Root, Successive-Cyclic and Feature-Splitting Internal Merge: Implications for Feature-Inheritance and Transfer

by

Miki Obata

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Linguistics)
in The University of Michigan
2010

Doctoral Committee:

Professor Samuel D. Epstein, Co-Chair
Professor Noam Chomsky, Co-Chair, Massachusetts Institute of Technology
Professor Richard L. Lewis
Associate Professor Acrisio M. Pires
Professor Hisatsugu Kitahara, Keio University
Professor T. Daniel Seely, Eastern Michigan University

Acknowledgments

It is finally time to express my gratitude to all the people who have helped with my training in linguistics. First and foremost, I am very grateful to my thesis committee members for their constructive comments and suggestions on every aspect of my thesis.

Sam Epstein, a co-chair of my thesis committee and my Ph.D. advisor, has always been one of my best friends since I first came to Michigan. If Sam had not been my advisor, I could not have reached the stage where I am able to write this acknowledgment section in my thesis. He has selflessly given of his time and labor to every aspect of my graduate life. Especially whenever I gave a presentation at a conference or had a job interview, he always sent me messages by e-mail for encouragement, which gave me a lot of confidence and made me feel that I always have his support. Also, it has been a real pleasure for me to co-author several papers with him. All the work and discussions we have had so far remain my precious memories forever. Although Sam is my ideal teacher and did everything he can do for me, one more thing I now hope for him is that he visits Japan in the near future overcoming airplane phobia! I hope it comes true someday!

It has been a great pleasure for me to have Noam Chomsky on my committee as a co-chair. Since Sam first suggested to me almost three years ago that I sent papers I had written to Noam, we have exchanged e-mails several times. I remember well that Noam welcomed me with a smile when I was very nervous at our first meeting. It was the most challenging job for me to convince him of the soundness of what I am proposing.

Whenever I tried to do so, he showed me other possibilities I have never thought of and also asked me about data I had not considered. One day, he even gave me topics for my second and third dissertations! All the discussions we had made evident his extremely profound knowledge, and I will treasure forever the opportunity to work with him!

Rick Lewis has given comments and suggestions on my thesis from his point of view as a psychologist, which was sometimes the hardest for me! Also, Rick gave me a chance to work on a psycholinguistics project, which expands my horizons as a linguist. By taking his classes and having discussions with him, I came to have more interests in how knowledge of language is used as well as in what constitutes that knowledge. He enabled me to have a bigger picture, namely to imagine where, how, and what I am working on is placed in the entire cognitive system.

Acrisio Pires has always given me detailed comments not only on my thesis but also on other papers I have written so far. Whenever I asked him questions, he seriously answered them even in the kitchen in the department! Also, Acrisio (along with Sam) helped me a lot when I had difficulties with some coursework in my first year. I would like to support my future students as he did for me.

I was lucky enough to spend two years in Michigan with Hisatsugu Kitahara as a visiting scholar in our department. Hisa and I were the only members of the department from Japan! When I decided to work with Sam in Michigan four years ago, some of the people around me said, “Aim for Kitahara!” I am not sure if I made it but I am pretty sure I have never met with a person who knows Noam’s work as much as Hisa (except Noam!). His comments have all been constructive and always encouraged me.

Finally but not least, Daniel Seely has been another good supporter in syntax for me. Daniel has always tried to expand my ideas by finding their positive aspects and encouraging me to work on whatever interests me. I remember well that he read almost every version of my paper with Sam and sent us comments every time. His sincere attitude to research told me how researchers should be. Also, I will not forget we had a fun time in Tennessee and the Basque country for conferences!

My thesis committee members aside, I also owe a great deal to the Department of Linguistics at the University of Michigan. I thank especially to Steve Abney, Marlyse Baptista, Pam Beddor, Julie Boland, Andries Coetzee, San Duanmu, Debby Keller-Cohen, Ezra Keshet, Natalia Kondrashova, Elaine McNulty, Carmel O'Shannessy, Robin Queen, Sally Thomason, Kristen McLeod, Sandie Pettie, Lori Scott and Sylvia Suttor for their teaching, advice, comments and general support. My thanks also go out to fellow students/scholars at the University of Michigan: Miyeon Ahn, Erica Beck, Christopher Becker, Eric Brown, Tim Chou, Gerardo Fernandez-Salgueiro, Catherine Fortin, Yufen Hsieh, Dina Kapetangianni, Donggeol Lee, Susan Lin, Kevin McGowan, David Medeiros, Christopher Odato, Hamid Ouali, Christopher Schad, Lauren Squires, Joseph Tyler, Damon Tutunjian, Jonathan Yip and Xinting Zhang. I thank especially to David Medeiros for his proofreading many of my papers including the dissertation. I wish to also thank my friends in the Japanese language program at the University of Michigan for their generous help: Shoko Emori, Kenji Endo, Yoshiro Hanai, Eiko Hirakawa, Satoru Ishikawa, Shuyun Ju, Junko Kondo, Yoshihiro Mochizuki, Mayumi Oka, Fumie Okudera, Shinya Suzuki, Satoko Tsuda-Petty, Nami Ujihara and Shoko Watarai.

I wish to express my appreciation to my colleagues at other institutions for their unsurpassed hospitality: Masahiko Aihara, Cedric Boeckx, Željko Bošković, Vicki

Carstens, Barbara Citko, Christopher Collins, Marcel den Dikken, Betty McNulty Epstein, Tomohiro Fujii, Yukio Furukawa, Ángel Gallego, Takuya Goro, John Hale, Norbert Hornstein, Takaomi Kato, Maki Kishida, Howard Lasnik, Hironobu Kasai, Masakazu Kuno, James McCloskey, Ai Matsui, Shigeru Miyagawa, Yoichi Miyamoto, Chigusa Morita, Chizuru Nakao, Andrew Nevins, Jonathan Nissenbaum, Jairo Nunes, Akira Omaki, Hajime Ono, David Pesetsky, Milan Rezac, Norvin Richards, Hiromu Sakai, Tetsuya Sano, Junko Shimoyama, Koji Sugisaki, Mina Sugimura, Lisa Travis and Masaya Yoshida.

As for my linguistics training in Japan, I thank the following professors and colleagues for their constant support: Kinya Fukawa, Kinsuke Hasegawa, Tomohiko Ishimori, Nobuhiro Kaga, Akio Kamio, Haruko Kono, Kazuma Matsubara, Edson T. Miyamoto, Manabu Mizuguchi, Minoru Nakau, Megumi Nishimura, Kaori Sato, Koichi Takezawa, Miyoko Yasui and Noriaki Yusa. I especially thank Koichi Takezawa for his encouragement and generous support to pursue a Ph.D. in the United States.

Lastly but not least, I wish to thank my family members for their invaluable help and financial support.

Table of Contents

List of Abbreviation

Types of Phrases

CP	Complementizer phrase
TP	Tense phrase
vP	Light verb phrase
VP	Verb phrase
DP	Determiner phrase
NP	Noun phrase

Types of Morphemes

Nom	Nominative Case
Acc	Accusative Case
Gen	Genitive Case
Dat	Dative Case
Part	Particle
C	Complementizer (morpheme)
Past	Past tense
Prog	Progressive

Types of Features

[iF]	Interpretable feature (at the CI interface)
[uF]	Uninterpretable feature (at the CI interface)
Valued [uF]	Valued uninterpretable feature
Unvalued [uF]	Unvalued uninterpretable feature
[Case]	Case feature
[Dec]	Declarative feature
[Op]	Operator feature
[Phi]	Phi feature
[Phon]	Phonetic feature
[Q]	Question feature

Others

NS	Narrow syntax
PHON	Phonetic component
SEM	Semantic component
CI	Conseptual-Intentional interface
SM	Sensory-Motor interface

Chapter 1

Goals and Organization

1.1 Goals of the Thesis

The cognitive capacity for language is unique to human beings and distinguishes us from other animals. All human beings can acquire at least one particular language by means of this biologically-determined faculty of language unless they have certain impairments. When we know a language, however, what exactly do we know? Behind production of a simple sentence such as "Sam likes dogs", each part of the language faculty computes e.g. for selecting words from the mental lexicon (i.e. morphology), for combing those words to generate a sentence (i.e. syntax), for assigning appropriate meanings (i.e. semantics) and for pronouncing the resulting sentence (i.e. phonology/phonetics) etc... By these mental computations being carried out in our brain, we finally produce the grammatical sentence. That is, explaining what linguistic computations take place in our brain leads to clarifying aspects of the (biologically-determined) human cognitive system that performs these computations. In this sense, linguistics can be regarded as a cognitive science and this is exactly what this thesis pursues. The goal of the dissertation is to determine aspects of the structure of the human language faculty, a cognitive system, specifically focusing on human <u>syntactic</u> systems, and then interaction with phonological and semantic interpretation which enable us to creatively produce an unlimited number of grammatical sentences.

The main focus is on the following two points: [1] (re-)formulating fundamental mechanisms of the computational system especially concerning the operation *Merge*, which assembles two elements (i.e. each a bundle of features) creating a new syntactic object, a set, and [2] addressing certain central issues of how syntactic representations (constructed by Merge) are (cyclically) transferred from the narrow syntax to the interface components, in which the semantic and phonological computations are carried out.

The thesis is mainly devoted to [1] and [2] within the framework of the phase-based derivational approach suggested in Chomsky (2000, 2001, 2004, 2007, 2008), where the syntactic computation is 'chunked' i.e. limited to small units called *phases*, by assumption, leading to the reduction of computational complexity. With respect to [1], I closely examine cases in which multiple heads are involved in triggering Internal Merge (i.e. movement) of a single element under the feature-inheritance system, introduced in Chomsky (2007, 2008). This system states that phi-features on T and on V are inherited from the phase heads v and C, respectively. Regarding [2], the mechanism of Transfer is discussed, especially focusing on the timing of its application and also seeking to explain which portion (category) within the constructed narrow syntax representation it applies to and what the narrow syntax, PHON and SEM representations are that result from Transfer.

Throughout the thesis, the central aims are to develop supporting argument for Chomsky's feature-inheritance system and to identify its theoretical/empirical consequences and implications regarding new phenomena and unexplored domains of inquiry. Also, the thesis will discuss how the narrow syntax is constrained by the interface conditions. I believe that the attempt made in this thesis if successful will throw new light on basic mechanisms of the syntactic computational system, a core aspect of the human language faculty. Section 1.2 summarizes what each chapter discusses and how each chapter contributes to accomplishing the main goals.

1.2 The Chapters

1.2.1 Chapter 2: Theoretical Background

The thesis is composed of seven chapters. Following Chapter 1 Goals and Organization, Chapter 2 provides the relevant theoretical background and also presents a selective historical review centering on the shift from the Principles and Parameters representational approach to the "Minimalist" derivational approach. Chomsky's phase-based derivational approach is summarized and some issues the thesis takes up for discussion are indicated.

1.2.2 Chapter 3: What Moves Where?: Simultaneous Attraction by Multiple Heads

One of the fundamental properties of the phase-based derivational approach is that it strictly limits the computational space.[1] By doing so, the derivational approach succeeds in reducing computational complexity. Accordingly, [A] how computation proceeds in such a limited domain, [B] also how those limited representations are sent to the interfaces, and [C] what is left behind by Transfer in narrow syntax, [D] then interpreted at the interfaces and [E] re-assembled into a complete "sentential" representation become our central concerns. Chapter 3 is devoted to the issue of Internal Merge in such a context,

[1] Aspects of this chapter are based on Obata (2008a) and Obata and Epstein (2008, in press, to appear).

especially focusing on cases where multiple heads (e.g. T and C) are involved in attracting a single element as a consequence of the feature inheritance system suggested in Chomsky (2007, 2008).

According to Richards (2007), valued [uF] cannot be stranded on the edge after Transfer in a convergent derivation. This is because [uF] valued on the edge of the lower phase is no longer distinguishable from [iF] in the next higher phasal computation. Consequently, Transfer fails to delete valued [uF] on an edge, causing crash (= Richards-Chomsky's value-Transfer simultaneity issue). In other words, phase heads cannot bear and retain [uF] which become syntactically valued, given that Transfer applies only to the domain of a phase and leaves phase-edge which becomes part of the next higher phase. If [uF] is introduced to the derivation on a phase head as in Chomsky's feature-inheritance system, therefore, [uF] has to be discharged off the edge and into the phasal domain. Although Richards mentions only phase heads, this chapter shows that the logic of the Value-Transfer Simultaneity analysis in fact applies not only to phase heads but also to a moving element (e.g. wh-phrases) which lands at a phase-edge position. Such moved DPs are also stranded on the edge after Transfer. A moving element such as an argument wh-phrase bears [uCase], which is valued as a consequence of phi-agreement with T or V. Then, the wh-phrase bearing valued [uCase] undergoes Internal Merge to a phase edge. But then the Chomsky-Richards issue in face predicts that the appearance of this valued [uCase] at the phase edge causes crash, at the next higher phase.

To avoid this unwelcome result, here, I propose Feature-Splitting Internal Merge, where features on a single element are split into two landing sites, which enables valued [uCase] not to appear at a phase edge, but rather it is split off to a non-edge landing site inside the phasal domain. One of the direct consequences obtained from the proposed mechanism is to explain improper movement phenomena, which has been a long-standing problem since Chomsky (1973) discovered it. Under the feature-splitting system, improper movement is ruled out by causing featural crash. In addition, this system implies a new way to define two types of syntactic positions--A/A'-positions. Also, this chapter extends the proposed system to Bantu languages specifically Kilega and Lusaamia in which improper movement is permissible as suggested by Carstens (2005). In addition, Japanese hyper-raising and raising to object and also English *tough*-constructions are examined as "grammatical" improper movement examples.

This chapter explores these issues to clarify some properties of the Merge operation and also lends empirical support to the feature-inheritance system, which makes simultaneous attraction possible.

1.2.3 Chapter 4: Problems Confronting Root-Clause Transfer: "Solutions" and Predictions

Chapter 4 further considers Richards and Chomsky's value-transfer simultaneity addressing mechanisms governing root transformations.[2] As summarized in the last section, the phasal Transfer system and the feature-valuation mechanism both lead us to the conclusion that [uF] on a phase head (occupying the edge) has to be discharged to a lower head inside the phase head complement. That is, valued [uF] can never remain on the edge after Transfer in a convergent derivation given the current Transfer system--only the domain of a phase is sent to the interfaces stranding the edge which is always transferred as a part of the next higher phase. But what happens in root clauses? The

[2] Aspects of this chapter are based on Obata (2009).

matrix phase-edge is never included in the domain of any phase. In a sentence like "what did you buy?", for instance, how does the portion of "what did" undergo Transfer? With respect to this issue, I advance Root Transfer Hypothesis (to be revised in Chapter 5): Unlike all other phases, the highest phase edge is transferred "for free" at the same time as its domain is transferred. That is, under this hypothesis, the root edge is transferred along with its domain, so that every representation can be sent to the interfaces without applying an additional Transfer only to the stranded matrix edge.

The proposed Root Transfer Hypothesis makes an interesting prediction about Richards and Chomsky's value-transfer simultaneity which implies that a phase-edge cannot keep a syntactically valued [uF]. Since the root edge is now transferred along with its domain, it is predicted that the matrix phase head unlike any other phase heads can bear valued [uF] and no crash arises because [iF] and valued [uF] should be still distinguishable becase there is no higher phase. In other words, the root phase-edge is distinct from other phase-edges in that the former, but not the latter, can keep syntactically valued [uF] without discharging it to lower heads within the phase head complement. In this chapter, I suggest that [uQ] on the matrix C becomes the driving force of root transformations (cf. Emonds, 1976) such as subject-auxiliary inversion in English direct questions. Also, I show that the proposed system can explain differences between Japanese long-distance scrambling and Japanese postposing. They are distinguished in terms of possible pronunciation sites of copies. In Japanese long-scrambling, scrambled phrases can be pronounced at any single phase edge. In Japanese postposing, on the other hand, moved phrases can be pronounced only at the matrix edge. I demonstrate how the system developed in this chapter explains their different behaviors and also show some theoretical and empirical implications obtained from the proposed analysis.

By considering Transfer of the root edge, this chapter further pursues Richards and Chomsky's value-transfer simultaneity issues, which leads to clarifying some more aspects of (root) Internal Merge and the nature of the Transfer system.

1.2.4 Chapter 5: A Generalized Theory of Feature Inheritance: How does Transfer Send What?

Chapter 5 pursues a more fundamental question relating to the previous two chapters, namely "Why does regular, non-root Transfer apply (only) to the phasal domain?"[3]

Recall, Chapter 3 examines what happens if multiple heads attract a single element. This simultaneous attraction is a natural consequence induced from Richards-Chomsky's value-transfer simultaneity, where syntactically valued [uF] cannot be stranded at (non-root) phase-edge positions because of the delayed application of Transfer. The Feature-Splitting system is an optimal way to avoid placing syntactically valued [uF] at a phase edge. Then, Chapter 4 addresses the issue of transferring the root edge, which is never included in the domain of any (higher) phase. To transfer the root edge appropriately, I stipulated that the root edge is transferred along with its domain. This enabled us to account for the fact that all root edges do (of course) undergo interface interpretation while also allowing us to explain some unique properties root clauses have, "root transformations", because valued [uF] can appear at the root edge which is transferred along with its domain without any delay. In these previous two chapters, it has been presumed that Transfer applies to the domain of a phase. But, why does it do so?

[3] Aspects of this chapter are based on Obata (2010).

That is, the mode-of-application of the fundamental operation Transfer is still unexplained: Why does Transfer apply to the domain of a phase? This is the issue I tackle with in this chapter.

The reason why it has been assumed that only the phase head complement is transferred is that edge positions work as “escape hatches” of successive cyclic movement, so that those positions need to be available for operations in the next higher phase. However, the following two questions arise: <i> "Is there any evidence that edge positions are left behind after Transfer when successive movement does not apply?" and <ii> "Is there any way to deduce the pure (empirically motivated) stipulation that Transfer is limited to a phase head complement?" In order to answer those questions, it seems to be reasonable to pursue another way of defining the Transfer system. In this chapter, I reformulate the Transfer system as follows: Within a single phase, Transfer can send any XP (unless other conditions exclude the result of such Transfer). Under the reformulated definition, the transferred domain is neither stipulated nor limited to the phase head complement but rather any kind of XP can be transferred. Also, I show that the reformulated version is preferable to the current definition in terms of the economy condition and enables us to abolish the Phase Impenetrability Condition in Chomsky (2000). In addition, the Root Transfer Hypothesis advanced in Chapter 4 can be also subsumed into the reformulated Transfer system, so that there is no need to stipulate a special kind of Transfer applying uniquely to the root edge.

Moreover, the chapter further pursues the nature of the Transfer operation. The questions to be tackled are (i) what is left behind in the narrow syntax representation after Transfer? and (ii) how are transferred pieces re-assembled for global computation at the interfaces (when the "entire sentence" is assembled)? For the question (i), I discuss the two logical possibilities: something is left vs. nothing is left. Then, I demonstrate that the latter seems to be more preferable in the system and also propose the Label-Copying system: The transferred phrase leaves only a copy of its label when it undergoes Transfer but none of its internal syntactic structure. I suggest that this mechanism gains theoretical motivation from the no-tampering condition suggested in Chomsky (2007, 2008) and also from the recoverability condition as discussed in Chomsky and Lasnik (1995). Also, it captures PIC effects since terms are gone. In addition, the copied label left behind by Transfer in narrow syntax works as a "guide" for reassembling each of the transferred pieces into a single interface representation, which is necessary for computing e.g. Condition C (SEM) and e.g. sentential intonation (PHON). Furthermore, I also reveal a potentially serious and widespread problem confronting phase-based derivations, namely how it is ever possible to move phases that are bigger than a phase head complement e.g. CP fronting as in "That John went to the store, I believe" and demonstrate that the proposed label-copy system plays an important role in moving a phrase bigger than phases, which lends empirical support for the proposed system.

This chapter elucidates basic properties of Transfer and also some aspects of the mechanism to reconstitute transferred pieces into a single representation. By getting rid of the stipulation that only the phase head complement is transferred, the proposed system can obtain several theoretical and empirical advantages.

1.2.5 Chapter 6: Thesis Summary and Architectural Implications

Chapter 6 discusses some of the fundamental issues concerning the current phase-based approach, especially comparing with Frampton and Gutmann's (2000) type of Crash-Proof syntax. Also, the chapter provides a brief summary of the thesis and concludes the study.

Chapter 2

Theoretical Background

2.1 Introduction

This chapter presents brief historical reviews of theoretical developments in generative grammar, originating in Chomsky (1957), which the thesis is specifically based on. The chapter especially focuses on the shift from the Principles and Parameters (P&P) approach to the minimalist program. What are differences between the P&P approach and minimalism? What are the goals of minimalism? Why is spirits of the minimalism important to push forward as a scientific approach to language? In this thesis, the arguments are developed within the framework of the minimalist program, so that discussing these points clarifies what the thesis specifically aims for and how the thesis contributes to the ultimate goal, namely determining aspects of the structure of the human language faculty, a biologically endowed cognitive system.

The chapter is organized as follows: Section 2.2 first summarizes the P&P approach as a departure point. Section 2.3 considers the shift to the minimalist program by discussing the changes it introduces, how exactly the changes are reflected in hypotheses concerning mechanisms of the language faculty and where motivations for the shift come from. Section 2.4 introduces some key concepts in the phase-based derivational approach, which plays an important role throughout the dissertation. Section 2.5 presents the Feature-Inheritance system proposed in Chomsky (2007, 2008), which will be given close consideration throughout the thesis. Also, the section addresses how the existence and obligatory application of the feature-inheritance system is deduced by Richards (2007) from independently motivated aspects of the timing and nature of the Transfer operation. Section 2.6 provides a summary of this chapter.

