 Conference, Lubbock, TX.

Travis, Lisa. 1984. Parameters and effects of word order variation. Unpublished doctoral dissertation, MIT, Cambridge, Mass.

Ura, Hiroyuki. 2000. *Checking Theory and Grammatical Functions in Universal Grammar*. Oxford: Oxford University Press.

Urban, Emily. 1999. *Exactly* stranding. Unpublished manuscript, University of California, Santa Cruz.

Uriagereka, Juan. 1999. Multiple Spell-Out. In Samuel D. Epstein and Norbert Hornstein, eds., *Working Minimalism*, 251–282. Cambridge, Mass.: MIT Press.

Watanabe, Akira. 2000. Feature copying and binding. *Syntax* 3: 159–181.

Wells, Rulon. 1947. Immediate Constituents. *Language* 23: 81–117.

Woolford, Ellen. 1999. More on the anaphor agreement effect. *Linguistic Inquiry* 30: 257–287.

Zubizarreta, Maria Luisa. 1998. *Prosody, Focus, and Word Order*. Cambridge, Mass.: MIT Press.

Zwart, Jan-Wouter. 1997. *Morphosyntax of Verb Agreement*. Dordrecht: Kluwer.

Zwart, Jan-Wouter. 2001. Syntactic and phonological verb movement. *Syntax* 4: 34–62.

Zwart, Jan-Wouter. 2002. Issues relating to a derivational theory of Binding. In Samuel D. Epstein and T. Daniel Seely, eds., *Derivation and Explanation in the Minimalist Program*, 269–304. Oxford: Blackwell.

Index

Linguistic Inquiry Monographs

Samuel Jay Keyser, general editor

1. *Word Formation in Generative Grammar*, Mark Aronoff
2. *X Syntax: A Study of Phrase Structure*, Ray S. Jackendoff
3. *Recent Transformational Studies in European Languages*, Samuel J. Keyser, Ed.
4. *Studies in Abstract Phonology*, Edmund Gussmann
5. *An Encyclopedia of AUX: A Study in Cross-Linguistic Equivalence*, Susan Steele
6. *Some Concepts and Consequences of the Theory of Government and Binding*, Noam Chomsky
7. *The Syntax of Words*, Elisabeth O. Selkirk
8. *Syllable Structure and Stress in Spanish: A Nonlinear Analysis*, James W. Harris
9. *CV Phonology: A Generative Theory of the Syllable*, George N. Clements and Samuel Jay Keyser
10. *On the Nature of Grammatical Relations*, Alec P. Marantz
11. *A Grammar of Anaphora*, Joseph Aoun
12. *Logical Form: Its Structure and Derivation*, Robert May
13. *Barriers*, Noam Chomsky
14. *On the Definition of Word*, Anna-Maria Di Sciullo and Edwin Williams
15. *Japanese Tone Structure*, Janet Pierrehumbert and Mary E. Beckman
16. *Relativized Minimality*, Luigi Rizzi
17. *Types of Ā-Dependencies*, Guglielmo Cinque
18. *Argument Structure*, Jane Grimshaw
19. *Locality: A Theory and Some of Its Empirical Consequences*, Maria Rita Manzini
20. *Indefinites*, Molly Diesing
21. *Syntax of Scope*, Joseph Aoun and Yen-hui Audrey Li
22. *Morphology by Itself: Stems and Inflectional Classes*, Mark Aronoff
23. *Thematic Structure in Syntax*, Edwin Williams
24. *Indices and Identity*, Robert Fiengo and Robert May
25. *The Antisymmetry of Syntax*, Richard S. Kayne
26. *Unaccusativity: At the Syntax-Lexical Semantics Interface*, Beth Levin and Malka Rappaport Hovav
27. *Lexico-Logical Form: A Radically Minimalist Theory*, Michael Brody
28. *The Architecture of the Language Faculty*, Ray Jackendoff
29. *Local Economy*, Chris Collins
30. *Surface Structure and Interpretation*, Mark Steedman
31. *Elementary Operations and Optimal Derivations*, Hisatsugu Kitahara
32. *The Syntax of Nonfinite Complementation: An Economy Approach*, Željko Bošković
33. *Prosody, Focus, and Word Order*, Maria Luisa Zubizarreta
34. *The Dependencies of Objects*, Esther Torrego
35. *Economy and Semantic Interpretation*, Danny Fox

36. *What Counts: Focus and Quantification*, Elena Herburger
37. *Phrasal Movement and Its Kin*, David Pesetsky
38. *Dynamic Antisymmetry*, Andrea Moro
39. *Prolegomenon to a Theory of Argument Structure*, Ken Hale and Samuel Jay Keyser
40. *Essays on the Representational and Derivational Nature of Grammar: The Diversity of Wh-Constructions*, Joseph Aoun and Yen-hui Audrey Li
41. *Japanese Morphophonemics: Markedness and Word Structure*, Junko Ito and Armin Mester
42. *Restriction and Saturation*, Sandra Chung and William A. Ladusaw
43. *Linearization of Chains and Sideward Movement*, Jairo Nunes.
44. *The Syntax of (In) dependence*, Ken Safir
45. *Interface Strategies: Optimal and Costly Computations*, Tanya Reinhart
46. *Asymmetry in Morphology*, Anna Maria Di Sciullo
47. *Relators and Linkers: The Syntax of Predication, Predicate Inversion, and Copulas*, Marcel den Dikken
48. *On the Syntactic Composition of Manner and Motion*, Maria Luisa Zubizarreta and Eunjeong Oh
49. *Introducing Arguments*, Liina Pylkkänen
50. *Where Does Binding Theory Apply?*, David Lebeaux
51. *Locality in Minimalist Syntax*, Thomas S. Stroik