2.2 On the P&P Approach: The Filter-Based System

This section reviews the essence of the P&P approach as a departure point to clarify what the minimalist perspective is, why it is important and why the thesis adopts the minimalist framework. Two important concepts in the generative tradition are descriptive adequacy and explanatory adequacy discussed in Chomsky (1964).

"… a grammar that aims for descriptive adequacy is concerned to give a correct account of the linguistic intuition of the native speaker; in other words, it is concerned with the output of the device (= the device being UG, MO); a linguistic theory that aims for explanatory adequacy is concerned with the internal structure of the device (= the devise being UG, MO); that is, it aims to provide a principled basis independent of any particular language, for the selection of the descriptively adequate grammar of each language."

(Chomsky 1964: 63)

To describe how passives are generated in English, for example, Chomsky (1957) advanced the following Passive transformation rule:

(1) *Passive*
Structural Analysis: NP - Aux - V - NP
Structural Change: $X_1 - X_2 - X_3 - X_4 \rightarrow X_4 - X_2 + \text{be} + \text{en} - X_3 - \text{by} + X_1$

(Chomsky 1957: 112)

Since native speakers of English are equipped with the passive rule, they can generate passive sentences based on the rule. That is, (1) describes native speakers' linguistic intuition, so that a grammar containing (1) attains descriptive adequacy. Chomsky and Lasnik (1995) also write:

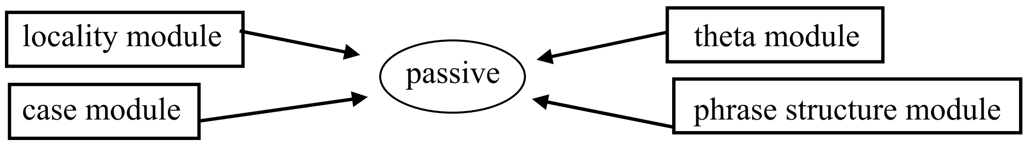
"In the early stages of generative grammar, the task was to find a rule system of the permitted form from which these phenomena (and infinitely many others) could be derived".

(Chomsky and Lasnik 1995: 28)

Assuming that (1) is a correct description of knowledge of passive (i.e. (1) is a part of the adult "English" grammar), then we ask a deeper question: How do children acquire a syntactic rule such as (1)? The second concept explanatory adequacy can be accomplished by answering this question, in other words, by deriving rule systems like (1) from general UG principles. This is what the P&P approach has tackled. In this sense, the P&P approach had a major impact on grammatical theories which had been previously postulated. "Constructions" had been the units of transformational analysis as exemplified by the Passive rule (1). But under the P&P approach, constructions no longer exist (i.e. have no theoretical status) and, for example, passives are by-products of the interaction of UG principles.

Now let us consider how the passive rule in (1) can be derived from UG principles. The first step is to ask the far more general question, where do NPs occur? Based on Vergnaud's (1977) Case filter stating that every NP must be assigned Case, NPs have to appear in Case-assigned positions by S-structure. (See also Chomsky and Lasnik 1977) That is, passive subjects first appear as the objects of accusative verbs. It is in this position that theta roles are assigned. Once those verbs get a passive morpheme "-en", they lose the ability to assign accusative Case. Therefore, passive subjects need to undergo movement to the Case-assigning positions (i.e. Spec-TP) assuming Case Filter ar S-structure, not at D-structure, otherwise the Case filter marks the representation with *. That is, now we can say that passive is a by-product of several UG principles. The following diagram is cited from Boeckx (2006), which provides an insightful review regarding this issue:

(2)



(Boeckx 2006: 65)

Passives are generated by (at least) these four modules/UG principles interacting. The theta module and Case module constrain appearance of NPs, which belong to structures constructed by the phrase structure module. Also, NP-movement to Spec-TP is restricted by the locality module. As a consequence of this collaboration, passive sentences can be generated without appeal to a construction-specific rule like (1).

Chomsky and Lasnik (1995) suggest that UG principles (i.e. I-language-invariant) are of two types: one applies to derivations as exemplified in the phrase structure module in (2) and the other applies to representations resulting from derivations such as the Case filter. These principles form modules and each of them is equipped with parametric values which are set differently among particular I-languages. Under the P&P system, language acquisition is characterized as parameter-setting based on primary linguistic data. That is, by choosing appropriate values established in principles, children come to generate, e.g. passive sentences as a consequence of the interactions of the principles. Construction-specific rules are now deduced from the UG principles, with the system attaining higher level explanation. This is one of the radical changes from pre-P&P approaches.

Another aspect which is worth mentioning is that the P&P approach is a "rule-free" system in contrast to the pre-P&P rule-based system. Chomsky and Lasnik (1995) write:

"Much of the most fruitful inquiry into generative grammar in the past years has pursued the working hypothesis that UG is a simple and elegant theory, with fundamental principles that have an intuitive character and broad generality. By dissolving the notion of construction and moving toward "rule-free" systems, the P&P approach carries this tendency considerably forward."

(Chomsky and Lasnik 1995: 29)

As discussed in the above paragraphs, the P&P approach recaptures traditional construction-specific rules by deducing them from general UG principles (i.e. filters), so that the P&P approach (i.e. the rule-free system) can be recognized as a more elegant and simple theory than a pre-P&P theory.

As Chomsky (1986a) mentions, on the other hand, the P&P approach is a "virtually" rule-free system. Chomsky (1986a) says:

"The representations that appear at the various levels are those that can be projected from semantic properties of lexical items in such a way as to accord with various principles of UG with their parameters set."

(Chomsky 1986a: 93)

Even in the P&P approach, the derivations, not only the representations, are somehow constrained by UG principles, so that movement operations, for example, are applied by S-structure, which are evidenced by the existence of traces and/or chains, as Epstein and Seely (2002) points out. In this sense, it seems to be reasonable to say that the P&P approach is a "more" rule-free system than pre-P&P.

This section has reviewed how the P&P approach developed from pre-P&P and what the approach attained.

2.3 The Shift from P&P to the Minimalist Program

This section discusses how P&P develops to Minimalism, the framework assumed in this thesis. What progress was made in Minimalism? Why is pursuing the Minimalist spirit important?

2.3.1 Why is Fewer Better?: Toward Higher Levels of Explanations

For the P&P approach to advance to a higher explanatory level, it appears to be necessary (as in any scientific inquiry) to keep asking "why?". Chomsky (2004) writes:

"… we can seek a level of explanation deeper than explanatory adequacy, asking not only *what* the properties of language are but also *why* they are that way."
(Chomsky 2004: 105)

That is, certain outcomes successfully obtained from the P&P approach are not the final goal but are only steps to make progress further than explanatory adequacy. Looking for answers to the *why*-questions is the objective the minimalist program attempts to attain.

One might wonder why it has to be "minimalist", not e.g. "maximalist"? Weinberg (2001) presents a nice answer key to this question:

"There are arrows of scientific explanation, which thread through the space of all scientific generalizations. Having discovered many of these arrows, we can now look at the pattern that has emerged, and we notice a remarkable thing: … These arrows seem to converge to a common source! Start anywhere in science, and like an unpleasant child, keep asking 'Why?' You will eventually go down to the level of the very small."
(Weinberg 2001:17-18 cited from Boeckx 2006: 15)

The *why*-question and "minimalist" spirit are not separable but the latter is a consequence of the former. That is, aiming for a higher level of explanation automatically leads us to a minimal world. This is why exactly pursuing the minimalist perspective marks a pivotal point in the development of a linguistic theory.

This way of thinking is not limited to linguistics but rather follows all scientific inquiry. Freidin and Vergnaud (2001) note that:

"… one can define Galilean science as the search for mathematical patterns in nature. … A significant feature of the Generative Revolution in linguistics has been the development of a Galilean style in that field. And, to a great extent, the recent developments within MP must be viewed in this light…".

(Freidin and Vergnaud 2001: 647)

Galileo considers that nature is simple and perfect. Weinberg (1976) expresses the Galilean style of research as follows:

"… we have all been making abstract mathematical models of the universe to which at least the physicists give a higher reality than they accord the ordinary world of sensation."

(Weiberg 1976 cited from Freidin and Vergnaud 2001: 647)

Language is a part of nature, so that it is not surprising that the same style of research is applied to linguistics. More concretely, how do we pursue the Galilean style of science in linguistic research? Freidin and Vergnaud (2001) state that the minimalist program practices Dirac's (1968) mathematical procedure within linguistics, writing:

"Dirac has identified two main methods within the mathematical procedure itself: one is to remove inconsistencies, the other, to unite theories that were previously disjoint… In linguistics, the inconsistencies primarily concern overlapping grammatical conditions, … which conflict with the basic assumption that C_{HL} has an optimal design… One aspect of Dirac's mathematical procedure as applied in linguistics involves the effort to extend and deepen the mathematical formalism used to express syntactic concepts and syntactic principles."

(Freidin and Vergnaud 2001: 647)

By unifying previously disjoint assumptions into a more general principle, that is, linguistic theories can go toward higher generality and via the postulation of increasingly simple substantives. One might object to minimalism by saying that it is too minimal to cover all of the empirical facts, which have been covered before by earlier theories. As long as generative grammar is an empirical science, of course, the theories need to be empirically grounded. Even if there are empirical data which cannot be explained in current minimalism, however, it is too early to conclude that this research program is wrong. Epstein and Seely (2002) say:

"… empirical 'coverage' (though obviously of great importance) is not the sole issue, rather how the theory 'covers' the data, always a very difficult question to answer, must be posed, discussed, and addressed."

(Epstein and Seely 2002: 3)

As Galileo did not give up his style of research even if he could not explain why objects do not fly off the earth's surface as Chomsky (1980) notes, it is important to emphasize that pushing forward the minimalist program itself is worthwhile.

In the next section, more detailed discussion is developed with respect to how exactly minimalism has higher generality than the P&P approach focusing on Case Filter issues.

2.3.2 From Modularity to Higher Generality

The Case filter (i.e. Case module) sorts out a representations including NPs without Case into ungrammatical in the P&P approach as discussed in Section 2.2. This makes the P&P approach proceed to a rule-free system, which is a significant progress from pre-P&P. Now for minimalism, it is necessary to identify overlapping grammatical conditions and to unify them to achieve higher generality. Freidin and Vergnaud (2001) provide a nice review of this issue and say that the Case filter and the Principle of Fill Interpretation, which requires that every element of PF and LF must receive an appropriate interpretation (cf. Chomsky 1986a), are overlapping, so that the former can be subsumed by the latter. The Case filter is a mere stipulation to exclude NPs without Case and there is no reason *why* NPs without Case is not permissible in grammar. That is, there is no answer to this *why*-question.

In the P&P approach, whether NPs "get" structural Case from T/V is a matter of concern. If NPs do not receive Case, the Case Filter excludes the representation. If we see

Case-features from a bit different point of view, there seem to be different consequences. Freidin and Vergnaud (2001) write:

"From the point of view of economy, Case features are extraneous to interpretation at the LF interface at least and therefore should be eliminated from the derivation before LF."

(Freidin and Vergnaud 2001: 642)

In other words, if Case features are intrinsically uninterpretable at the LF interface, the representations including NPs bearing those features are excluded by the Principle of Full Interpretation. Now, NPs are equipped with Case-features when they are introduced into the derivations and need to "lose" them to get interpreted at the interface. For example, in Chomsky (2001):

"For the Case/agreement systems, the uninterpretable features are phi-features of the probe and structural Case of the goal N… Structural Case is not a feature of the probes (T, v), but it is assigned a value under agreement, then removed by Spell-Out from the narrow syntax."

(Chomsky 2001: 6)

In the narrow syntax derivation, [uCase] on NP/DP needs to be valued by an appropriate head to specify its phonetic form, e.g. "he" or "him". But once valuation is done, [uF] now valued (i.e. [uCase] in this case) needs to be removed by Spell-Out (= Transfer in Chomsky 2004 and later), otherwise the derivation crashes because of a violation of the Full Interpretation Principle. The same scenario is also true of other [uF] e.g. phi-features. For example, [uPhi] on T needs to be removed from the narrow syntax once it gets valued by a DP. That is, formal features including Case features and phi-features are captured under a more general principle, namely the Principle of Full Interpretation, which enables us to abolish the Case Filter from the system. Also, Freidin and Vergnaud (2001) say:

"In short, the heuristic of eliminating overlapping conditions, which has resulted in much fruitful research over several decades, is one of the central motivations for switching from the analysis of GB to that of the MP."

(Freidin and Vergnaud 2001: 642)

Furthermore, the shift from P&P to minimalism influences other aspects including levels of representation and structural relations such as government and (representational) c-command (e.g. in Reinhart 1976), both of which have played central roles especially in the P&P approach. Epstein et al. (1998) and Epstein (1999) argue that neither government nor (representational) c-command is primitive and address that a reason stems from the existence of D-structure.

> "… by appeal to D-structure (DS), a level of representation, complex 'sentential' structures are first completely assembled, all at once (= the operation called *Satisfy* in Chomsky 1993, MO). Then, and only then, are relations expressible, and since the structure has already been entirely built (by virtue of the role and function of DS) the only way to express relations is to nonexplanatorily define them on the already-built representation."
>
> (Epstein and Seely 2002: 6)

For the issues of D-structure, government and c-command, there is apparently room for compromise in accordance with Epstein and Seely's points. At the present point, that is, there is no answer to *why*-questions such as "why is government, not other equally definable representational relations, necessary?", "why does c-command have to be that way?" and "why does D-structure exist?" As an attempt to answer these questions, for example, Epstein et al. (1998) and Epstein (1999) suggest "derivational" c-command, where the structural relation is subsumed into the operation *Merge* which is presumably attributable to third factor conditions in Chomsky's (2005) sense.

2.3.3 Derivation vs. Representation: Toward More Restricted Derivation

Another thing worth mentioning about the shift from the P&P approach is that the minimalist program shifts toward a more "derivational" account, which means more "restricted" system, within which there is the possibility of appealing to third factor notions regarding computational efficiency (cf. Chomsky 1998). The P&P system is ("virtually") a rule-free system, which allows syntax to freely generate any representations, and then the resulting representations are ruled out by filters such as the Case Filter stipulated to apply at S-Structure level (i.e. representational approach). In other words, the syntactic computation in the P&P system is little constrained and filters sort out every kind of representations, so that the system lays all the empirical burden on those filters. Thus, the question why some principles are representational while others are derivational is avoided because all principles are representational in the rule-free system. But filters are not fully explained as mentioned in the previous section. On the other hand, minimalism eliminates the syntactic filters while syntactic computation constructing representations are more highly constrained. That is, it is no longer rule-free but rule-constrained. Chomsky (1998) says:

"To clarify, by a 'derivational' approach, I mean one that takes the recursive procedure literally, assuming that one (abstract) property of the language faculty is that it forms expressions step-by-step by applying its operations to the pool of features… By a 'representational' approach I mean one that takes the recursive procedure to be nothing more than a convention for enumerating a set of expressions…"

"Language uses the derivational approach."

(Chomsky 1998: 126)

That is, minimalism puts more weight on each step of structure-building (i.e. each rule application). In the representational approach, on the other hand, the resulting representations are more important than how the representations are constructed. But why is this shift made? Chomsky (1998) also writes:

"… the recursive operation is designed so as to overcome problems of computational complexity."

(Chomsky 1998: 126)

The iterative rule application reduces computational complexity involved in linguistic computation which is in favor of a more simple and economical system. While minimalism still maintains representational aspects in that the resulting mental representations need to satisfy Bare Output Conditions imposed by the external system (Chomsky 1995a), it is also derivationally constrained.[1]

As alluded to in the previous section, for example, the D-structure representation in the P&P approach is one appearing "all at once" observing the X'-schema. This seems to be one of the exact examples, which regards "the recursive procedure as nothing more than a convention". With respect to this issue, Epstein et al. (1998) writes:

> "Well-formedness conditions on phrase-structure representations (such as X'-structure) are rejected in favor of a *derivational* approach to structure-building, whereby admissible structures are determined by whether or not they can be constructed by an apparatus of Binary and Singulary Generalized Transformations (GT)--namely the rules Merge and Move, respectively."
>
> (Epstein et al. 1998:5)

The D-structure representation in P&P is NOT one constructed by successive application of structure-building rules, so that, to move to minimalism, it has to be somehow reformulated. In Chomsky's (1995b) system of Bare Phrase Structure, the operation called *Merge* execute structure-building: "Applied to objects α and β, *Merge* forms the new object γ… γ must be constituted somehow from the two items α and β… (Chomsky 1995b:396)". By applying *Merge* recursively, a representation appearing in an "all-at-once" fashion no longer exists but rather every syntactic representation created by Merge application is constructed under the recursive procedure. In this sense, syntactic computation is more restricted in minimalism than in the P&P approach. The operation *Merge* opens up new possibilities of explanation of syntactic relations, which leads to a higher explanatory level. Then, as Epstein et al. (1998) says:

[1] Another type of derivational approach is suggested in Frampton and Gutmann (2002) called *Crash-Proof Syntax*, which I will discuss in Ch.6.

"…if X and Y are concatenated [by *Merge*, MO] then X and Y naturally enter into syntactically significant relations. If correct, this theory will allow us to dispense with stipulated and hence unexplained definitions of syntactic relations defined on phrase-structure representation."

(Epstein et al. 1998: 3)

Clarifying the nature of *Merge* and its mode of application is a central concern in this thesis.

This section overviewed the motivation to shift from the P&P approach to minimalism, why the shift is important and what progress has been made so far. The aim of this thesis is also to push forward the minimalist spirit, which if successful leads us to a higher level of explanation and finally we hope to a more simple and perfect system as nature is inclined to be.

The next section discusses Chomsky's (2000 and later) phase-based derivational approach, which I specifically assume in the thesis, and introduces some of the key concepts in the system.

2.4 The Phase-Based Derivational Approach

As discussed in the last sections, "fewer" or "simpler" has been a key term in the development of generative grammar. By answering *why*-questions, explanation proceeds to higher levels and gets simpler and simpler, for which minimalism aims. One of those attempts is how to reduce computational complexity (i.e. complexity of the generative procedure.) Chomsky (2000) writes:

"One category (of this issue, MO) concerns 'least effort' conditions, which seek to eliminate anything unnecessary: (a) superfluous elements in representations, (b) superfluous steps in derivations."

(Chomsky 2000:99)

The legibility conditions require semantic and phonological representations to be legible by the interfaces, so that those representations must be fully interpretable by containing only interpretable features (i.e. no superfluous elements). Also, the derivation constructing those representations in narrow syntax must be the most economical among possible derivations based on the same lexical array/numeration, under the inclusiveness condition, which states that no new element can be added in the course of the derivation. Observing these two conditions, the legibility condition and the inclusiveness condition, leads linguistic computation to having less computational burden.

Moreover, Chomsky (2000) continues:

"Another category seeks to reduce 'search space' for computation… yet another imposes 'local' determinability conditions (barring 'look-ahead', 'backtracking', or comparison of alternatives.)"

(Chomsky 2000:99)

Search space is limited to a small computational unit called a *phase*, namely CP and vP, by Phase Impenetrability Condition. Syntactic computation iteratively applies within these units, and once the procedure is done, the operation *Transfer* (Chomsky 2004) sends the complements of phase heads to the interfaces before the derivation proceeds to the next phase. That is, by "forgetting" the representation whose computation is already done, C_{HL} (= computation of human language) comes to have less computational complexity. Also, syntactic operations such as Merge are applied "blindly", i.e. without knowing what its application finally causes. The evaluation will be made later at the interface (i.e. whether or not the resulting representations satisfies the interface conditions).

In this thesis, I adopt that the phase-based derivational system. Within such a system, the focus will be given to the issues of how Merge works in certain environments, how Transfer applies and how phenomena requiring global computation are explained.

The next section overviews another new concept called the Feature-Inheritance system discussed in Chomsky (2007, 2008) and its relevant discussion in Richards (2007), both of which play central roles throughout the thesis.

2.5 Chomsky (2007, 2008): The Feature-Inheritance System

In this section, I consider Chomsky's (2007, 2008) feature inheritance system and reveal new aspects of its predictive content. Since *On Phases*, the treatment of T (and also V) has radically changed: T does not bear phi-features inherently, but rather inherits phi-features from C.[2] Consequently, T cannot operate as a probe until C is introduced into the derivation. This feature inheritance analysis from C to T makes it possible to non-lexically distinguish a finite/control T from the one appearing with raising predicates. T in raising predicates is not selected by C, so T does not inherit phi-features. Lacking phi-features, T appearing with raising predicates does not have the ability to value Case. That is, the availability of C determines the potential of T. Therefore, we no longer need to stipulate that the lexicon contains two different Ts. Rather, the features of the sole lexical T are determined "functionally" by the syntactic (c-selectional) context in which the T appears. Crucially, under this system, movement to Spec-CP and to Spec-TP takes place simultaneously and independently. This analysis, whereby certain movements to Spec-CP do not proceed through Spec-TP (more generally, A-movement does not feed A'-movement), is empirically motivated to account for asymmetries in the suppression of subject condition effects as presented in Chomsky (2008):

[2] I assume that the same feature inheritance system applies to v/V following Chomsky 2007, 2008: V inherits its features from v. For the limited scope of this paper, however, we mainly focus on C-to-T inheritance.

(3) a. *Extraction out of the object DP*
Of which car did they find [the driver _]?
b. *Extraction out of the subject DP*
*Of which car did [the driver _] cause a sandal?
c. *Extraction out of the passive subject DP*
Of which car was [the driver _] awarded _ a prize?
(Chomsky 2008:147)

In the grammatical (3a), PP is extracted out of the object DP. In the ungrammatical (3b), the same extraction takes place out of the subject phrase. The sentence is ungrammatical because of “a subject condition violation”. In (3c), like (3a), PP is apparently extracted out of the subject phrase. Nevertheless, the sentence is grammatical in contrast to (3a). Thus, (3c) behaves like object extraction but NOT like subject extraction. How can this be explained? To capture this, Chomsky proposes that in (3c), the wh-phrase ("of which car") moves out of the post verbal position directly to Spec-CP. "Passivization" (of "the driver of which car") i.e. movement to Spec-TP applies independently and simultaneously. Crucially, the wh-PP “of which car” was never extracted from the subject position. This explains why (3c) is grammatical, i.e. it fails to involve a subject-condition violation. Only in the case of (3b), a wh-phrase is necessarily extracted out of the subject position Spec-vP. In the other two cases, a wh-phrase moves directly from within the object position to Spec-CP. (see Chomsky 2007, 2008 for further details of this analysis.) This system clarifies an unclarity regarding the possible status of TP as a phase head: phases are only CP and vP, but not TP in that T does not have the ability to serve as a probe by itself and has to inherit features from C.

I adopt the feature inheritance system in the following discussion and further consider its implications. A fundamental question concerning this system still remains unanswered: How/why do features on C get inherited by T? Richards (2007) claims that all of the uninterpretable features (henceforth [uF]) on C are sent to T as a necessary precondition for convergence.[3] He deduces this from Chomsky's system of feature-valuation and the timing of Transfer. According to Richards' argument, Transfer (i.e. transfer to the interfaces) and feature valuation must occur simultaneously for convergence. That is, one operation can neither precede nor follow the other. If Transfer applies BEFORE feature valuation, [uF] is sent to the interface unvalued, which causes crash of the derivation. If Transfer applies AFTER feature valuation, on the other hand, the distinction between syntactically valued [uF] and [iF] disappears in the eyes of Transfer (See Chomsky 2001, 2007 and Epstein and Seely 2002 for analysis and possible problems). This leads to crash of the derivation because the operation Transfer fails to remove uninterpretable, but now valued [uF] from syntactic objects that are sent to the semantic component. With respect to this issue, Chomsky (2007) writes:

"If transferred to the interface unvalued, uninterpretable features will cause the derivation to crash. Hence both interface conditions require that they cannot be valued after Transfer. ….. Furthermore, this operation [= Transfer, MO] cannot take place after the phase level at which they are valued, because once valued, they are indistinguishable at the next phase level from interpretable features, hence will not be deleted before reaching the Conceptual-Intentional (CI) interface. It follows that they must be valued at the phase level where they are transferred, that is, at the point where all operations within the phase take place and the Transfer operation therefore "knows" that the feature that has just been valued is uninterpretable and has to be erased at (or before) CI."

(Chomsky 2007: 18-19)

That is, a valued [uF] has to be deleted by Transfer early enough for the computational system to distinguish it from inherently interpretable features [iF]. This entails that valuation and Transfer must occur simultaneously for a derivation to converge. Given this argument, Richards (2007) suggests that [uF] cannot remain on C but has to be

[3] Chomsky (2007) claims that an edge feature (EF) is an uninterpretable feature and does not reach the interfaces by being deleted by Transfer like other uninterpretable features. However, EF can be deleted by Transfer regardless of whether it is satisfied or not, unlike others. Section 3 also discusses this issue.

discharged to T. This is because Transfer of a phase edge (including C) is suspended until the domain of the next higher phase is transferred, based on the Transfer/Spell-out system suggested in Chomsky (2000). But syntactically valued features appearing at the edge (such as valued [uPhi] on C, which would appear if C did NOT transfer its phi to T but instead probed directly thereby valuing [uF] on the edge of C) are valued [uF]. These valued [uF] are indistinguishable from inherently interpretable features [iF]. Having only one-phase memory, Transfer will not know to remove them at the next phase level, and the derivation will crash. Therefore, (Richards argues) convergence is possible only when all of the valued [uF] are included in the domain of a phase. This is precisely what C-to-T feature inheritance accomplishes and so Richards deduces the application of C-to-T.[4]

This analysis of simultaneity of Transfer and feature valuation has an interesting implication. Richards only focused on features on a phase head (probe), so that feature inheritance must happen for convergence. But, what about goals such as DP? DP also bears [uF], namely [uCase], which is valued by T (or V). If a phase head (which, by definition, occupies the edge) is not allowed to bear valued [uF], the same should be true of elements moved to phase-edge positions, since Transfer of these objects is also suspended until the domain of the next higher phase. In Ch3, I will address this issue and propose a new analysis based on Obata (2008) and Obata and Epstein (2008, in press).[5]

2.6 Summary

This chapter has (selectively) reviewed the history of generative grammar focusing on the

---

[4] One might think that languages which have C-agreement are problematic for this view in that C cannot trigger Agree because it lacks [uF] given feature inheritance from C to T. I will discuss this issue later in this thesis.

[5] In later discussion, I will assume that features are inherited before Agree takes place following Chomsky's original claim.

shift from the P&P approach to the minimalist program. The shift is not mere framework change but to make progress to attain higher levels of explanation by answering the *why*-question: *why* is it that way? As Galileo states that nature is simple and perfect and is explained by mathematics, it is not surprising to assume that language, which is also part of nature (the human mind/brain), is simply designed and a perfect system and that is studied by means of established mathematical methods including finding redundancy within theories and unifying theories to whatever extent it is empirically possible to do so. Minimalism is a program based on Galilean method. This thesis tries to attain the extremely difficult (but worthwhile) goals set by the minimalist program, namely to seek explanation, not mere description of the properties of the human language faculty. The thesis also makes the same attempt as minimalism does and aims for providing more elaboration for the minimalist program.

Chapter 3

What moves where?: Simultaneous Attraction by Multiple Heads

3.1 Framing the Issues

This chapter further explores implications of Chomsky's (2007, 2008) feature-inheritance system and Richards' (2007) analysis, which were summarized in Ch. 2.[1] Richards (2007) explains why uninterpretable features cannot remain on phase heads: If valued [uF] is stranded on a phase edge after Transfer, it is not indistinguishable by Transfer from [iF], so that Transfer cannot detect valued [uF] and fails to delete it. As a result, valued [uF] is sent to the interfaces and causes crash of the derivation. This is why [uF] cannot remain on a phase head after Transfer of a phase-head-complement.

In this chapter, I will further extend this Chomsky/Richards Value-Transfer issue to another aspect--elements moved to the phase edge. Remember that the portion stranded after Transfer is not only a phase head but also its edge position. Given their argument, it is predicted that elements undergoing Internal Merge to a phase edge position also cannot retain valued [uF], e.g. valued [uCase]. But, how is that requirement met? In the next section (Section 3.2), based on Obata (2008a) and Obata and Epstein (2008, in press), I propose the Feature-Splitting Internal Merge, where features on a single element are split into two landing sites. Under the feature-splitting system, valued [uCase] on a moving e.g. a wh-DP is split off and does not move to a phase edge position,

[1] Aspects of Ch.3 are based on Obata (2008a) and Obata and Epstein (2008, in press, to appear).

so that the portion which remains after Transfer of a phase-head-complement (i.e. the phase head and its edge) will not contain any valued [uF]. Thus, via Feature-Splitting the Chomsky/Richards reduction prohibiting syntactically valued [uF] on a phase edge, can be maintained. In Section 3.3, moreover, I will demonstrate that the proposed system succeeds in giving a "local" agreement-based account to improper movement phenomena. In Section 3.4, the proposed system is further applied to Bantu languages which allow improper movement, unlike in English. In Section 3.5, some of the consequences obtained from the current analysis are discussed.

3.2 The Mechanics of Feature-Splitting

As discussed in the last section, no valued [uF] is allowed to appear at a phase-edge, given the Chomsky-Richards analysis. For instance, valued [uCase] on a wh-DP should never occur at Edge-CP in a convergent derivation:

(1) Who do you think bought the book?[2]

a. Embedded CP

$[_{CP}$ C $[_{TP}$ T$_{[uPhi]}$ $[_{vP}$ who$_{[iPhi][uCase][Q]}$ $\boxed{[_{VP} \text{ bought the book}]}]]]$

AGREE TRANSFER

VP is transferred.
[uPhi] on T agrees with "who".

b. $[_{CP}$ ***who*** C $[_{TP}$ who T$_{[uPhi]}$ $[_{vP}$ who$_{[iPhi]\cancel{[uCase]}[Q]}$ $[_{VP}$ . . . $]]]]$

INTERNAL MERGE

A and A'-movement occur separately.
A-movement does not feed A'-movement following Chomsky (2007, 2008).

c. $[_{CP}$ ***who*** C $\boxed{[_{TP} \text{ who T}_{[uPhi]} \; [_{vP} \text{ who}_{[iPhi]\cancel{[uCase]}[Q]}}$ $[_{VP}$ . . .$]]]]$

TRANSFER

Let us focus only on the derivation of the embedded CP. The embedded C is externally

[2] I will ignore the accusative Case assignment to make the discussion simpler.

merged into the derivation bearing [uPhi] which are then inherited by T. T bearing [uPhi] agrees with the subject DP "who" bearing [iPhi] at Edge-vP as in (1a). As a consequence of this phi-agreement, [uCase] on "who" is valued. The edge feature (EF) on C and on T each (independently and simultaneously) attracts "who" to its edge position as illustrated in (1b). Then, TP is transferred as in (1c). If Richards' analysis is on the right track, "who" at the edge of CP must NOT bear any valued [uF]. If it did, the derivation would incorrectly crash at the next highest phase since Transfer would not know to remove valued [uF] which is no longer distinguishable from [iF].

There appear to be at least two ways to overcome this problem. One is valued [uCase] on "who" "disappears" by some mechanism, so that it is not copied to the occurrence of “who” at the edge of CP. The other way is that valued [uCase] does appear at the edge, but (contra Chomsky) the computational system, specifically, Transfer, still can still somehow see the difference between valued [uF] and [iF] (this assumption would in turn deprive us of maintaining Chomsky/Richards’ deduction, since this assumption is inconsistent with Chomsky/Richards by virtue of allowing syntactically-valued [uF] on the edge). In this paper, I will pursue the former possibility and will maintain Chomsky/Richards’ deduction. (See Epstein, Kitahara and Seely 2008 for further discussion regarding the latter possibility.)

What mechanism makes it possible that valued [uCase] on "who" at the edge of vP is not copied to the edge of CP? Recall that T can work as a probe only by receiving phi-features from C. After these features are inherited from C, T finally begins to work as a probe. Then, as exemplified in (1b), EF on C and on T each independently attract the single element "who" occupying edge of vP, (which as Chomsky (2008) argues explains suppression of subject condition effects, since A-movement does not feed A'-movement, as mentioned in Section 2.5). What happens in the simultaneous attraction of the single element by these two different heads? I propose that features on the attractee are "split" into the two different landing sites (= feature-splitting).

(2) (=(1b)): Feature-Splitting

[Q] C [uCase][iPhi] T [vP WHO [iPhi][uCase] [Q]

As a consequence of this simultaneous attraction by C and T, features on "who" are decomposed into the two positions as displayed above i.e. [Phi/Case] moves to Edge-CP and [Q] moves to Edge-CP. Since Chomsky (2000), Case-valuation on DP has been a reflex of phi-valuation, so that [Case] and [Phi] are unified. As in (2), if T attracts [Case & Phi] and C attracts [Q], valued [uCase] on DP is transferred as a part of TP = the phase-head-complement of the phase head C. Therefore, this valued [uCase] never makes it to the edge of CP and Richard's (deductive) condition is maintained. In addition, the derivation converges, as required for empirical adequacy. Furthermore, the feature-split analysis is compatible with Chomsky's (1964/1995a) view that non-branching lexical wh-phrases are in fact composed of two distinct morphological feature sets: WH on the one-hand and an indefinite QP like "someone/something". Given this view, I naturally hypothesize that WH and "someone/something" each move independently.

One might ask the following two things regarding feature-splitting: [1] What mechanism makes it possible that [uCase & iPhi] are attracted by T but not by C?, and [2] Why can a single feature or featural proper subset of a syntactic category land at edge positions?[3] One possible answer to the first question is that T only attracts features it has agreed with. In (2), T phi-agrees with [iPhi] on DP and [uCase] on DP is valued as its reflex. That is, both [uCase] and [iPhi] on DP "participated" in agreement with T. In contrast, [Q] is not involved in this phi-agreement. Therefore, T attracts only [uCase & iPhi] which it phi-agreed with while C attracts the rest, namely [Q] (and phonological features in English).[4] In fact, this view is fully compatible with (if not forced by) Chomsky's (2007) characterization of the A/A'-distinction: "A-movement is IM (internal merge) contingent on probe by uninterpretable inflectional features, while A'-movement is IM driven by EF. (Chomsky 2007:24)" In other words, an A-position is one that results from attraction by EF immediately preceded by Agree in contrast to A'-movement which is triggered solely by EF. The difference between A-movement and A'-movement has been widely accepted. Feature-splitting, whereby T attracts only features which it agreed with, enables us to capture derivationally and featurally two different types of movement. In this sense, the feature-split analysis is a natural implementation of Chomsky's A/A'-movement distinction and is deduced from the (explicable) prohibition against valued [uF] on the edge. Note also that if interpretation is at the interface, these distinct features allow us to featurally represent A vs A' in the interface representation, which presumably does not have access to the earlier-applied movement type that created the position "back in" the narrow syntax. In this sense, perhaps feature-splitting is the narrow syntax optimally exploiting independent mechanisms (Agree vs. pure edge attraction) to featurally distinguish position types, rendering stipulative position-type definitions in the P&P era (e.g. "Spec-CP = A'-position, while Spec-TP = A-position") eliminable.

[3] This is an extension of Chomsky's (1995b) MOVE-Feature proposal. Also see Toyoshima (2000) and Matushansky (2006) for related ideas.
[4] See Obata (2008b) for more details about how phonological features are split.

As for the second question, a single feature or featural subset CAN be regarded as a maximal projection under bare phrase structure: "a category that does not project any further is a maximal projection XP and one that is not a projection at all is a minimal projection X^0. (Chomsky 1995b:396)" In the configuration of (2), the moved WH D^{max} does not project further at Spec-CP and at Spec-TP. Therefore, those split features i.e. [Phi & Case] and [Q] can each be regarded as a maximal projection in its own right. In this sense, the current analysis also lends support for bare phrase structure.[5,6]

One might wonder how an element internally-merged to a phase-edge position carries a Case-morpheme despite the fact that valued [uCase] is split off to the edge of TP/VP prior to movement to a phase edge position under the current analysis. Consider:

(3) *English*
Whom did John see?

(4) *Japanese*
Dare-o_1 Taro-ga t_1 mi-ta no?
Who-ACC Taro-NOM see-PAST Q
"Whom did Taro see?"

In (3), "whom" is valued as accusative Case by V before undergoing Internal Merge to the edge of CP. Also in (4), "Dare-o" is valued as accusative Case before undergoing scrambling. If Case-features are uninterpretable features as we assume, however, they cannot be moved to a phasal edge position along with other interepretable features such as [Q] or [iPhi] because of Chomsky/Richards Value-Transfer simultaneity analysis, which disallows syntactically-valued features on the edge. With respect to this issue,

---

[5] Hisatsugu Kitahara (p.c.) suggests that feature-splitting may correspond to DP decomposition into D and NP (assuming that DP is not a phase): D bearing [Q] moves to Edge-CP/vP and NP bearing [phi & Case] moves to Edge-TP/VP.

[6] Notice that feature-split violates neither the inclusiveness condition nor the no-tampering condition. Feature-split involves no new features but only splits existing features observing the former condition. With respect to the latter condition, in the configuration of (2), [Q] is internally-merged by C. The no-tampering condition says that merge of [Q] and C leaves the two syntactic objects unchanged. Feature-split only makes it possible that [Q] moves separately from [Phi & Case], so that it does not affect the merged syntactic objects. That is, feature-split is compatible with these two conditions.

Chomsky (2007) says:

> "Valuation of uninterpretable features clearly feeds A'-movement (e.g., in "whom did you see?"). Hence valuation is "abstract", functioning prior to transfer to the SM interface, as are the uninterpretable features themselves."
>
> (Chomsky 2007: 18)

Although he does not clarify the details, I interpret his "abstract" valuation as motivated so that we can maintain Chomsky/Richards deduction of feature-inheritance but which disallows "whom"--bearing a valued (accusative) Case--on the edge. I assume that abstract valuation operates as follows maintaining Chomsky/Richards deduction:

(5) $V_{[uPhi][Phon]}$ Agree $DP_{[iPhi][uCase][Phon]}$

(6)
T ... DP
[uPhi] --Agree--> [iPhi]

v[uPhi] <--Value-- [iPhi]
[Phon/α] --Value--> Value v[uCase]
Abstract Phon Specification as +Nom
[Phon / Nom]

Taking T as an example, [uPhi] on T, which is inherited from C, triggers phi-agreement with DP as exemplified in (5). (6) is the detailed illustration of phi-agreement and its subsequent Case-valuation in (5). First, [uPhi] on T agrees with [iPhi] on DP. Then, phi-values are assigned from DP to T. At this point, [Phon] on T undergoes "abstract" specification. If T is e.g. "be" and the DP is e.g. "Mary", [Phon] on T is abstractly specified as [Phon/IS]. Notice that this phon specification takes place abstractly, so that realization of "BE" as /iz/ is calculated later in the morphological component. (cf. Legate 2008) At the same time as this abstract phon specification of T occurs, [uCase] on DP is valued as a reflex of phi-agreement. Now, the abstract phon specification of DP takes place contingently on the Case-valuation. [Phon] on DP is abstractly specified as Nom.

Again, the concrete phonological realization is assigned later in the morphological component. Notice that valued [uCase] on the DP has to be deleted in a convergent derivation in accordance with Chomsky/Richards' deduction. Therefore, the idea here is that "footprints" of Case-valuation are left behind in a form of abstractly specified phonological features.[7]

Now, let us see how the abstract phon specification system works for a sentence like (3) assuming the feature-splitting system. For ease of discussion here, "whom" is written as "WH" since its realization is not calculated within narrow syntax.

(7) Whom did John see?

Step1: $[_{vP}$ John $[_{VP}$ see$_{[uPhi][phon]}$ WH$_{[iPhi][uCase][Phon][iQ]}$ $]$

Phi-agreement between V and WH

Step2: $[_{vP}$ John $[_{VP}$ see$_{[uPhi][Phon/\alpha]}$ WH$_{[iPhi]\cancel{[uCase]}[Phon/Acc][iQ]}$ $]$

[Phon] on V is abstractly specified.
[uCase] on WH is valued.
[Phon] on WH is abstractly specified.

Step3: $[_{vP}$ WH$_{[iQ][Phon/Acc]}$ John $[_{VP}$ WH$_{[iPhi]\cancel{[uCase]}[phon/Acc]}$ V WH$_{[iPhi]\cancel{[uCase]}[Phon/Acc][iQ]}$

Feature-Splitting

Let me focus on the derivation for the vP phase. As presented in (6), I suggest that abstract phon specification is also part of phi/Case-valuation. As in Step 1and 2, [Phon] on V and DP are both specified abstractly, which makes it possible that WH is realized as /hum/ and the verb as /si/ later in the morphological component. Notice that [Phon] is automatically specified as a consequence of Case-valuation, but [Phon] is not an

[7] I believe that this phonological specification does not violate the Inclusiveness Condition. I conjecture that the phonological specification is another type of feature-valuation like the kind [uCase] or [uPhi] undergoes. Another possibility is that [phon] contains every possibility of Case-realization e.g. /hi/, /hiz/, /him/. Under this assumption, phon specification deletes unnecessary candidates and leaving only one in the representation. This idea seems to be reminiscent of the checking theory.

unvalued feature unlike Case. Therefore, underspecification of [Phon] does not cause featural crash (although causes violation of the recoverability condition in some cases.) That is, [Phon] can be moved to the phase edge in convergent derivation as in Step 3 along with [iQ]. But Case is a feature requiring valuation, so it cannot appear at the edge in convergent derivation under Chomsky/Richards' deduction. Since all the WH copies bear [Phon/Acc], they can be realized as /hum/ later in the morphological component. The same discussion is also applied to the Japanese case in (6). This is why elements whose Case-features are split off are properly realizable even if they do not carry Case-features. This is one of the possible ways to derive (3) and (4) maintaining Chomsky/Richards' Value-Transfer simultaneity analysis and formalizing Chomsky's "abstract" hypothesis, which will be assumed in this thesis.

In a nutshell, for Richards' deduction based on Chomsky's system to go through, there is a need to avoid the appearance of valued [uF] on a phase head (Richards' deduction of inheritance) and, as I note here, there is entailed a more general prohibition against any valued [uF] including those of the goal of an Internal Merge operation appearing at an edge position. As for a phase head probe, Chomsky's feature inheritance system forces the valued features on C not to stay there i.e. remaining on the edge. Regarding goals internally-merged to phase edges, the analysis proposed here splits off valued [uCase], and just like C-to-T feature inheritance (deduced by Richards), keeps valued [uF] "off the edge" and within the transferred phase head complement ensuring convergence hence empirical adequacy. In the next section, I present empirical and theoretical advantages provided by the feature-split analysis by considering improper movement phenomena as a case study.

3.3 Explaining Improper Movement

The proposed analysis is independently motivated by explaining improper movement phenomena. Improper movement, originally discussed in Chomsky (1973), was characterized as movement from a COMP position to a non-COMP position, and was illustrated by (8). Chomsky (1973) stipulates a ban on COMP-to-non-COMP movement excluding the derivation of (8):

(8) *[$_{CP}$ What [$_{TP}$ t'' was asked [$_{CP}$ t' [$_{TP}$ to read t by John]]]?

Chomsky (1973: 98)

This prohibition requires an explanation and also raises a puzzle regarding learnability. In more (but not the most) current terms, why is it that A-to-A, A'-to-A' and A-to-A' are allowed, yet A'-to-A movement is by hypothesis excluded? How do children know or come to know this exception? The question: "What is the nature of the mechanism or constraint responsible for the prohibition?" has been a central concern since Chomsky (1973).

. Before going into further discussion about feature-splitting, the following three sections (Section 3.3.1 through 3.3.3) are devoted to reviewing some previous approaches to the phenomena at issue.

3.3.1 May (1979): A Condition C Approach

May (1979) first sought to explain Chomsky's (1973) stipulation and has been one of the most widely assumed approaches. His attempt is to deduce COMP-to-COMP, i.e. successive cyclic A'-movement from the binding theory.[8] The ban on improper

[8] In this paper, I rephrase May's idea in terms of the binding theory. May (1979) in fact predates Chomsky's (1981) formulation of Condition C. May instead appeals to Chomsky's (1980) "On Binding" (OB) indexing algorithm. May reveals that under this independently motivated algorithm, improper movement results in a

movement is a consequence of violating the independently motivated Binding Condition C under the assumption that a wh-trace is an R-expression.[9] (See also Freidin and Lasnik, 1981.) Let us see how May's system works for Chomsky's (8), repeated as (9):

(9) *[$_{CP}$ What [$_{TP}$ t'' was asked [$_{CP}$ t' [$_{TP}$ to read t by John]]]?

The wh-phrase is extracted from the complement position of "read". On the way to the matrix [Spec,CP], it stops at the embedded [Spec,CP] and then moves to the matrix [Spec,TP]. The step from t' to t'' constitutes A'-to-A movement, which Chomsky (1973) prohibited by stipulation. In May's analysis, the source of improper movement is not the movement from t' to t''. Rather the wh-trace t, created by movement to Spec-CP, is by definition an R-expression. However, it is A-bound by the co-indexed and c-commanding t” in Spec-TP as follows:

(10) *[$_{CP}$ What [$_{TP}$ t_i''(A) was asked [$_{CP}$ t_i'(A') [$_{TP}$ to read t_i(A) by John]]]?

Thus, (SS) representations resulting from improper movement are ruled out by virtue of constituting a Condition C violation. This system can appropriately capture all of the three types of binding relations wh-movement creates:

(11) a. A'→A'→A': OK (e.g."How do you think John bought the book _?")
b. A→A'→A': OK (e.g. "What do you think John bought _?")
c. A→A'→A (Condition C violation (e.g.(10)))

---

variable bearing a referential index identical to its anaphoric index, e.g.

(i) NPi{i}

As May notes, an NP assigned such an index would be construed as necessarily referentially distinct from its own reference, since it bears the referential index i, but also the anaphoric index i, where anaphoric indices are interpreted as the set of referential indices from which an NP is interpreted disjointly.

Interestingly, the feature-splitting approach below similarly seeks to deduce improper movement from an interpretability failure at the LF interface, but for me, not a binding theoretic one, but rather illegitimacy induced by the appearance of unvalued phi and/or Case features at the LF interface.

As I will discuss below, there are two types of (non-unified improper movement phenomena, one kind in which their original launch position of movement is not Case-marked. I believe May’s (1979) analysis would not rule these examples of improper movement out, since crucially May’s (1979) analysis inserts variables, bearing anaphoric indices, only in Case-governed positions.

[9] Condition C: An R-expression is A-free. (Chomsky 1981: 188)

Thus, May (1979) deduces the ill-formedness of improper movement by appeal to the independently motivated binding condition C, a condition on representation.

3.3.2 Fukui (1993): Chain Uniformity

Another analysis of improper movement is Fukui's (1993) chain uniformity approach based on an earlier version of the minimalist program suggested in Chomsky (1995a). Under the chain uniformity condition, a chain must be uniform with respect to A/A' properties to satisfy the full interpretation principle at the interface level. That is, an A→A→A or A'→A'→A' chain is legitimate at the interface level since all of the members comprising the chain share uniformly A or uniformly A' properties. On the other hand, an A→A'→A' chain is regarded as non-uniform in this sense. As exemplified in (11b), however, the sentence which involves this chain formation has to be generable. However, it is not uniform. In order to solve this contradiction, Chomsky and Lasnik (1995a) suggest that intermediate traces can be deleted (under Last Resort) but only when their presence makes a chain non-uniform: an extension or account of trace-deletion as originally proposed in Lasnik and Saito (1984, 1992). As a consequence of trace deletion, the intermediate trace at an A'-position in the A→A'→A' chain is deleted, leaving just an A→A' chain behind. Since Chomsky and Lasnik (1995) postulate that the resulting chain composed of operator and variable is uniform, (more on this below) this A'-A chain is deemed a legitimate object.

(12) a. A→A→A = uniform "John seems to buy the book."
b. A'→A'→A' = uniform "How do you think John bought the book _?"
c. A→~~A'~~ →A' = uniform "What do you think John bought _?"

Now, consider the chain formed by improper movement. In (10), an A→A'→A chain is created. As we already know, an A→A'→A chain is not uniform since the only kind of uniform chain which mixes A and A' properties is a two-membered chain consisting of an operator and variable. As in (12c), the trace deletion operation is also applied to make the chain uniform. Consequently, an A→A'→A chain is transformed into an A→A chain, which corresponds to a uniform chain in (12a).

(13) *[$_{CP}$ What [$_{TP}$ t''(A) was asked [$_{CP}$ ~~t'(A')~~ [$_{TP}$ to read t(A) by John]]]?

Thus, in (13) the trace *t* at the launching site is no longer locally A'-bound since *t'* is deleted. It is locally bound by *t''*, Spec-TP. This configuration is in relevant respects, the same as one generated by A-movement. The trace at the launching site *t* is no longer a wh-trace (locally A'-bound) but an A-trace (assuming SS representational functional determination of empty categories, Chomsky 1982, although the trace is created by movement to Spec-CP, is nonetheless an A-trace since it is locally A-bound). On the assumption that the binding theory being a condition or conditions on semantic interpretations applies only at LF as proposed in Chomsky (1995a), an A-bound variable, violating Condition C, never appears in this derivation. Therefore, Fukui (1993) claims that May's binding approach to improper movement is rendered ineffective under the "trace-deletion and chain uniformity" analysis as coupled with the hypothesis that Binding Theory applies at LF only.

In order to rule out improper movement observing the chain uniformity condition, Fukui (1993) attempts to constrain the operation of chain formation itself by reformulating the original "representational" chain uniformity condition as a "derivational" constraint on the formation of chains.

(14) The Uniformity Condition on Form-Chain
Form-Chain must apply to form a uniform chain.

Fukui (1993: 114)

One radical difference from the original condition is that under (14) a non-uniform chain is never even created in the course of the derivation given this condition. Therefore, the trace deletion operation plays no role because all of the chains formed under this condition are uniform and legitimate to begin with.[10] Now, let us see how improper movement can be blocked under this view. Observing the condition in (14), a uniform chain is formed as follows:

MOVE MOVE

(15) *[$_{CP}$ What [$_{TP}$ t''(A) was asked [$_{CP}$ t'(A') [$_{TP}$ to read t(A) by John]]]?

CHAIN

(14) allows only uniform chains to be created, so the trace t' at the intermediate Spec-CP cannot be included in the chain containing t and t'' as illustrated in (15). Fukui claims that the chain shown in (15) is uniform and legitimate at the interface level. However, as Fukui argues this chain formation violates the economy principle 'shortest possible link' in that it fails to incorporate t' and so chain links are not minimized as fully as possible. That is, this derivation is ruled out in terms of an independent economy condition governing construction of (uniform) chains.[11, 12]

[10] These analyses do not entirely specify the chain formation algorithm nor the chain re-formation algorithm applied after trace deletion. In addition, notice that prior to trace deletion a chain CANNOT be regarded as a single syntactic object, since part of it will be deleted. Notice also that inter-phase, chain formation cannot possibly apply "in a single step of chain-formation" (i.e. assuming "Form a single chain" is a basic operation) occurring in the narrow syntax, since transfer must apply to part of the as yet unformed total A'-chain.

[11] See Fukui for a second proposal excluding improper movement by an economy condition on adjunction; an analysis which seeks to unify improper movement and that-trace phenomena.

[12] The reader will notice that an Operator + argument-variable A'-chain is in fact not uniform with respect to A'/A properties; the head of the chain is A', while the tail is A. In order to avoid excluding such Operator-variable chains as non-uniform, Fukui defines "uniform chain" so that the chain tail is ignored in

3.3.3 Comments on May (1979) and Fukui (1993)

I have reviewed two influential previous deductive approaches to improper movement suggested in May (1979) and Fukui (1993). One of the crucial differences between May (1979) and an approach to be proposed based on feature-splitting (see Section 3.3.4 below) is that the latter excludes improper movement by local computation while the former is based on global computation. As mentioned in Ch.2, Chomsky (2000) discusses that local computation can reduce computational complexity by limiting search space for computation to small units, namely phases. In May (1979), the computational work space expands to the entire representation constructed by syntactic operations. Although Condition C itself requires such global computation, if improper movement is explained within local computation (i.e. without appeal to Condition C), it leads to less complexity, which should be more preferable, and perhaps takes us closer to eliminating all global computation.

In addition, as Fukui himself points out, his chain uniformity approach suffers when dealing with the argument/adjunct asymmetry in wh-islands, which is the core motivation for the original chain uniformity condition in Chomsky and Lasnik (1995) based on the discussion in Lasnik and Saito (1984, 1992). Recall Fukui's version of chain formation (14) does not allow a non-uniform chain to be formed. Accordingly, the operation of trace deletion is nullified. In order to capture the argument/adjunct asymmetry, however, formation of a non-uniform chain and trace deletion accompanying it are both required under Chomsky and Lasnik's (1995) approach. That is, Fukui's analysis is not equipped with the means to explain the argument/adjunct asymmetry. Also,

---

computing the uniformity/non-uniformity of a given chain. (See Fukui 1993, definition (14)). (Cf. Lasnik and Uriagereka, 2005) I regard this as a possibly serious explanatory drawback confronting the Chain Uniformity approach to chain-legitimacy and hence, its application to improper movement.

the chain uniformity condition itself embraces a general problem. Both in Chomsky and Lasnik (1995) and in Fukui (1993), the chain resulting from wh-movement from an A-position as in "What did you see?" needs to be regarded as an exceptional case, i.e. it is obvious that an A→A' chain is <u>not</u> uniform with respect to A/A'-properties. That is, chain uniformity, as defined, seems descriptively inadequate/overly restrictive. (Note also that, even setting aside this problem, there appears to be no independent motivation for assuming that uniform chains, and only these, are semantically interpretable at the LF interface. In addition, the stipulation that uniform A-chains and A'-chains are legitimate, rests also on the definition of A vs. A' which, being GB-era notions, are arguably not clearly defined.) Another possible problem for Fukui (1993) concerns the reformulated chain uniformity condition as a derivational constraint. If Chain Uniformity is to be explained in terms of interface conditions, it is counterintuitive then to also have a constraint such as (14) requiring that ONLY uniform chains can be formed. This is a serious redundancy, and appeals to structure-building derivational constraints on the process of chain formation. If derivational constructs are appealed to, we might ask whether they suffice, as in a chain-free phase-based derivational framework. In this thesis, I argue that certain specific proposals regarding phase-based derivation do indeed suffice to explanatorily exclude improper movement on independent grounds and with no appeal to global interface conditions like Condition C or descriptive Chain conditions. Moreover, I argue that this deduction of the ill-formedness of improper movement is also empirically preferable to previous accounts.

In conclusion, the extent to which the chain uniformity condition itself is empirically motivated and explicable remains unclear. These potential problems motivate the exploration for alternative analyses.

3.3.5 Improper Movement as an Agreement-Failure Phenomenon

In this section, I demonstrate how the ban on improper movement is recaptured locally as an unvalued phi-features crashing phenomenon, under the current phase-based derivational approach equipped with feature-splitting. (The analysis makes no appeal to unbounded dependencies expressed by Condition C and Chain-based approaches, nor to the Activity Condition). There are two types of improper movement. Consider, e.g.

(16) a. *Who seems it is likely to leave?
[$_{CP}$ Who$_1$ seems [$_{CP}$ t'''$_1$ [$_{TP}$ it is likely [$_{CP}$ t''$_1$ [$_{TP}$ t'$_1$ to [$_{vP}$ t$_1$ leave?]]]]]]
b. *Who seems will leave?
[$_{CP}$ Who$_1$ seems [$_{CP}$ t''$_1$ [$_{TP}$ t'$_1$ will [$_{vP}$ t$_1$ leave?]]]]]

In (16a), [uCase] on "who" is valued AFTER movement to the edge of the embedded CP. By contrast, [uCase] on "who" in (16b) is valued BEFORE movement to the edge of the embedded CP. The former case can be explained by saying that [uCase] on "who" is transferred unvalued as part of the lowest TP, which causes crash of the derivation. The same scenario does not go through for the latter case because [uCase] on "who" is properly valued by the embedded T. Let us see how the latter case is derived under the feature-splitting approach (indices on "who" for expository purposes only):

(17) *Who seems will leave?
a. Embedded vP
[$_{\underline{v}P}$ <who$_{1[Q][Phi][Case]}$> [$_{VP}$ leave]]
b. Embedded CP
[$_{CP}$<who$_{3[Q]}$> C [$_{TP}$<who$_{2[iPhi][uCase]}$> T [$_{\underline{v}P}$<who$_{1[Q][iPhi][uCase]}$> [$_{VP}$ . . .]]]]
c. Matrix CP
[$_{CP}$ C$_{[EF]}$ [$_{TP}$ T$_{[uPhi]}$ seems [$_{CP}$ <who$_{3[Q]}$> [$_{TP}$ . . .]]]]
Embedded TP-Transfer

In (17a), the embedded subject "who" is externally-merged into Edge-vP. In (17b), after embedded T phi-agrees with "who", EF on C and on T each attracts the single element "who" occupying Edge-vP. As mentioned in the last section, T attracts only a featural subset of "who" which it agreed with and C attracts the rest. That is, features on "who" are split into the edge of CP and Edge-TP: [uCase & iPhi] move to Edge-TP while [Q] moves to the edge of CP. Then, the embedded TP is transferred and the derivation continues on to the matrix clause. In (17c), [uPhi] on the matrix T seeks a Matching Goal bearing [iPhi]. However, "who" at the edge of the embedded CP has already lost [iPhi] as a consequence of feature-split. In addition, "who" at the embedded Edge-TP is not in the minimal search domain of the matrix T because of the Phase Impenetrability Condition (PIC).[13] Therefore, [uPhi] on the matrix T is not valued, which causes crash of the derivation. The derivation in (17) crashes, locally and featurally, as desired.[14]

What happens in analyses lacking feature-split? If there is no feature-split, it is predicted that "who" in embedded Edge-CP still has its inherent [iPhi]. Therefore, in the configuration of (17c), the matrix T can phi-agree with “who” in embedded Edge-CP. The derivation converges,[15] yielding the wrong prediction. What mechanism excludes this derivation if feature-splitting is not assumed? A separate principle, namely the Activity Condition, is stipulated.

[13] Phase Impenetrability Condition (Chomsky 2000:108): In phase α with head H, the domain of H is not accessible to operations outside α, only H and its edge are accessible to such operations.
[14] As a potential problem the question of how a proper operator-variable binding relation can be established given that the moved wh-phrase does not itself bear phi-features under feature-split. The treatment of binding relations is a general issue which needs to be closely considered within an inclusive minimalist approach and requires further investigation. (It is not clear to me that operator-variable binding invariably requires that phi-features appear on the operator, as in e.g. wh-adjunct variable binding, involving no phi-features at all.)
[15] I will argue that this is what happens in English *tough*-constructions. (cf. Obata and Epstein, to appear) See Section 3.6 below. (cf. Rezac 2006 for a related previous analysis of *tough*-constructions.)

(18) *Inactivity of an XP*:
An XP that eliminates its uninterpretable features (Case, wh-) in rendered inactive.

(19) *The Activity Condition*:
Inactive elements are not accessible for further operations.

(Nevins 2005: 291)

Thus, even if "who" at the edge of the embedded CP has [iPhi], there is no [uF] on it. Therefore, the Activity Condition prohibits the probe (i.e. the matrix T) from agreeing with the inactivated "who". As a result, [uPhi] on the matrix T is not valued, which causes crash. If feature splitting does not exist, the Activity Condition is crucial for excluding such improper movement phenomena. But the Activity Condition is a stipulation, that is, “who” in embedded Edge-CP bears lexically inherent [iPhi], it is in the minimal search domain of upstairs T, and therefore should value T’s phi-features under Match. In other words, feature splitting, which is naturally induced from Chomsky's system, enables us to explain improper movement without appeal to the SMT-violating Activity Condition. Given Nevins’ (2005) arguments that the Activity Condition is empirically problematic, and partially redundant with PIC, our approach is arguably preferable in this respect.[16]

In this section, I have demonstrated how improper movement is excluded as a direct consequence of the feature-split analysis. In the next section, I will suggest further empirical advantages obtained from this analysis.

3.3.6 Further Support: A'-Opacity Effects

As discussed in the last section, the current idea of feature-splitting makes it possible to exclude improper movement without appeal to the Activity Condition. The crucial property of feature-splitting is that DP loses [iPhi] once it undergoes A'-movement to an

[16] See also Bośković (2007) for attempts to eliminate the Activity Condition.

edge-position. DP lacking [iPhi] fails to value [uPhi] on an upstairs T, causing crash. This is what happens in the improper movement derivation (17). In addition, this hypothesis that no [Phi] appears at the edge of a phase enables us to capture so-called "A'-opacity effects" discussed in Rezac (2003): elements bearing phi-features at A'-positions do not block Phi-Agree occurring between A-positions.[17] As an illustration, consider the following data from Icelandic:

(20) Strákarnir$_2$ höfðu [$_{vP}$ **[engu grjóti]**$_1$ [$_{vP}$ *t*$_2$ [$_{VP}$ hent ***t*$_1$** í bílana.]]
The boys had no rock thrown in the cars
"The boys had thrown no rocks at the cars."
(from Svenonius, 2000)

In (20), the derived position (the edge of vP) of "engu grjóti" bearing phi-features intervenes (in the minimal search "command path") between T and the in-situ position of the subject "strákarnir (t$_2$)". Nonetheless, "engu grjóti" is (somehow) invisible as an intervener to phi-agreement applying between T and the subject (i.e. nominative assignment). Why does this intervening DP not induce phi-intervention effects? The same scenario, in fact, applies to English:

(21) What cars does John buy?
([$_{CP}$ what cars does [$_{TP}$ John T [$_{vP}$ **<what cars>** [$_{vP}$ John [$_{VP}$ buy <what cars>]]]])
(pl) (pl) (sing)

When T ("does") phi-agrees with the subject "John", the intervening DP "what cars" bearing "plural" can be mysteriously skipped. As in the Icelandic data, elements at A'-positions do not function as interveners of T-DP phi-agreement in (21). Why do A'-interveners not block A-agreement relations? If the A'-position bears [phi] and Minimal Search --a "highly valued" third factor notion (Chomsky 2005) by hypothesis-- is correct,

[17] As Rezac himself points out, however, the A'-opacity effects are not uniformly observed cross-linguistically, an important empirical issue to which I will return.

then A’ positions hosting phi-features SHOULD block phi-agreement. Any departure from this maximally simple assumption is potentially stipulative and hence inconsistent with the SMT. With respect to this point, Chomsky (2001) suggests: If “what cars” in Edge-vP in (21) undergoes further movement to the edge of CP, phi-agreement between T and the subject “John” is, as a result, viable by virtue of ignoring the intervener-copy of “who”. More precisely, the analysis refers to phonological features on “what cars”, and claims that: if the intervening element at the edge of vP has no phonological features, T-agreement is not blocked. However, it is not clear to me why or how phi-agreement applying in the narrow syntax (not in the phonological component) can be sensitive to, or can detect, phonological features without narrow syntax look-ahead to the phonological component. (See Richards 2004 in which the same problem is pointed out.) [18] That is, it is not clear how the computational system “knows” that phonological features of “what cars” move further (given copy theory) or stay in-situ at the narrow syntax-internal point of application of T-agreement with the subject, an operation which precedes the determination of which parts of the wh-chain will be +/-phonological. Nor is it clear to me why in general Phi-Agree intervention effects should be sensitive to the presence vs. absence of phonological features at all. The system of feature-splitting explains these phenomena (without look-ahead, or Agree-sensitivity to intervening phonological features) by hypothesizing that elements attracted by EF on phase heads, C or v, do not keep their phi-features. If an intervening element has no phi-features, it does not serve as a blocker of phi-agreement. In this sense, the proposed analysis unifies improper movement and A’-Opacity effects, explaining each as the result of the fact that A’-elements lack phi-features. The data in (20)-(21) further support the feature-split

[18] See Kitahara (2006) for more detailed discussion regarding this issue.

analysis.[19,20]

3.3.7 Eliminating the Concept of A/A'-Positions: A Conjecture

The proposed analysis of improper movement and intervention also throw new light on the A/A'-distinction. The existence of two different types of positions has been widely accepted given distinct behaviors in e.g. binding. Two ways of distinguishing these two position-types have been proposed, as follows:

(22) Chomsky (1981: 45):
An A-position is one in which an argument such as a name or a variable may appear in D-structure; it is a potential theta-position. The position of subject may or may not be a theta- position, depending on properties of the associated VP. Complements of X' are always theta- positions… An A'-position is that of an adjunct of one sort or another.

(23) Chomsky and Lasnik (1995: 64):
Given a lexical head L, we say that a position is L-related if it is the specifier or complement of a feature of L. The L-related positions are the former A-positions, …. (i.e. The non-L-related positions are A'-positions. [MO])[21]

---

[19] One might think that the same correct results follow from maintaining the activity condition, (a condition I seek to eliminate following Nevins 2005, Chomsky 2007, Bošković 2007). With respect to (20) and (21), the activity condition straightforwardly rules out A'-intervention. The A'-intervener is simply phi-inactive by virtue of already being Case-marked)." In e.g. (21), I agree that the activity condition correctly prevents object agreement between T and (the inactivated) "what cars". However, I disagree that the activity condition allows the necessary T-Subject agreement which is blocked in the configuration (21) by the defective intervention constraint in analyses lacking feature-splitting.

[20] Another type of data from Icelandic seems to be worth thinking about. Svenonius (2000) presents the following data involving quantifier movement in Icelandic.

(i) a. Á prófinu mun hann snast geta margt.
on the.test will he seem solve many
'On the test he will seem to be able to solve many [problems].'
b. Á prófinu mun hann margt snast geta.
on the.test will he many seem solve
'On the test he will seem to be able to solve many [problems].'

In the above data, the subject "he" moves from the subject position of "solve." Given that the moved quantifier "many" lands at the edge of a phase in (i)b, the subject "he" is transferred before the matrix T agrees with it because "he" is included in a domain of the vP phase headed by "seem". That is, the proposed system wrongly predicts that this type of Icelandic data is ruled out.

However, though this is a problem for the current system, I think this is not a problem only for this system but rather a general problem for this particular formulation of the timing of Transfer. Regardless of assuming feature-splitting, this data causes a problem because as long as "many" lands at the phase edge, the domain including "he" is transferred along with [uCase], causing crash. Although this data is important, this issue is left for future research.

Chomsky (2007) distinguishes A-movement from A'-movement as follows, but the general definition of A vs. A' is not altogether clear.

(24) Chomsky (2007: 24):
A-movement is IM (internal merge) contingent on probe by uninterpretable inflectional features, while A'-movement is IM driven by EF.

In Chomsky (1981), A/A'-positions are defined in terms of potential theta-positions (22). As pointed out in Chomsky and Lasnik 1995, however, this definition is not straightforward in that a certain position needs to be regarded as an A-position despite the fact that there is no theta-relation to its head because the definition is based on transderivational comparison. Chomsky and Lasnik (1995) redefine it in a less global way by introducing the concept of "L-relatedness" as in (23). The L-related positions include specifiers and complements of Agr and T in that those heads bear features of V inherently. By contrast, C does not. The definitions in (22) and (23) refer to phrase structure representations. In contrast, in Chomsky (2007), the differences between A-movement and A'-movement are explained by appeal to what features are involved in Internal Merge as in (24). That is, his definition is for distinguishing not "positions" but rather different types of movement by making use of features involved in Internal Merge. (Although (24) captures the A/A' distinction for Internal Merge in terms of triggering features, (24) does not characterize the A/A' status of externally-merged elements e.g. adjuncts and arguments since A/A' is defined in (24) only with respect to Internal Merge. The A/A' distinction for External Merge is captured by Chomsky's Set-Merge vs. Pair Merge.)

As mentioned in Section 3.2, feature-splitting decomposes features on a single

[21] Interestingly, note that (23) refers to "complement of a feature" as opposed to complement of a head (= bundle of features).

lexical element. If the subject wh-phrase "who" is taken as an example, "who" bears [uCase], [iPhi] and [Q]. T attracts [Phi & Case] which it agreed with and C attracts [Q] purely by EF. That is, a copy of "who" bearing [Phi & Case] is located at Edge-TP and another copy bearing only [Q] appears at the edge of CP. Based on the feature-split analysis, I suggest the following definition:

(25) An category at an A-position is reanalyzed as a category bearing phi-features, while a category at an A'-position is reanalyzed as a category lacking phi-features.

The definition in (25) is distinct from Chomsky's (1981) and Chomsky and Lasnik's (1995) type of definitions in that it recaptures position types in terms of phi-features without appeal to non-featural position-typing stipulations. Another conceptual advantage is that the current definition no longer struggles with answering why position types matter in the CI representation under binary compositional interpretation of categories. Rather, it is preferable that interpretable phi-features on N (or Q-features), which are all sent to the CI interface, not to the sensory-motor (SM) interface, play important roles for interpretation. Especially, traditional A/A'-positions behave differently with respect to binding phenomena, which are presumably computed in the semantic component as discussed in Chomsky (1995a). That is, it seems conceptually more plausible that the presence/absence of phi-features serves as the crucial factor in distinguishing the two kinds of positions. Also, in an interface representation, we cannot see that Edge-TP was earlier created by an Internal Merge operation applied in the narrow syntax in which T, then bearing uninterpretable features which are crucially absent in the convergent CI representation, probed DP. That is, recapturing A/A'-distinctions in terms of features on goals, but not in terms of probes, enables the CI interface to detect the A/A'-status of a certain element without appeal to features that were present in the narrow syntax but which are absent in the CI representation. This is a crucial difference between our approach based on phi-features on goals and Chomsky's (2007) approach appealing to features on the probe triggering Internal Merge.[22]

Furthermore, Lasnik and Stowell's (1991) weakest crossover effects lend empirical support to the analysis proposed:

(26) *Mary asked me who to persuade his mother to vouch for?
(Mary asked me [who$_1$ [PRO to persuade [his$_1$ mother]$_2$ [PRO$_2$ to vouch for t$_1$]]])

(27) Who should be easy to persuade his mother to vouch for?
(Who$_1$ should be easy [OP$_1$ [PRO to persuade [his$_1$ mother]$_2$ [PRO$_2$ to vouch for t$_1$]]])

(Lasnik and Stowell, 1991: 703)

As (26) and (27) show, null operator movement (if it is exists) does not cause weak crossover effects in (27) in contrast to (26), where the lexical wh-phrase (i.e. a true quantifier) does. Although weak crossover effects are widely used as a diagnostic for A'-movement, Lasnik and Stowell's findings imply that it is descriptively inadequate to say that weak crossover is caused by a certain configuration of A'-binding. The weakest crossover effects instead suggest that only a true quantifier, which can be defined as a category bearing a quantifier, not merely an A'-position, causes weak crossover effects. If so, the CI system needs to see features inside a category to determine whether an A'-binder is a true quantifier or not. (See also Koopman and Sportiche, 1983, fn.17 regarding topicalization of a proper name and Chomsky, 1982 fn.11 regarding relative pronouns (non-quantifiers) suppressing Bijection (WCO) violations). These facts seem to support

[22] The proposed approach based on presence of phi-features predicts that the edge of the embedded CP is an A-position in a sentence like "*Who seems it is likely to leave?", which I discussed in Section 3.2. [uCase] on "who" is not valued in the embedded clause, so that features on "who" are not split off before moving to the edge of the embedded CP. That is, "who" lands at the edge of the embedded CP still bearing phi-features. Although this derivation never converges because [uCase] on the copies of "who" in the embedded TP are transferred causing crush (following Nevins' (2005) analysis of improper movement), this nonetheless seems to be an interesting prediction made by the current analysis. At this point, however, I do not know of any way to test it.

the current hypothesis that A/A'-properties should be at least in part defined categorially/featurally.

3.3.8 Interim Summary

In Section 3.3, I have shown how the feature-split analysis rules out improper movement and explains A'-opacity effects, whereby elements at A'-positions are invisible to Agree between T and a subject. Moreover, feature-splitting, which is compatible with Chomsky's (1964, 1995a) hypothesis that wh-phrases like "who" are morphologically complex, [wh+something] consisting of a "Q-part" and a "non-Q-part", provides us with a new possibility for capturing the A/A'-distinction in terms of features of categories. In Section 3.4, I will extend the discussion to the Bantu language Kilega and consider potential problems the analysis confronts. Morphosyntactic variation regarding feature-splitting and, as a result, the parametrization of (i) A'-opacity effects and (ii) "proper improper movement" are discussed.

3.4 Extension to Improper Movement in Bantu Languages

3.4.1 Brief Review and Problems

Remember that one of the crucial motivations for the C-to-T feature inheritance system in Chomsky (2007, 2008), which the current system is based on, is that C cannot keep its inherent [uF] in a convergent derivation. Rather, [uF] has to be removed from C and is assumed to be inherited by T. As a result of the inheritance, T attracts features which it agreed with, while C attracts the rest of the features purely by EF (i.e. without Agree). That is, the prediction is:

(28) C never directly enters into a Phi-Agree relation with a goal because C lacks [uF], given feature-inheritance.

One problem the feature-inheritance analysis must address then is the apparent phenomenon of C-agreement which appears inconsistent with (28).

Second, concerning A'-opacity, our feature-split analysis predicts that a DP copy at a phase-edge lacks phi-features, thus explaining its invisibility to phi-agreement (as was discussed in Section 3.3.6). This mechanism predicts:

(29) Once an element is attracted to a phase-edge, the element can never be the GOAL of phi-agreement probing.

This too as we will see is not universally true. In the next section, I will explore these potential problems (Case, and DPs on an edge position serving as goals of phi-agreement probes). A solution based on a natural, though descriptive morpho-syntactic parameter is proposed.

3.4.2 Potential Problems: C-Agreement Languages

In this section, I consider Kilega, which is a Bantu language which appears to violate predictions made by the current analysis that: (i) a phase head never acts as a direct probe in Agree and (ii) an element at a phase-edge never serves as the goal of phi-agreeing probe. Consider the following wh-constructions in Kilega:[23]

(30) a. Bábo bíkulu b-á-kás-íl-é mwámí bíkí mu-mwílo?
2that 2women 2SA-A-give-PERF-FV 1chief 8what 18-3village
"What did those women give the chief in the village?"

b. Bíkí bi-á-kás-íl-é bábo bíkulu mwámí mu-mwílo?
8what 8CA-A-give-PERF-FV 2that 2woman 1chef 18-3village
"What did those women give the chief in the village?"

(from Carstens 2005: 220)

[23] The abbreviations used in (30) are: SA: Subject Agreement, CA: Complementizer Agreement, A: Kilega vowel /a/, FV: final vowel of Bantu verbs, PERF: perfect tense, Arabic numerals: noun class.

Kilega wh-constructions allow both in-situ wh-questions as in (30a) and overt wh-fronting as in (30b). When the wh-phrase "bíkí" remains in-situ in (30a), the verb agrees with the subject "bábo bíkulu" and bears the subject agreement morpheme. These two categories also agree in noun class as indicated by the number "2". In contrast, when the wh-phrase overtly moves as exemplified in (30b), the verb moves to C and agrees with the fronted wh-phrase. As a result of this agreement, the verb gets the complementizer agreement morpheme and also the fronted wh-phrase and C share the same noun class number "8". Let us see how Carstens (2005) derives (30b):

(31) (= (30b))

a. $C_{[uPhi]}$ $[_{TP}$ $T_{[uPhi]}$ $[_{vP}$ $\text{what}_{[phi]}$ $[_{vP}$ $\text{Subj}_{[phi]}$ $[_{VP}$ . . . (Obj) t_{what}]]]]

MOVE

PHI-AGREE

"What" moves to the edge of vP and then undergoes phi-agreement with T.

b. $C_{[uPhi]}$ $[_{TP}$ $\text{what}_{[phi]}$ $[_{vP}$ t_{what} $[_{vP}$ Subj $[_{VP}$ . . .]]]]]

MOVE

PHI-AGREE

"What" moves to Edge-TP and then undergoes phi-agreement with C.

c. $[_{CP}$ what C $[_{TP}$ t_{what} $[_{vP}$ t_{what} $[_{vP}$ Subj $[_{VP}$ . . .]]]]]

MOVE

"What" moves to the edge of CP.

In (31a), the object "what" is attracted to the edge of vP after its accusative Case is valued. Then, T phi-agrees with this fronted "what". In (31b), EF on T attracts "what" to Edge-TP and [uPhi] on C agrees with "what", i.e. C-agreement. That is, C appears to retain phi-features--they are not inherited by T—and C directly probes "what" as its goal. Finally in (31c), EF on C attracts the wh-phrase. Carstens (2005) assumes that Case-valuation in Kilega takes place independently of phi-valuation, unlike Chomsky (2007, 2008). That is, in (31a), T phi-agrees with the fronted wh-phrase at the edge of vP but Case is assigned to the subject, which remains at the lower edge of vP.

The predictions in (28) and (29) are therefore false for Kilega wh-constructions. The current system predicts that C never acts directly as a phi-probe because [uF] cannot be stranded on C but must be inherited by T. However, in Kilega, C serves as a probe in phi-agreement. Also, contra (27) the wh-phrase fronted to the edge of vP does indeed undergo phi-agreement serving as the goal of the probe T and also the goal of C in Kilega. If the same feature-split as in English occurred in Kilega, [Phi] on the wh-phrase at the edge of vP should already be split off, so that the wh-phrase at the edge lacks phi-features. That is, [uPhi] on C and T should never be valued, contrary to fact.

In the next section, I will suggest that feature-splitting is, in fact, parameterized between English-type and Kilega-type I-languages. Also, I will demonstrate that this parameterization predicts <u>permissible</u> improper movement in Kilega and <u>lack of</u> A'-opacity (i.e. A'-transparency) in agreement patterns, and so further supports the feature-split analysis.

3.4.3 How to Split Features: The Separation of [Phi] and [Case]

As explicated in the last section, the feature-split system has potential problems, namely (28) and (29) making incorrect predictions for languages such as Kilega. There are at least two issues which seem to be incompatible with our analysis: **[Problem1]** the Kilega wh-phrase retains phi-features at and beyond the edge, and **[Problem2]** a C head in Kilega retains [uPhi], which triggers C-agreement with a wh-phrase, contra Richards and Chomsky's feature-inheritance analysis which predicts that valued [uF] can never remain on the edge. In this section, I will consider how to accommodate these incompatibilities within the system proposed in Section 3.3.

3.4.3.1 [Problem1]: Phi-features Moved to Phase-Edge Positions

Regarding Problem 1, the question is how the system makes it possible that an element moving to a phase edge keeps its phi-features. In the previous sections, I assumed, following Chomsky (2007, 2008) that, in the case of C-to-T feature inheritance, T attracts only features which it agreed with and C attracts the rest of the features purely by EF (see (24)). Since Chomsky (2000), Case-valuation has been considered a reflex of phi-feature valuation between T and DP. Therefore, features which participated in, or features valued as a result of, Agree with T are both [iPhi] on T and [uCase] on DP. Given these assumptions, [iPhi] and [uCase] on DP (the features participating in Agree with T) are attracted to Edge-TP while [Q] is attracted to Edge-CP. Recall feature-split was crucial to explain improper movement phenomena as agreement failure (see Section 3.3). Now, let us consider what features can participate in Agree with T in Kilega. Carstens (2005) suggests that Kilega Case-valuation is not tied to phi-agreement (contra Chomsky 2001), as exemplified in (32)-(33):[24]

(32) **Ku**-Lúgushwá **kú**-kili **ku**-á-twag-a **nzogu** maswá
17- **17**SA-be.still **17**SA-A-stampede-FV **10elephant** 6farm
"At Lugushwa, elephants are still stampeding over (the) farms."

(33) a. **Mutu** t-**á**-ku-sol-ág-á **muku** wénené
1person NEG-**1SA**-PROG-drunk-HAB-FV **6beer** alone
"A person does not usually drink beer alone."

b. **Maku** ta-**má**-ku-sol-ág-á **mutu** wéneéné
6beer NEG-**6SA**-PROG-drink-HAB-FV **1person** alone
"No one usually drinks beer alone."

(from Carstens 2005: 265)

In (32), the subject is “elephants”, whose noun class is marked with "10". This element should be assigned nominative Case by T. However, the verb “stampede” and the

[24] Each of the abbreviation used in (32) and (33) stands for: SA: Subject Agreement, A: Kilega vowel /a/, FV: final vowel of Bantu verbs, PERF: perfect tense, NEG: negation, PROG: progressive, HAB: habitual, Arabic numerals: noun class.

auxiliary “be still” on T bear the same noun class markers as the sentence-initial DP “Lugushwa”, which is marked with "17". That is, T in this example assigns Case to the subject “elephant” but [uPhi] on T is valued by the other DP “Lugushwa”. Chomsky’s analysis, whereby Case-valuation on DP is universally a reflex of phi-agreement, incorrectly predicts that T and the subject “elephants” are both marked with "10". The same contrast can be observed in (33). If the subject “person” is placed at Edge-TP as in (33a), T gets the marker "1" from the subject. If the subject stays in the post-verbal position as in (33b), the other DP “beer” at Edge-TP assigns its noun class marker "6" to T and the subject “person” keeps the other marker "1". That is, these Kilega data suggest that phi-valuation does not universally coincide with Case-valuation. Based on these kinds of data, Carstens (2005) concludes that Kilega Case-valuation takes place independently of phi-agreement, unlike in English.

Carstens’ analysis provides us with a new possibility in feature-splitting. Under Chomsky’s system of Case-valuation, a “single” Agree operation values both Phi and Case features. But in Kilega, a single Agree operation involves only Phi-agreement or only Case-valuation. In other words, T triggers two independent Agree operations: one is for [Phi] and the other is for [Case].[25] What do these differences predict regarding feature-split? In English, both [Case] and [Phi] are involved in e.g. T-DP Agree at once.

[25] One might ask how T triggers Case-valuation independently without T bearing [uCase]. Under the current system, [uF] on a phase head drives Agree, but if T has [uCase] to execute Case-valuation, it is not clear how “valuation” takes place between [uCase] on T and [uCase] on DP because neither of them has a value. (See Epstein and Seely 2006.) How to explain Kilega phi-independent Case-valuation under the current minimalist approach is beyond the scope of this thesis but for the sake of discussion, I will tentatively assume that the Kilega Case-feature system has something similar to wh-phrases in Chomsky (1995b) and Grewendorf (2001) in terms of feature distributions, where a wh-phrase has both [uWh] and [iQ]. That is, Kilega Case-assigning heads are equipped with [iCase] and [uKase] and DP bears [uCase] and [iKase], so that [uKase] on T triggers Agree with [iKase] on DP and as its consequence, [uCase] on DP is valued independently of phi-valuation. Again, this is just a tentative approach to capture Carstens’ data and the precise parametric difference between English and Kilega Tense mophosyntax under the current feature-valuation system. Further analysis is required.

Therefore, I assume that these two features cannot be separated after Agree, so that they are "fused" in English-type languages. (See Section 3.3 for more detail.) As a result, EF on T only has the option to attract this “inseparable” feature set. In Kilega, on the other hand, [Phi] and [Case] never combine with each other because Agree applies to them separately. How then does the Kilega EF on T decide what features to attract? There seem to be two options, in contrast to English:

(34) Kilega
  a. Option1: T/V attracts both [iPhi] and valued [uCase] on DP.
  b. Option2: T/V attracts either (i) [iPhi] or (ii) valued [uCase] on DP.

The first option is available both in English-type languages and Kilega-type languages. Although T in Kilega agrees with [iPhi] and values [uCase] in a different manner from English, it is obvious that both [iPhi] and [uCase] are finally involved in valuation by T. In addition, Kilega has the second option because of its unique system of Case versus Phi-valuation. As mentioned above, [Phi] and [Case] are never amalgamated in Kilega, unlike English, so that EF on T can choose either of the features which it agreed with. If [iPhi] is attracted by T, C attracts valued [uCase]. If valued [uCase] is attracted by T, on the other hand, C attracts [iPhi]. These three possibilities are logically available for feature-splitting. However, it turns out that two of the three options, in fact, deducibly cause crash.

For Option1 (= (34a)), this is the same type of feature-splitting as in English-type languages. But remember that in Kilega the wh-phrase moved to the edge of vP undergoes phi-agreement with T and C in the derivation illustrated in (31). Thus, if Option1 is chosen, [uPhi] on T or C is never valued, causing crash just the same as in English. This option never converges in Kilega. Similarly, in one of the two possibilities in Option2 (= (34b(i)), whereby valued [uCase] (and [Q]) is attracted by C and [iPhi] is attracted by T, the derivation crashes because the appearance of valued [uF] at a phase-edge causes crash at the next phase (as discussed in Section 3.2).

On the other hand, the other possibility with Option2 (= (34b(ii)) converges in that a copy at the phase-edge has only [iF], namely [iPhi] (and [iQ]). Under this option, a moving element succeeds in carrying its [iPhi] along with its [iQ] to the phase edge. When this option is taken, the Kilega derivation can converge. Notice that a universal Activity Condition would incorrectly predict crash. That is, the Activity Condition incorrectly prevents a higher T from getting its phi-features valued by a wh-phrase on the edge of CP since the wh-phrase lacks Case and so is inactive. Following Carstens' (2005) analysis of Case-valuation as an independent operation, T/C in Kilega induces a slightly different type of feature-split from the English-type, enabling an element moved to a phase-edge to keep its [iPhi] rendering it available as the goal of phi-agreement with a higher probe. On the other hand, the English-type always chooses Option1 as discussed in Section 3.3. Thus, an element moved to a phase edge invariably loses its [iPhi], so that in cannot serve as a matching goal of a higher phi-probe, causing crash because of unvalued [uPhi] on the higher probe.

3.4.3.2 [Problem2]: [uPhi] on C

Next, consider Problem2: a C head in Kilega retains its inherent [uPhi], which enters into direct C-agreement probing with a wh-phrase. As we have seen in Section 3.2, valued [uF] staying on a phase head causes crash of the derivation, following Chomsky-Richards. However, C in Kilega directly enters into phi-agreement with a goal wh-phrase and they share the same noun class marker. That is, C itself is a probe and bears valued [uPhi] as a result of agreement with a goal. Thus, C seems not to transfer its phi-features to T. (See Ouali 2006 for important discussion regarding possibilities of stranded [uF] at a phase head in Berber.) In this sense, Kilega C-agreement is incompatible with Chomsky-Richards. The point to be clarified is what kind of mechanism allows valued [uPhi] on C to converge. With respect to this problem, I suggest the following two kinds of EF:

(35) Parameter: Two Types of EF
EF
i. EF: pure edge-feature
ii. EF_{phi}: phi-edge-feature

The first type (i) is "Pure" Edge-Feature, which triggers internal merge without Agree. English-type languages take this type of EF as a value of the parameter. The second type (ii) is "Phi" Edge-Feature, where EF contains [uPhi] inside. That is, phi-features are part of EF. [uPhi] within EF triggers phi-agreement with a goal bearing [iPhi], then EF triggers Internal Merge of the element which it agreed with. In other words, this type of EF triggers phi-agreement as well as internal merge. Kilega-type languages choose this type. (See Baker and Collins 2006 for a similar view.)

One might wonder why, under the Chomsky-Richards analysis, EF can be at a phase head, hence on the edge, despite the fact that it is uninterpretable. Regarding this issue, Chomsky (2007) argues:

"As an uninterpretable feature, EF cannot reach the interface, so presumably deletion of EF is an automatic part of the operation of Transfer. Note that the same cannot be assumed for the standard uninterpretable features, which can be deleted only when certain structural conditions are satisfied, and will crash the derivation otherwise."

Chomsky (2007:11 fn.16)

EF is neither phonological nor semantic, so that regardless of it being valued or unvalued it should (but somehow does not) cause crash.[26] That is, EF has to disappear once it is satisfied in order to allow convergence. Given this argument, EF_{phi} in Kilega-type is automatically deleted by Transfer including the valued [uPhi] within the automatically deleted EF, so that the derivation never crashes. Thus, Kilega phi-features, by being embedded as a part of EF, get deleted automatically as a free-rider in EF auto-deletion.[27]

Summarizing so far, I suggested that there are two types of EF: pure-EF and phi-EF. That is, EF is parameterized and equipped with those two values. English-type chooses the former, while Kilega-type chooses the latter. Although EF is [uF], it is automatically deleted by the Transfer operation because of its different property from other [uF]. Therefore, EF_{phi}, as [uF], triggers phi-agreement with a wh-phrase, yet still undergoes automatic Transfer. This parametrization enables us to "solve" Problem 2: a C head in Kilega has valued [uPhi] under direct C-agreement with a wh-phrase. Under this proposal, C in Kilega can trigger phi-agreement and bear valued [uPhi] at the edge, yet can do so without causing crash.[28]

In this section, I have addressed the potential problemsfor Chomsky/Richrads' deduction presented by C-agreement languages such as Kilega. The following table summarizes what I have claimed in this section and identifies two predictions that I will discuss in the next section:

---

[26] This is the current incarnation of the mysterious EPP (See e.g. Epstein, Seely and Pires 2005 and Epstein and Seely 2006 and the many references cited for discussion.)

[27] One potential problem with this "free-rider" approach is that there is a possibility that unvalued [uPhi] on C can be deleted i.e. entirely oblitenated as part of EF without ever getting valued. It predicts that C-agreement in Kilega is optional because unvalued [uPhi] on C does not cause crash. Since it is entirely removed and never reaches the interface (just like EF in which it is embedded). A possible answer to this problem is that while [uPhi] on C can be automatically deleted along with EF regardless of being valued or not, but it nonetheless leaves [Phon] on C unspecified, which is not interpretable at the SM interface. Therefore, [uPhi] on C has to agree with a DP (= C-agreement) to specify [Phon] on C. Otherwise, the derivation crashes.

[28] I admit that embedding phi-features within EF is at best a descriptive parameter (if correct and SMT-consistent).

(36)

	EF	[Case & Phi] separability	Improper Movement	A'-Opacity
English	Pure EF	NOT separable	DISALLOWED	YES
Kilega	Phi EF	Separable	ALLOWED	NO

The parameters developed in this section enable us to capture the differences between English-type and Kilega-type especially regarding C-agreement. Now, the proposed parameters also make strong predictions with respect to both Kilega improper movement and A'-opacity, as illustrated in TABLE (36). The current system allows an element at a phase-edge to bear [iPhi] in Kilega unlike English which invariably splits off [iPhi & uCase] internally-merging to the edges of VP/TP. That is, improper movement should be "proper" movement in Kilega i.e. the goal can undergo Internal Merge to the phase edge and keep its phi-features, which in turn allow an upstairs T to agree with a phi-bearing wh-phrase in the edge of CP. In addition, A'-opacity effects should not be observed--an Kilega element at an A'-position (phase-edge) itself bears phi-features unlike English feature-split and so should block phi-agreement across it. In the next two sections, I argue that these predictions are supported by the data.

3.4.3.3 More Empirical Consequences: A'-Transparency Effects

First, let us examine the predicted suppression of A'-opacity effects in Kilega. In Section 3.3.1, I discussed Rezac's (2003) A'-opacity effects, whereby an element at an A'-position does not block Agree in A-positions, and showed that our feature-split analysis explains this. In English, an element moved to a phase-edge e.g. Edge-vP does not bear [iPhi], so does not block phi-agreement between a probe T and a goal subject despite being c-commanded by the probe and c-commanding the goal (= intervention). In contrast, the current system exemplified in Section 4 makes it obligatory that an element moved to a phase-edge in Kilega retains its [iPhi]. This predicts that Kilega does NOT show A'-opacity effects, rather it should show A'-transparency effects, whereby intervening A'-elements <u>do</u> block Phi-agreement between A-positions. This prediction is confirmed by the following data presented in Carstens (2005):

(37) *Bíkí **bi-b**-á-kás-íl-é bábo bíkulu mwámí mu-mwílo?
8what **<u>8CA-2SA</u>**-A-give-PERF-FV 2that 2woman 1chef 18-3village
"What did those women give the chief in the village?"
(Carstens, 2005:237, cited from Kinyalolo, 1991)

(38) $C_{EF(Phi)}$ [$_{TP}$ T$_{[uPhi]}$ [$_{vP}$ what$_{[phi]}$ [$_{vP}$ Subj [$_{VP}$ . . .]]]]]
AGREE

(37) has the same base form as (30) but the verb in (37) contains the morphemes both for subject agreement and for C-agreement. This data indicates that those two agreement morphemes cannot co-occur in a single sentence. Either of them is possible but the two never co-occur at once.[29] Why? As illustrated in (38), "what", which was attracted from the object position, is at the edge of vP intervening between T and the subject "those women". But notice that in Kilega [iPhi] on DP is attracted by a phase head (v in this case) along with [Q]. That is, the intervening "what" retains [iPhi], unlike in English-type languages. The [uPhi] on T then agrees with the closer [iPhi], that is, "what", but not the subject. Therefore, [uCase] on the subject is never valued causing featural crash. This is why the sentence in (37) is ruled out.

A'-transparency/opacity effects can be explained by the proposed slight parameterization of feature-splitting operations and the parameter concerns features of

[29] As discussed in Den Dikken (2001), however, if the subject were a pronoun in (37), then the verb can undergo both Subject Agreement and C-Agreement. This may well be due to the possible incorporation of pronominals. (See Baker 1988, Bošković 1997 and discussion of Bošković 1997 in Epstein and Seely 2006:83 for related phenomena concerning pronominal/non-pronominal Case-Agree asymmetries.)

functional categories (not: N, V, A, P) as restrictively postulated by Borer (1984) and Fukui (1986).

3.4.3.4 "Proper" Improper Movement

The second prediction made by the current analysis is that Kilega should allow improper movement. In the derivation of English-type improper movement, recall [uPhi] on upstairs finite T is not valued because a moving element, which is the only potential goal available for T, lacks [iPhi] by virtue of feature-split. In Kilega, on the other hand, moving elements retain their inherent [iPhi] throughout the derivation. That is, the prediction is that Kilega-type feature-split renders a moving element available as the goal of phi-agreement by a probing upstairs T and improper movement thought to be universally excluded should be permissible.

Consider again the derivation for Kilega C-agreement illustrated in (31). Movement of "what", in fact, occupies the following four positions traditionally (see Chomsky 1981) characterized as follows: Object position (A) → Edge-vP (A') → Edge-TP (A) → Edge-CP (A'). The steps (i.e. A→A'→A) have been traditionally considered improper movement, which is not permissible in English-type languages. But in Kilega, the movement through these positions is allowed and the derivation in (31), in fact, converges.

In addition, another Bantu language Lussamia, which shows the same kind of C-agreement as Kilega, similarly allows improper movement. In the following example, the embedded subject "Ouma" gets nominative Case from the embedded T whose features are by assumption inherited from the embedded C. (Therefore, CP must be present in these representations.)

(39) Lusaamia
   a. Bi-bonekhana Ouma a-bwereo
      8SA-appear Ouma 3SA-leave
      "It appears that Ouma left"
   b. Ouma a-bonekhana a-bwereo
      Ouma 3SA-appear 3SA-leave
      "Ouma appears as if he left."

(from Carstens 2008)

The subject "Ouma" has the noun class marker "3". In (39a), the embedded verb "leave" is marked with "3" but the matrix verb "appear" is marked with "8". This means that "Ouma" stays in the embedded clause and agrees only with the embedded T. In (39b), on the other hand, both of the verbs are marked with "3", which is assigned by the subject "Ouma". That is, "Ouma" first agrees also with the embedded T. Then, it moves to the matrix clause and agrees with the matrix T. The subject movement ends up landing at the matrix Edge-TP. The latter case is the same type of raising case as the English data as "*John seems (that) is intelligent". The Lussamia sentence (39b) is grammatical, in contrast to the ungrammaticality of the English analog.[30]

As expected, improper movement is "proper" movement in Bantu languages such as Kilega and Lussamia. That is, these data suggest that improper movement, which concerns Agree between morphological features under our approach, is in fact parameterized as is the morphology of these languages. In this sense, the current system can be distinguished from previous approaches, where improper movement is universally excluded, while also indicating a connection between (un)availability of C-agreement and the permissibility of improper movement and A'-opacity/transparency.[31]

[30] In fact, Section 3.6 will argue that the Kilega/Lussamia-type derivation occurs even in English, namely in *tough*-constructions, which are notorious for their display of seemingly improper movement.

[31] Dutch is, in fact, a language between English-type, which disallows improper movement and does not

3.4.3.5 Remaining Issues: Is Kilega and Lusaamia Wh-Movement A- Movement?

Here, I revisit the A/A'-distinction issue again, which was discussed in Section 3.3.7. The feature-split analysis enables us to recapture A/A'-position types in terms of features of categories. In the English-type derivation, a copy appearing at a phase-edge position lacks [iPhi] due to feature-split. In the previous section, I suggested: A category at an A-position is reanalyzed as a category with phi-features, while a category at an A'-position is reanalyzed as a category without phi-features. This category/feature-based distinction makes a prediction that phase-edge positions such as Edge-CP bearing phi-features are A-positions in Kilega and Lusaamia. This also entails that wh-movement in Kilega and Lusaamia is, actually, A-movement, in contrast to English-type wh-movement construed as A'-movement. With respect to this issue, Vicki Carstens (p.c.) reports that Kilega and Lusaamia informants show reconstruction in wh-movement but no weak crossover effects. The following data are from Lusaamia[32]:

(40) Ni esitabu sina khu **biaye*i*** bidatu si o-para mbwe **buri omwana**
COP 7book 7which LOC 8his 8three 7wh.agr 2S-think that **every 1child**
we esikuli*i* a-som-anga
1of 7school 3S-read-HAB
"Which among his 3 books do you think every student reads?"

(41) **Wina*i*** yi embwa **eyae*i*** i-ya-khera?
1who 1wh.agr 9dog **9his** 9SA-PRES-love
"Who does his dog love?"

In (40), the bound variable reading is available between "every student" and "his" in Lusaamia, like in English. This indicates that the wh-phrase is reconstructed to the

show C-agreement, and Kilega-type, which allows improper movement and has C-agreement. Dutch also shows C-agreement like in Kilega, but does not allow improper movement. The current system cannot capture this type of language straightforwardly. As for this issue, I will tentatively assume that Dutch also takes the English-type parameter value, so that ungrammaticality of improper movement is explained. With respect to C-agreement, I tentatively adopt Zwart's (2006) analysis, where C-agreement in Dutch can be explained without appeal to the Agree operation, and hence is argued not to be an agreement phenomenon in the formal sense.

[32] I am very grateful to Vicki Carstens for providing me with the Lusaamia data.

original position which "every student" can bind. In (41), on the other hand, "his" can receive bound variable interpretation in Lusaamia, unlike in English. That is, no weak crossover effect is observed in this language. The same behaviors are also reported in Kilega according to Carstens (p.c.). It seems premature to say based only on such weak crossover data that Kilega and Lusaamia wh-movement is A-movement. However, the data provides interesting independent reasons for believing that Kilega and Lusaamia wh-moved DPs, which bear phi-features, behave differently from English wh-moved DPs, which do not have phi-features, with respect to traditional A'-movement diagnostics. That is, it might be possible to consider that these differences usually attributed to position types are instead attributable to the presence vs. absence of phi-features on moved wh-phrases, although much further research is required to determine the adequacy of this proposal (=(25)) that A-positions are phi-bearing categories while A'-positions are categories without phi-features. (See Obata 2008c regarding the argument/adjunct distinction based on Case-features.)

3.5 Extension to Improper Movement in Japanese

This section further extends the proposed system to Japanese cases, which appears to be more similar to the Bantu-type than to the English-type.[33] Like in Kilega/Lusaamia, Japanese also permits improper movement. The following subsections present two kinds of those cases and demonstrate how they are explained in the current system. Furthermore, the proposed mechanism gives an interesting implication to mechanisms behind the clausemate condition observed in Japanese NPI-licensing.

---

[33] I thank Noam Chomsky for pointing out that Japanese might be a potential problem for the feature-split system, worthy of investigation.

3.5.1 Raising to Object in Japanese Moving through the Edge of CP

The first case of grammatical improper movement is the Raising-to-Object (RTO) phenomenon in Japanese discussed in Kuno (1976) and Tanaka (2002). The interesting point in Japanese RTO is that an embedded subject rises to an object position in the next higher clause across the CP-boundary whose head is phonologically realized as "to", which is a declarative complementizer in Japanese. Consider,

(42) a. John-ga [Bill-ga baka-da to] omot-teiru.
John-NOM [Bill-NOM fool-COP COMP] think-PROG
"John thinks that Bill is a fool."
b. John-ga Bill-o$_1$ [t$_1$ baka-da to] omot-teiru.
John-NOM Bill-ACC [fool-COP COMP] think-PROG
"John thinks of Bill as a fool."
(Tanaka 2002: 637)

These two sentences are logically equivalent. The only difference is that "Bill" is marked with nominative Case in (42a) but with accusative Case in (42b). As indicated in (42b), the complement subject "Bill" is assumed to be raised from the embedded clause <u>across CP</u>. Since this movement is string vacuous, one might wonder whether "Bill-o" in (42b) is really in the matrix clause. There are several arguments indicating that the complement subject is in the matrix clause, not in the embedded clause. One of them concerns A vs. A' scrambling. Consider the following data presented by Tanaka (2002):

(43) a. ??Otagai$_1$-no sensei-ga karera$_1$-o$_2$ [t$_2$ baka-da to]
Each other-GEN teacher-NOM they-ACC [fool-COP COMP]
omot-teiru.
think-PROG
"*Lit.* Each other's teachers think of them as fools."
b. Karera$_1$-o$_2$ otagai$_1$-no sensei-ga t$_2$ [t$_2$ baka-da to]
They-ACC each other-GEN teacher-NOM [fool-COP COMP]
omot-teiru.
think-PROG
(Tanaka 2002: 640)

(43a) is degraded because the reciprocal "otagai" is not bound by its antecedent "karera" violating Condition A. (43b), on the other hand, is grammatical. This is because the scrambled phrase “karera-o” binds the reciprocal satisfying Condition A. In other words, the scrambled phrase serves as an A-binder here. This tells us that “karera-o” underwent A-scrambling, which takes place only clause-internally. If the scrambled phrase moves directly from the embedded subject position, this would not be short/A-scrambling but long/A’-scrambling with result that "karera" could not A-bind the reciprocal. The data show that “karera-o” is in an A-position and so it short/A-scrambled from the object position in the matrix, not in one-fell swoop from the embedded subject position. This supports the RTO analysis. Although there are some arguments against the RTO analysis e.g. Saito’s (1983) Control analysis, this thesis (of course, tentatively) will assume that the RTO analysis is right.

Under Chomsky’s phase-based approach, the complement subject in the RTO context has to move through the edge of CP on its way to the object position in the matrix clause. Otherwise, the moved element "Bill-o" in (44) will be removed by Transfer of the embedded TP and consequently is incorrectly predicted to be unable to escape (move-out-of) the embedded TP.

(44) John-ga [$_{VP}$ Bill-o$_1$ [$_{CP}$ t$_1$ C [$_{TP}$ t$_1$ baka-da] to] omot]-teiru.
John-NOM Bill-ACC fool-COP COMP] think-PROG
“John thinks of Bill as a fool.”

(Tanaka 2002: 651)

As Tanaka (2002) also points out, “Bill-o” moves: Edge-TP (A) → Edge-CP (A’) → Edge-VP (A), which is improper movement in Chomsky’s (1973) sense. Tanaka (2002) noticing that the movement seems improper stipulates that the intermediate edge of CP is an A-position. Later in this section, I will propose a different account based on the feature-splitting system. This concludes the discussion of our first case of grammatical improper movement.

3.5.2 Hyper-Raising in Japanese

The second example of grammatical improper movement is the so-called "Hyper-Raising" phenomenon. Ura (1994) presents hyper (and super) raising data in various languages including Japanese.

(45) a. Karera-ga$_1$ kyoo-no kaigi-de (Mary-niyotte) [t_1 asita kuru to]
They-NOM today-GEN meeting-at (Mary-by) tomorrow come COMP
houkokus-are-ta .
report-PASS-PAST
"*Lit.* They$_1$ were reported by Mary at today's meeting that t_1 would come tomorrow. "

b. pro$_1$ kyoo-no kaigi-de [karera-ga$_1$ asita kuru to]
pro today-GEN meeting-at they-NOM tomorrow come COMP
houkokus-are-ta .
report-PASS-PAST
"It was reported at today's meeting that they would come tomorrow. "
(Ura 1994: 66)

In (45a), "karera-ga" moves to the subject of the matrix clause originating as the subject of the embedded clause.[34] Comparing (45a) with (45b), the two sentences are logically equivalent. One might wonder whether there is really movement of "karera-ga" in (45a) from the embedded subject position to the matrix subject position. Notice that the matrix clause is passive, so that the subject position is a non-theta position. Therefore, the only way for "karera-ga" to get a theta-role is for it to move from the embedded subject position, as Ura (1994) discusses. He also provides the following data to show that the moved phrase is in an A-position (i.e. Edge-TP):

[34] Also see Takezawa (2006) for relevant discussion for subject movement to Spec-TP in Japanese.

(46) Karera$_1$-ga$_2$ kyoo-no kaigi-de otagai$_1$-no sensei-ni
They-NOM today-GEN meeting-at each other-GEN teacher-to
[t$_2$ asita kuru to] houkokus-are-ta
tomorrow come COMP report-PASS-PAST.
"*Lit.* They$_{1,2}$ were reported to each other's$_1$ teacher at today's meeting that t$_2$ would come tomorrow."
(Ura 1994: 74)

Here, the moved phrase "karera-ga" can A-bind the reciprocal "otagai" satisfying Condition A. The data imply that the moved phrase underwent A-movement to Edge-TP. However, notice that the embedded clause in (45)/(46) is CP evidenced by the fact that the declarative COMP "to" exists, as it did in the RTO case in the last section. That is, the movement to Edge-TP must take place across the CP-boundary, given PIC. Ura (1994) assumes that the subject directly moves from the embedded Spec-IP to the matrix Spec-AgrsP for Case supposing that the embedded Spec-IP is not a Case-position. However, this analysis is inconsistent with current phase-based derivation. Under the phase-based approach, "karera-ga" first needs to move to the edge of the embedded CP to escape Transfer of the embedded TP. Then, it can enter into the computation in the matrix phase. That is, the same disucssion as the RTO case applies to the hyper-raising cases: "Karera-ga" moves: embedded Edge-TP (A) → Edge-CP (A') → matrix Edge-TP (A). Again, this is improper movement in the 'updated sense' of Chomsky (1973). This then constitutes our second case of grammatical improper movement.[35]

Section 3.5.1 and 3.5.2 showed that there are at least two cases of grammatical improper movement in Japanese. As in Kilega/Lusaamia, Japanese is also a language which does not exclude improper movement. In the next section, the account based on

[35] Although I focus only on these two types of A-movement across phase-boundary, Uchibori (2000) also demonstrates that long distance A-scrambling out of subjunctive CPs in Japanese is possible. I thank Shigeru Miyagawa for reminding me of Uchibori's analysis.

feature-splitting is proposed for these two cases in Japanese (RTO and hyper-raising) by appeal to the fact that the Japanese phi/Case valuation system has propeerties slightly, but imprtantly different from both English-type languages and Kilega/Lusaamia-type languages.

3.5.3 Feature-Splitting in Japanese

How is Japanese grammatical improper movement explained? Before going into further discussion about Japanese, let us recall the proposed account for English-type languages and Kilega/Lusaamia-type languages, comparing them with Japanese:

(47)

	EF	[Case & Phi] separability	Improper Movement	A'-Opacity
English	Pure EF	NOT separable	DISALLOWED	YES
Kilega	Phi EF	separable	ALLOWED	NO
Japanese	Pure EF	?	ALLOWED	?

In English, Case and Phi are NOT separable by Internal Merge, so that both features are attracted by a head (T or V) which agreed with them. As a result, only the operator feature of a wh-phrase can appear at the edge of phases. This is why improper movement causes featural crash of unvalued [uPhi] on the matrix T entailing that in English A-movement across the CP-boundary is not permissible. In Kilega/Lusaamia, Case-valuation is not a reflex of phi-agreement, so that Case and Phi are valued independently as exemplified in (32)/(33), where a single head such as T agrees with one element for Phi yet it agrees with another element for Case. This suggests that Phi and Case are separable by Internal Merge and phi-features on DP CAN move to a phase edge (i.e. Phi can move across phasal boundaries by later operations). Also in Kilega/Lusaamia, EF_{phi}

triggers C-agreement with a moving element bearing Phi in this language. As predicted by the proposed system, this language allows A-movement across i.e. out-of CP/vP, so that improper movement does not cause any agreement failure, unlike the English case.

What about Japanese? Japanese permits improper movement like in Kilega/Lusaamia as mentioned in the last sections. This implies that Phi can move across CP/vP, which makes possible A-movement across phasal boundaries. Like in Kilega/Lusaamia, do Phi-agreement and Case-valuation take place independently in Japanese? And if so, is this independent agreement a necessary and sufficient condition for a grammar to allow Case/Phi Feature-Splitting Internal Merge? In the next subsection, close consideration will be given to Japanese Phi-agreement and Case-valuation and I suggest that they are NOT independent of each other exactly like in English but rather the connection of Phi and Case is much "weaker" than the corresponding connection in the English case. Therefore, unlike English, Internal Merge can separate those two features like in Kilega/Lusaamia. This is why Japanese permits improper movement. In other words, I will come to the conclusion that Japanese is a third type of feature-splitting Case-Agree system, falling “between” English-type languages and Kilega/Lusaamia-type languages.

3.5.3.1 The Weak Connection between Phi and Case Valuation in Japanese

Under the proposed feature-splitting system, whether or not [Phi] and [Case] can be separated by Internal Merge and [Phi] can be merged to the phase edge alone is crucial for explaining the (im-)permissibility of improper movement. Comparing with English and Kilega, let us consider what happens in Phi/Case-agreement in Japanese.

Unlike in Kilega, phi-agreement in Japanese is not phonologically realized. English does not have “rich” overt morphology of phi-features compared with Kilega but Japanese is almost entirely "silent" with respect to phonetic realization of phi-features relevant to Case-valuation. Let us see how Phi and Case interact. First, when T agrees with a subject pronoun, how do the phi-features on the subject affect T? In other words, how are the valued phi-features realized on T?

(48) *English*

T			**SUBJ**
BE	am	←	I
	are	←	you/they/we
	is	←	he/she/it
DO	do	←	I/you/they/we
	does	←	he/she/it

(49) *Japanese*

T			**SUBJ**
SURU	suru	←	I/you/he/she/it/we/they
"do"			

In English, depending on what kinds of phi-features the subject has, T is realized in morpho-phonetically different forms. For example, the realization of "is" is affected by [person] and [number] while the realization of "are" is affected by [number]. Also, the do/does realization depends on [person] of the subject. That is, [uPhi] on T valued by the subject is morpho-phonetically realized. In Japanese, on the other hand, the realization of "suru" is insensitive to the values of phi-features assigned by the subject. Next, let us see how Case assigned by T/V/D (precisely, assigned as a consequence of agreement with T/V/D) is realized on the subject.

(50) *Case-Realization*

	ENGLISH		JAPANESE	
Nominative (T)	I	he	watasi-**ga** "I"	kare-**ga** "he"
Genitive (D)	my	his	watasi-**no** "my"	kare-**no** "his"
Accusative (V)	me	him	watasi-**o** "me"	kare-**o** "him"

In English, Case is realized in morphologically different free morphemes. Depending on which heads the pronouns agree with and what kinds of phi-features the pronouns have, the phonetic realization is totally different. If the pronoun which has [3rd person], [masculine] and [singular] agrees with V, it is realized as "him". If the pronoun bearing [1st person] and [singular] agrees with T, it is realized as "I". That is, Case realization in English pronouns is affected by types of heads (T/V/D) and also phi-features of the pronouns in English. In Japanese, on the other hand, the Case realization is affected by types of heads for sure (T "-ga"/V "-o"/D "-no") but phi-features on the pronouns seem to play no role. Whatever kinds of phi-features the pronoun has, the heads always assign the exact same invariant Case-morpheme. This observation can be also enforced by the fact that Japanese allows multiple nominative, as widely discussed in Ura (1994) and Hiraiwa (2002):

(51) Taro-**ga** [$_{TP}$ Hanako-**ga** me-**ga** waru-**ku**] kanji-rare-ta koto.
Taro-NOM Hanako-NOM eye-NOM bad-INF think-PASS-PST that
"(that) Taro thought that Hanako had a bad eyesight."

(52) T DP$_1$(Nom)… [$_{TP}$ DP$_2$(Nom) …… DP$_3$(Nom)… … V] …

(Hiraiwa 2002: 19)

Hiraiwa (2002) suggests that these three nominative Cases are valued simultaneously by the single T. Notice that these three DPs have different phi-features. "Taro" and "Hanako" are distinguished by gender, and also "me ("eye")" does not appear to have any gender. However, the single T agrees with all of them and nominative Case is valued on all three. Hiraiwa (2002) argues that the closest phi-features (i.e. DP_1) are assigned to the probe T under a locality condition. If that's true, the other two phi-features have no contribution to phi-agreement on T. These data presented in (48)-(50) imply that Japanese phi-features on DP have less contribution to phi-valuation and Case-valuation than English phi-features on DP.[36] Apparently, Kilega and Lusaamia behave quite differently from Japanese (and English). These languages have very rich morphological systems, so that phi-agreement is morpho-phonetically "observable" as presented in Section 3.4. Based on these observations, I propose the following that Japanese has two distinct ways of splitting features:

(53) Japanese Feature-Splitting:
Option1: T/V attracts both [iPhi] and valued [uCase] on DP.
Option 2: T/V attracts only valued [uCase] on DP.

Given Chomsky's (2007, 2008) distinction between A/A'-movement, T/V attracts features on DP which it agreed with. Given that Case-valuation is a reflex of phi-agreement in Chomsky (2000), I conclude that Japanese Case-valuation is also a reflex of phi-agreement. This is why Option1 is possible in that both [Case] and [Phi] are involved in agreement by T/V just like in English. On the other hand, how strongly [Phi] and [Case] are involved in T/V-agreement is different from the English cases, as discussed in this section. Compared to English, the involvement of phi-features in agreement by T/V is "weaker". In other words, [Case] and [Phi] are equally involved in agreement with T/V

[36] The "weak-phi" analysis presented here is similar in spirit to Miyagawa (2010) who suggests that in a discourse-configurational language such as Japanese, the phi-agreement operative in an agreement language such as English is replaced by focus/topic-agreement. Also, Saito (2007) proposes that Japanese Case valuation is not tied to phi-agreement (unlike in English) but rather Case is assigned/valued contextually. Although I propose in this thesis that Japanese Case-valuation is a reflex of phi-agreement following Chomsky's analysis, I would like to emphasize that the Case/Phi system in Japanese is nonetheless distinct from English in that Case-valuation in Japanese is tied to phi-agreement more weakly/less than in English. In this sense, the idea presented here is similar in spirit to Miyagawa's and Saito's lines of argument.

in English while in Japanese [Phi] has less of a contribution to the agreement than [Case] does. This is why Option2 allows Internal Merge in which only [uCase] is attracted by T/V. In this sense, we can say that [Phi] and [Case] are more strongly connected in English than in Japanese, so that English never allows [Phi] and [Case] to be separated by Internal Merge while Japanese "sometimes" does. The following table is summary of feature-splitting in English, Kilega and Japanese:

(54)

	[Phi]&[Case] connection
English	STONG
Japanese	WEAK
Kilega/Lusaamia	NO

(55)

	[Phi]&[Case] together are/can be attracted by T/V	[Case] can be independently attracted by T/V.	[Phi] can be independently attracted by T/V
English	YES	**NO**	NO
Japanese	YES	**YES**	NO
Kilega/Lusaamia	YES	**YES**	YES*

Notice that Kilega/Lusaamia T/V can attract only [Phi] and allows valued [uCase] moves to the edge of a phase as marked by * in the above table. This option is logically possible but is excluded for an independent reason: valued [uCase] at a phase edge causes crash under Richards-Chomsky's Value-Transfer simultaneity system as already discussed in 3.4. Also, English does NOT allow: only [Case] on DP is attracted by T while the remaining features including [Phi], move to the phase edge, (as highlighted above). This is why only English does not allow improper movement since [Phi] cannot be split off and internally-merged to e.g. the edge of CP, there is invariably featural crash by unvalued [uPhi] on "upstairs" T. In other two language-types, an element moving to the edge can retain [Phi], so that no agreement failure results in the higher clause i.e. upstairs T can find a matching phi-goal on the edge under Minimal Search. Therefore, the feature-splitting system can capture these cross-linguistic variations in improper movement by appeal to slightly different morpho-phonetic properties. In addition, it is worth mentioning that <u>different "constructions"</u> observed in each of the particular languages are now recaptured as <u>slight one-feature differences</u>, which trigger different ways of feature-splitting.

In the next section, I show how the derivations are carried out to generate grammatical improper movement examples in Japanese.

3.5.3.2 Derivations

As discussed in the earlier sections, under the system proposed here, A-A'-A movement is not regarded as ungrammatical in itself. Rather, improper movement is captured as local agreement failure causing featural crash. That is, even if A-A'-A movement takes place, the representation is not excluded unless agreement failure results. This is why A-A'-A movement in Kilega/Lusaamia is allowed i.e. it does not cause agreement failure, since [Phi] can be split off and moved to the edge by Internal Merge in this I-language.

Now, let us apply the proposed system to the Japanese cases. First, consider Japanese hyper-raising phenomena. The data in (45a) is repeated here as (56):

(56) Karera-ga$_1$ kyoo-no kaigi-de [t$_1$ asita kuru to] houkokus-are-ta .
They-NOM today-GEN meeting-at tomorrow come COMP report-PASS-PAST
"*Lit.* They$_1$ were reported at today's meeting that t$_1$ would come tomorrow."

Here, the embedded subject "karera-ga" is moved to the matrix Edge-TP across the CP boundary. How is this grammatical sentence derived?

(57) ***Step1:*** $[_{vP}$ karera-ga asita $[_{VP}$ kuru] v]
Transfer

V is an intransitive verb. VP is transferred.
The derivation proceeds to the next higher (CP) phase.

Step2: $[_{CP}$ $[_{TP}$ $[_{vP}$ **karera-ga**$_{[iPhi][uCase]}$] T$_{[uPhi]}$] to(C)]

After [uPhi] is inherited from C to T, T phi-agrees with the subject.

Step3: $[_{CP}$ **karera-ga**$_{\mathbf{[iPhi]}}$ $[_{TP}$ **karera-ga**$_{\mathbf{[uCase]}}$ $[_{vP}$ **t**$_{[iPhi][uCase]}$] T$_{[uPhi]}$] to(C)]

The subject is attracted by EF on T and C simultaneously.
By feature-splitting, [iPhi] moves to the edge of CP and valued [uCase] moves to the edge of TP.

Step4: $[_{CP}$ **karera-ga**$_{\mathbf{[iPhi]}}$ $[_{TP}$] to(C)]
Transfer

TP is transferred.

Step5: $[_{vP}$ **karera-ga**$_{\mathbf{[iPhi]}}$ $[_{VP}$ $[_{CP}$ **t** to(C)] houkoku(V)] –sare(v)][37]

After the matrix vP phase is introduced, EF on v attracts "karera-ga" now occupying the embedded edge of CP (without agreement by V since it is passive).

Step6: $[_{vP}$ **karera-ga**$_{\mathbf{[iPhi]}}$ $[_{VP}$ $[_{CP}$ **t** to(C)] houkoku(V)] –sare(v)]
Transfer

VP is transferred.

Step7: $[_{CP}$ $[_{TP}$ $[_{vP}$ **karera-ga**$_{\mathbf{[iPhi]}}$ -sare(v)] –ta (T$_{[iPhi]}$)] C]

After [uPhi] is inherited from the matrix C by T, T phi-agrees with "karera-ga" in the matrix edge of vP.

Step8: $[_{CP}$ $[_{TP}$ **karera-ga**$_{\mathbf{[iPhi]}}$ $[_{vP}$ **t** -sare(v)] –ta (T$_{[iPhi]}$)] C]

EF on T attracts "karera-ga" to its edge position.

---

[37] I will omit adverbs in the matrix clause because of space reasons.

When "karera-ga" undergoes Internal Merge by T and C in the embedded clause at Step 3, only valued [uCase] moves to Edge-TP while [iPhi] moves to the edge of CP. Then, in the matrix clause, the moving element, retaining its [Phi], undergoes (is the goal of) phi-agreement with [uPhi] on probe T as in Step 7 and is attracted to the edge of TP as in Step 8. Notice that at Step 7, the matrix T agrees with the moving element “karera-ga” now bearing only [iPhi]. If the Activity Condition were adopted, there would be no way for T to agree with “karera-ga” because “karera-ga” is already inactive since its [uCase] was split off in the embedded clause (in order to prevent syntactically valued features from reaching the edge, as this is prohibited by Chomsky/Richards’ analysis. See relevant discussion in Section 3.3). Finally, the derivation converges. Like in Kilega (but unlike in English), no agreement failure happens in this case. Therefore, the sentence can be derived.

One might wonder what happens if [iPhi] is split off to the edge of embedded TP at Step 3, which is a splitting option allowed by Japanese.[38] The answer is that the same thing as English improper movement crash happens. That is, the element moved to the edge without [iPhi] cannot value [uPhi] on the matrix T, causing featural crash. There is, however, no undergeneration i.e. while this derivation of hyper-raising crashes, Japanese has a way of generating convergent hyper-raising as shown in (57). In addition, as discussed in Section 3.3.7, the current system enables us to recapture the A/A’-distinction in terms of phi-features on categories: A category in an A-position is reanalyzed here simply as a category bearing phi-features, while a category in an A’-position is reanalyzed here simply as a category lacking phi-features. Based on this type of distinction, "karera-

[38] This option corresponds to A'-scrambling. That is, if a moving element bears [iPhi], the movement is regarded as A-scrambling. If a moving element does not have [iPhi], the movement is regarded as A'-scrambling, in which the moving element never becomes a goal of phi-agreement after its [iPhi] is split off.

ga" moves through A-A-A positions in the hyper-raising case (without causing featural crash): the edge of the embedded CP is regarded as an A-position.

Second, consider the RTO cases. The RTO data in (42b) is repeated as (58) below:

(58) John-ga Bill-o$_1$ [t$_1$ baka-da to] omot-teiru.
John-NOM Bill-ACC [fool-COP COMP] think-PROG
"John thinks of Bill as a fool."

Here, the embedded subject undergoes movement to the edge of VP in the matrix clause across the CP boundary. Again, the same scenario as the last case is applied to this case. Therefore, "Bill-o" retaining [iPhi] moves from embedded Edge-TP (A) to the edge of the embedded CP (A) to the edge of VP (A). If [iPhi] on the moving element is split off to embedded Edge-TP, then [uPhi] on the matrix V is not valued causing featural crash. However, there is one thing which takes place specifically in the RTO case, but not the last case in (57). It is that Case is "revised" from nominative Case (valued by T in the first Merge position of "Bill" in the edge of vP) to accusative Case as a consequence of subsequent phi-agreement with the matrix V. Reconsidering (57), the matrix T agrees with the embedded subject which is moved to the matrix edge of vP as follows:

(59) T$_{[uPhi]}$ > karera-ga$_{[iPhi]}$
Agree

Since [Case] on "karera-ga" was already split off to the edge of the embedded TP, the matrix T in this case only phi-agrees with the moving element. Nominative Case, which was valued in the embedded clause is retained and realized in the position overtly occupied by "Karera-ga". However, in the RTO case, Case of the moved element is realized with the second Case what was valued i.e. accusative Case. How is this explained?

(60) matrix V$_{[uPhi]}$ "Bill-ga"$_{[iPhi]}$ *Bill-ga
Agree *Bill-ga-o
√Bill-o

Notice that RTO is allowed only when the matrix V is an ECM verb. That is, such Case-revision is possible only with ECM verbs in Japanese. If the matrix verbs are not ECM verbs, Case-revision (i.e. RTO) is disallowed:

(61) a. John-ga [Bill-ga baka-da to] kii-ta.
John-NOM [Bill-NOM fool-COP COMP] hear-PAST
"John heard that Bill is a fool."
b. ??John-ga Bill-o$_1$ [t$_1$ baka-da to] kii-ta.
John-NOM Bill-ACC [fool-COP COMP] hear-PAST
"John heard of Bill as a fool."

Based on these facts, I suggest that [uPhi] on V$_{ECM}$ in Japanese can revise phonological (i.e. phonological) features on a raised subject from nominative Case to accusative Case as a consequence of phi-agreement. This is one of the lexical properties ECM verbs in Japanese have. This is why in (60), the phonological realization of Case feature on "Bill-ga" is revised to "Bill-o (Bill-Acc)" by phi-agreeing with V$_{ECM}$ as opposed to the hyper-raised subject in (56). As we have already seen, in Japanese ECM constructions the embedded clause is a CP taking a finite TP complement. Therefore, if the embeded subject stays in the embedded clause, it is realized as nominative Case as in the contrasting case of (42), which is repeated as (62) below:

(62) a. John-ga [Bill-**ga** baka-da to] omot-teiru.
John-NOM [Bill-NOM fool-COP COMP] think-PROG
"John thinks that Bill is a fool."
b. John-ga Bill-**o**$_1$ [t$_1$ baka-da to] omot-teiru.
John-NOM Bill-ACC [fool-COP COMP] think-PROG
"John thinks of Bill as a fool."

This differs behavior from English ECM constructions:

(63) a. John believes that she is a fool.
b. John believes her to be a fool.

c. *John believes she to be a fool.

That is, English ECM verbs do not have the ability to revise Case (i.e. Case-revision). With respect to Case-revision, Bejar and Massam (1999) report the same kind of (construction-specific) phenomena in other languages such as Hungarian. (In Section 3.6, I revisit Case-revision and propose that English also permits (construction-specific) Case-revision in *tough*-constructions derived by "improper movement").

The proposed feature-splitting system can explain grammatical improper movement cases in Japanese by appeal to its weakly cpmmected Phi/Case-agreement processes, which determines how features on a moving element can be (phi/Case) split.

3.6 *Tough*-Constructions as Proper Improper Movement

This section further extends the feature-splitting system and shows that there is grammatical/proper improper movement not only in Kilega/Lusaamia and Japanese but also in English, namely *tough*-constructions.[39] In this section, I propose that *tough*-constructions are, in fact, derived by "proper improper movement", movement of an argument DP (cf. Epstein 1989, Brody 1993, Hornstein 2001 for similar analyses and Hicks 2009 for a proposal that both a lexical DP and a null operator undergo movement in the derivation of *tough*-constructions), not a null operator (contra. Chomsky, 1982).[40] Furthermore, this section reveals how the feature-split system rules in tough-constructions but rules out improper movement.

[39] Aspects of this section are based on Obata and Epstein (to appear).

[40] Although our approach and Hornstein's (2001) approach both suggest that an argument DP, not a null operator, moves in *tough*-constructions, I still assume PRO in contrast to the movement approach to Control discussed in O'Neil (1995) and Hornstein (1999, 2001).

3.6.1 Framing the Issues: Three Research Questions

This section extends the Feature-Split system by considering the following three questions. The first issue concerns *tough*-constructions:

(64) John is easy to please.
(65) *John seems that Bill likes.

One of the reasons why *tough*-constructions have always attracted attention is that one path "John" might in principle take is to move from the object position in the embedded clause to the matrix Edge-TP via the embedded Edge-CP. But of course, this is improper movement, as in (65). This is why a lot of attempts have been made to explain this construction without appeal to improper movement which has been widely assumed to be universally excluded and has thus far been regarded as a phenomenon completely unrelated/insensitive to variant feature constitutions of different types of T. (cf. Lasnik and Fiengo 1974, Chomsky 1982, Browning 1987 among others) As summarized in the previous sections, given the feature-splitting system, improper movement is ruled out because it causes featural crash by unvalued [uPhi] on the matrix T. That is, another way to possibly account for *tough*-constructions is to hypothesize that in those constructions, [uPhi] on the matrix T in (64), but not in (65), can somehow successfully phi-probe the *tough*-subject ("John" in (64)) when it has moved into the embedded Edge-CP and still bears its inherent [iPhi]. "John" would subsequently move improperly from Edge-CP to matrix Edge-TP. This kind of convergent improper movement derivation must somehow be blocked in cases like (65) discussed in Section 3.3 but should be possible in *tough*-constructions.

The second issue concerns, not the matrix T, but the embedded T, specifically the differences between finite T and control T. Notice that an embedded finite T as in (65) or with *tough*-constructions "John is easy that Mary pleases" is ungrammatical. Since Chomsky (2001), it has been assumed that both control T and finite T consist of [uPhi], which triggers phi-agreement with a subject and values Case on the subject as a consequence. That is, finite T and control T are not distinguishable in terms of their phi-feature content, i.e. both have a full set of [uPhi]. However, a well-known question confronting this analysis is why finite T cannot license PRO:

(66) *(John thinks that) PRO eats apples.

This data implies that control T and finite T behave differently with respect to subject-licensing. But how do they differ? One possibility to be pursued in this section is that finite T bears [uPhi] while control T bears [uCase]. Assuming different Case-valuation mechanisms between these two Ts provides us with a new way of feature-splitting, which is dependent on features on probes.

The third issue is why the theory of UG incorporates a null Case assigning T, but there is implicitly assumed to be no null Case assigning V. Why does the T head of a phase-head-complement, but not the V head of a phase-head-complement, value null Case? As addressed in the last paragraph, control T and finite T should be distinguished in terms of their feature content. The same logic arguably applies to V. After V inherits phi-features from v, it phi-agrees with an object DP. Then, [uCase] on DP is valued as a consequence of phi-agreement and is realized as accusative Case, while finite T values [uCase] on a subject DP and nominative Case is realized. But what about control T? Control T can license PRO valuing its null Case but not a lexical DP, unlike finite T. But why doesn't a corresponding V, licensing a PRO object, exist? Is there any logical reason which makes V licensing PRO impossible--or is this just an implicit stipulation? Is there empirical evidence in favor of null (PRO) objects? Arguably, "Yes", even in English as in e.g. (67) which appears to take object PRO:

(67) a. John washes PRO.
(cf. John washes himself.)
b. John is eager to please PRO
(cf John is easy to please, Chomsky 1965)

(67a) is interpreted as "John washes himself" without having any visible object. That is, it is possible that the object position is filled by PRO controlled by the subject "John". (cf. Rizzi, 1986 for Italian, Epstein 1984 for English and Chomsky, 1981 for his analysis of *tough*-constructions assuming an object PRO)[41] In (67b), there "must be" an argument in object position to saturate the internal theta role assigned by "please", so that there exists by hypothesis a discourse-determined antecedent for the null object, again, possibly PRO. If this speculation regarding null objects is on track, then there should be two Vs, just like there are 2 Ts: one licensing a lexical object and the other licensing PRO (leaving aside Raising T, a third kind irrelevant to this particular discussion). This assumption--that there exists finite V and also a Control V--renders T and V symmetrical in terms of their feature content and syntactic behavior, thereby eliminating an unexplained, if even recognized, lexical asymmetry in the feature content of the heads of phase-head-complements.

In the next section, the second and third issues regarding T and V are discussed in detail by advancing a more explicit hypothesis regarding their (parallel) feature content.

[41] A comprehensive examination of the voluminous and important literature concerning crosslinguistic variation concerning null objects lies beyond the scope of this paper.

3.6.2 Phase Head Complement Feature Symmetry

3.6.2.1 Three Types of T/C and (Correspondingly) Three Types of V/v

In this section, I further consider two of the three issues raised in the last section regarding properties of T and V. As summarized earlier, it has not been clearly stated how to distinguish control T and finite T and also why there is no V licensing PRO. In order to overcome these unanswered problems, I advance the following hypothesis:

(68) Hypothesis:
There are uniformly 3 kinds of C/T and 3 kinds of v/V distinguished in terms of their feature content.

(69)

C/T	
a. C $\boxed{\cancel{[uPhi]}}$ $T_{[uPhi]}$ inheritance	Nominative
b. C $\boxed{\cancel{[uCase]}}$ $T_{[uCase]}$ inheritance	Null Case
c. $T_{[---]}$	No Case

v/V	
d. C $\boxed{\cancel{[uPhi]}}$ $T_{[uPhi]}$ inheritance	Accusative
e. C $\boxed{\cancel{[uCase]}}$ $T_{[uCase]}$ inheritance	Null Case
f. $V_{[---]}$	No Case

In (69), finite T and control T can be distinguished by their feature make-up. That is, nominative Case is assigned as a consequence of phi-agreement triggered by [uPhi] on T. Meanwhile, null Case is assigned by Case-agreement triggered by [uCase] on T, so that there is no phi-agreement with a control T. Also, under this system, null Case assignment is not an ad hoc property of $T_{[uCase]}$ but there similarly exists a null Case assigning V, namely $V_{[uCase]}$. Furthermore, suppose three kinds of T and V are freely combined, i.e. there is no selectional constraint on the structure building combination of these.

(70) T[uPhi]
a. V[uPhi]
b. V[uCase]
c. V[---]

(71) T[uCase] a. V[uPhi]
b. V[uCase]
c. V[---]

(72) T[---] a. V[uPhi]
b. V[uCase]
c. V[---]

The following sections clarify what types of sentences are derived by each of these nine combinations.

3.6.2.2 [uCase]: On Null Case Assignment

Let us begin by considering how [uCase] valuation takes place. In the analysis discussed above, [uCase] plays an important role in distinguishing between T/V licensing a lexical DP and T/V licensing PRO. The first issue in need of clarification is how an unvalued feature ([uCase] on T) values another unvalued feature ([uCase] on DP). The basic properties of the feature valuation system are: [1] an unvalued feature on a probe triggers Agree with a goal bearing its valued counterpart, and then [2] the unvalued probe feature receives a value from the goal. In the case of T/V$_{\text{[uCase]}}$ probing, however, this scenario is not applicable in that the goal DP also has [uCase] like the probe, not the valued counterpart. That is, neither the goal nor the probe has a value. (See also Epstein and Seely 2006 for discussion of Agree between unvalued features.) With respect to this problem, I propose that it is in this context that "null/zero" Case is valued.

(73) The mechanism of null Case assignment:

PROBE	GOAL			
T/V$_{\text{[uCase]}}$	PRO or lex DP$_{\text{[uCase]}}$	→	T/V$_{\text{[null Case]}}$	PRO/DP$_{\text{[null Case]}}$
Case-Agree				

Null Case is, I propose, to be understood as phonetically "unspecified" Case in this context. Crucially, the computational system recognizes that [uCase] is valued, here as [null Case], which when transferred, does not cause crash-due-to-no-value. That is, [uCase] valued by $T/V_{[uCase]}$ as in (73) is regarded as null Case. But since the assigned value is "null", however, phonological features on PRO or DP in (73) are unspecified. More concretely, a feature bundle comprising e.g. [+ pronominal], [3rd person], [singular], [masculine] and [Nom] is phonetically specified as "he" when valued by $T_{[uPhi]}$. On the other hand, the same feature bundle is NOT phonetically specified if valued by $T/V_{[uCase]}$, though it is valued. That is, valuation (as null Case) has occurred, but phonetic specification hence phon-licensing of DP/PRO (at this point) is lacking. This is a desired result for PRO because PRO is phonetically empty to begin with hence requires/receives no phonetic encoding, and perhaps disallows it--as in P&P Case approaches to the distribution of PRO. However, what happens if a lexical i.e. phonetically overt DP receives null Case? It has phonological features which must undergo interpretation by the Sensory-Motor system (SMS). Consider the following standard paradigm:

(74) a. Mary hopes to go.
b. *Mary hopes Bill to go.

$T_{[uCase]}$ assigns null Case to PRO in (74a) and to "Bill" in (74b). Why is (74b) ungrammatical? Following Epstein (1990), I make the empirical claim that what is wrong with (74b) is phonetic, not theta-theoretic or semantic, as arguably evidenced by the coherent semantic interpretation we (our CI-systems) can impose on (74b). I propose the following PF and LF representations for (74b):

(75) PF and LF representation for (74b)
PF: Mary tried to go.
LF: Mary tried Bill to go.

Crucially, there is no Case crash in the PF representation i.e. the derivation converges at

PF. That is, Case is in fact valued, specifically it is valued as [null Case], which is a legitimate feature at the PF interface. Since "Bill" has [iPhi], the representation at the LF side contains "Bill". On the other hand, phonological features on "Bill" are unspecified by null Case, so that it is unpronounceable with null Case by SMS reading the PF representation, i.e. only specified phonological features are "pronounceable". Consequently, the inherent lexical phonological features on "Bill" never undergo their required PF interpretation. This violates the Recoverability Condition, which requires that no lexical, interface-interpretable information be "deleted" or rendered uninterpretable by narrow syntax operations (Chomsky and Lasnik, 1995). In this case, phonological information on "Bill" is lost. This is why (74b) is ruled out. Again, it will be crucial to the proposed analysis of *tough*-constructions that null Case valued on lexical DP does NOT crash, i.e. the Case feature of "Bill" *is* valued (as null Case).

There is another possible outcome for such phonetically unlicensed lexical DPs like "Bill" in (75). The phonological features on those elements could in principle become specified by other (higher) probes later on in the derivation. In other words, the valued null Case (which does NOT cause crash but renders Bill unpronounceable) might be later "revised" by another higher Case-valuer, so that the phonological features of "Bill" are licensed at the PF side observing the recoverability condition, informally it *is* pronounceable I claim that this is exactly what happens in the derivation of *tough*-constructions. Putting aside the Feature-Splitting system for the time being, consider:

(76) He is easy to please.

I propose that the DP is first merged into the direct object position of "please". In this example, suppose the computation has selected the null Case valuing form of "please" (not the accusative valuer which inherits phi-features from v). Then, in the embedded clause, Case on the object is valued as null Case. As a result, Transfer of the embedded VP does not crash—precisely because Case *is* valued, in this case, as null.

(77) In the embedded clause:

please$_{\text{[uCase]}}$ $\boxed{\text{he/him}_{\text{[uCase][iPhi]}}}$ $\longrightarrow$ he/him$_{\text{[null Case]}}$

Case-Agree

But, as noted above, null Case is insufficient to license SMS pronunciation of a lexical DP, i.e. a DP bearing inherent phonological features. Suppose (somehow) the matrix T ("is") also agrees (later in the derivation) with the DP, changing its Case from null to Nom.

(78) In the matrix clause:

is$_{\text{[uPhi]}}$ $\boxed{\text{he/him}_{\text{[iPhi]}}}$ $\longrightarrow$ he/him becomes HE.

Phi-Agree

In (76), the *tough*-subject "he" is interpreted as the object of "please" but is pronounced Nom at the matrix subject position. That is, "he/him" is first merged into the object position of "please". If V ("please") has [uPhi], it assigns accusative Case to the object, which is realized as "him" as in "It is easy to please him". If V has [uCase] as illustrated in (77), Case-valuation takes place between V and the [uCase] object and null Case is assigned. Again, [uCase] on the object is valued (there is no crash upon Transfer of the embedded VP) but the phonological features are still unspecified. Then, the object undergoes movement to the edge of embedded CP. Since this is a lexical DP, it has [iPhi]. The matrix $T_{\text{[uPhi]}}$ can now phi-agree with the moving element as shown in (78). As a consequence of this phi-agreement, phonological features are finally specified as "he", which is represented at PF. That is, no recoverability violation occurs. This is why (76) is grammatical. (Section 3.6.4 will further consider the derivation for *tough*-constructions later highlighting how they differ from other improper movement phenomena under the feature-split analysis.)

Thus, under the analysis proposed here, Case on a *tough*-subject is "revised" (or perhaps stacked) in the course of the derivation, which makes it possible to specify phonological features. But is such Case revision an ad hoc mechanism lacking any independent motivation, beyond our analysis of *tough*-constructions? Interestingly, Case-revision is also observed in other languages. Bejar and Massam (1999) present data from languages such as Hungarian, where Case on DP can be "revised" after the DP undergoes A'-movement. They call this phenomenon "multiple Case-checking".

(79) kiket mondtad hogy szeretnél ha eljönnének
Who-Acc you-said that you-would-like if came(3pl)
"Who did you say that you would like it if they came?"
(Bejar and Massam, 1999:66, cited from Kiss, 1985)

"Kiket" is extracted out of the subject position of the finite *if*-clause, in which it received nominative Case. On the way to the final landing site, it subsequently gets accusative Case from the intermediate verb ("said") and the second Case--accusative Case--is always the one that is phonetically realized. That is, Case is multiply assigned to "kiket", although as Bejar and Massam point out, this is (interestingly in our view) a construction-specific phenomenon, like *tough*-constructions are in English.[42] Also, I proposed in the previous section that Japanese RTO phenomena involve Case-revision and demonstrated how it takes place. (See also Horvath 1981 for relevant discussion.)

Summarizing the discussion so far, [uCase] on a V-probe triggers Case-agreement with a goal and null Case is assigned. Since null Case is a valued Case, its appearance at

[42] There is something different between our Case-revision and Bejar and Massam's multiple Case-checking: In the former case, the first Case is always null Case while in the latter, the first Case is not null but phonetically realizable.

PF when transferred, does NOT induce crash. But null Case is phonetically unspecified Case, phonological features are not represented at the PF representation. Therefore, if the assignee is a lexical DP, not PRO, Recoverability requires it to be phonetically specified by agreeing with another probe at some subsequent point of the derivation.

3.6.3 Nine Possibilities

The above section explained how [uCase] behaves in the derivation. Now, let us examine the nine possible T-V combinations in (70)-(72) one by one.

(80) T[uPhi] a. V[uPhi] → e.g. He likes her.[43]
b. V[uCase] → e.g. He washes.
c. V[---]→ e.g. He was arrested.

The first set includes three combinations each with $T_{[uPhi]}$. $T_{[uPhi]}$ is a finite T, so that it can assign nominative Case whose phonological features are specified as "he" in (80). $V_{[uPhi]}$ values accusative Case on the object, so that it is realized as "her" as in (80a). In (80b), $V_{[uCase]}$ values null Case as summarized in the last section. If the object is PRO, which is controlled by the subject "he", the sentence is fine. But if a sentence like "He washes the dog" is derived under this combination, the derivation is ruled out because the object "the dog" which gets null Case never undergoes Case-revision thereby causing a PF recoverability violation. In (80c), v which transmits phi/Case-features to V is missing. This is why V does not have any unvalued feature triggering Agree. This corresponds to e.g. the passive case, so that the object needs to get nominative Case from $T_{[uPhi]}$.

Next, consider the second set of T/V combinations which contains three different types of V each selected by $T_{[uCase]}$.

[43] In Section 3.3, I assumed that improper movement cases fall into this combination. The verb "seem" selects either a finite CP or an infinitival TP but not an infinitival CP as in "*It seems [$_{CP}$ PRO to go]".

(81) T[uCase] a. V[uPhi] → e.g. It is easy to please him.
Mary wants to please him.
b. V[uCase] → e.g. He is easy to please.
c. V[---]→ e.g. He wants to be arrested.

The underlined parts i.e. the embedded clauses exemplify the T-V combination at issue. $T_{[uCase]}$ values null Case of the subject. This is why the subjects in (81) are all PRO, which does not require Case-revision. In (81a), $V_{[uPhi]}$ assigns accusative Case to the object, so that the objects are realized as "him". In (81b), the *tough*-subject "he" first gets null Case from $V_{[uPhi]}$ when occupying the direct object position. Since null Case is valued case, there is no Crash when the embedded VP containing the lexical DP with null Case is transferred. A second Case (i.e. nominative) is later assigned by the matrix finite T when he/him occupies embedded Edge-CP. Case is successfully revised and the pronominal DP can be pronounced as "He" and so there is no Recoverability violation. In (81c), the embedded V does not assign any Case to PRO. Rather, PRO gets null Case from the embedded $T_{[uCase]}$. The derivation converges.

Finally, the third set includes three V types co-occurring with $T_{[---]}$ as follows:

(82) T[---] a. V[uPhi] → e.g. He seems to please Mary.
b. V[uCase] → e.g. He seems to wash.
c. V[---] → e.g. He seems to be arrested.

In (82), T does not have any phi-features triggering Agree because C is missing. That is, this T is the kind selected by a raising predicate. In (82a), $V_{[uPhi]}$ assigns accusative Case to the object “Mary”. Then, the subject of the embedded clause “he” cannot get Case from $T_{[---]}$, so that the matrix finite T assigns nominative Case to it. In (82b), [uCase] on “wash” triggers Case-agreement with an object PRO. Again, “he” moves and gets Case from the matrix finite T. In (82c), $V_{[---]}$ does not trigger any agreement just as $T_{[---]}$ does not, so “he” finally agrees with the matrix finite T.

As illustrated above, each of the nine combinations works to derive each type of sentence. Given this, there is no need to be troubled with two of the three questions specified in Section 3.6.1: [1] What distinguishes a finite T which assigns nominative Case, and a control T, which assigns null Case to PRO, given that both types of T have identical [uPhi] under Chomsky (2001, 2007, 2008)?, and [2] why is there a null Case assigning T but there is implicitly assumed to be no null Case assigning V? For the first question, our answer is that those two different Ts are distinguishable in terms of their feature content. In response to the second question, now I can say that there is a null Case valuing V just like there is a null Case valuing T, thereby eliminating the cross categorial asymmetry. In the next section, I will tackle the remaining question about *tough*-constructions and improper movement based on Feature-Splitting Internal Merge.

3.6.4 Feature-Splitting: Separation of [Case] and [Phi]

3.6.4.1 How to Distinguish ('ungrammatical') Improper Movement from ('grammatical') Tough-Constructions

This section especially considers what kind of mechanism allows us to derive *tough*-constructions (see (76)-(78)) but concomitantly rule out classic improper movement. Under the view that a *tough*-subject itself moves from the embedded object position to the matrix subject position, the derivation for *tough*-constructions displays the same derivational steps as improper movement does, namely movement from A-to-A'-to-A positions as follows:

(83) *John seems (that) Mary likes.
[$_{TP}$ John seems [$_{CP}$ <John> (that) [$_{TP}$ <John> likes]]]

(84) John is easy to please.[44]
[$_{TP}$ John is easy [$_{CP}$ <John> [$_{TP}$ to please <John>]]]

However, the outcome is different: (83) is ungrammatical while (84) is grammatical. The main goal of this section is to explain why.

3.6.4.2 Another Way of Feature-Splitting

Section 3.3 discussed Feature-Splitting Internal Merge. Recall the way in which features get split depends on whether those features participate in Agree with T/V. Consider the following improper movement case:

(85) *Who seems left?

After "who" is first merged into Edge-vP in the embedded clause, the embedded finite T whose features are inherited from C phi-agrees with it. Then, EFs on T and C attract "who" simultaneously as follows:

(86) [$_{CP}$ C [$_{TP}$ T$_{[uPhi]}$ [$_{vP}$ who$_{[iPhi][uCase][iQ]}$ …]]]

In this context, features on "who" are split into the two syntactic landing sites as a consequence of simultaneous attraction by C and T:

(87) [$_{CP}$ who$_{[iQ]}$ C [$_{TP}$ <who$_{[iPhi][uCase]}$> T [$_{vP}$ <who$_{[iPhi][uCase][iQ]}$> …

---

[44] *Tough*-constructions are prohibited when a subject in the embedded clause is moved as in "*John is easy to please Mary". This is an interesting case of overgeneration made possible by our more permissive system which allows improper movement derivations of *tough*-constructions, and allows a lexical DP to receive null Case, provided it is revised--a condition satisfied in this ungrammatical example. I speculate that this subject-object asymmetry is properly ascribed to ECP which, in GB, prohibited movement to be launched from an ungoverned position licensing PRO. (cf. Epstein 1984) (See Obata and Epstein, in prep for further discussion of comp-trace effects and "for" as a Case-valuer.)

I suggested that T attracts features it agreed with and C attracts the rest of the features, which can be induced from Chomsky's (2007) A/A'-distinction--"A-movement is IM (Internal Merge) contingent on probe by uninterpretable inflectional features…" (Chomsky, 2007:24) Therefore, both [uCase] and [iPhi] on "who" since both were involved in phi-agreement with T go to Edge-T, while [iQ] on "who" moves to the edge of CP, so that "who" at the edge of CP bears only [iQ] but has no [Phi] and no [Case]. Since the embedded TP is transferred, and "who" in the embedded Edge-CP lacks phi-features, this leaves [uPhi] on the matrix T unvalued, since it can find no goal DP bearing [iPhi]. Therefore, (85) crashes, as desired (without global Condition C or global Chain Uniformity Condition invoked to rule out improper movement, which is here re-analyzed as a phi-agreement failure i.e. local, featural crash).

If some mechanism were to make it possible that "who" at the edge of CP were to keep its [iPhi], and carry these features to Edge-CP, then this derivational variant of (87) would successfully converge, since a higher, externally-merged matrix T could phi-agree with who in Edge-CP. What is "some mechanism"? Recall that I suggested in Section 3.6.2.1 that there are three kinds of T/C and three corresponding kinds of V/v and demonstrated what kinds of sentences are derived in each of the combinations. Given that, let us examine exactly how *tough*-constructions are derived:

(88) John is easy to please.

Tough-predicates do not take a finite CP, i.e. "John is easy that Bill pleases" and "*It is easy that Bill pleases John" are ungrammatical. That is, *tough*-predicates always select $T_{[uCase]}$, which is a null Case assigner. With respect to the embedded V, since "please" in (88) is optionally an accusative or null Case valuing verb, under my analysis, it is either

$V_{[uPhi]}$ or $V_{[uCase]}$. If $V_{[uPhi]}$ is chosen, then the object receives accusative Case. That is, a sentence like "It is easy to please him" is derived. As discussed earlier, *tough*-subjects get Case in the course of the derivation first from "please" (null Case) and later from the matrix T (nominative Case). In other words, Case is revised, stacked, or multiply assigned. $V_{[uCase]}$ values null Case, so there is no crash of previously transferred phase head complements as explained in the last section. Now, let us see how a *tough*-construction is derived, precisely. (NB: The capital letter "HE" below stands for a phonetically unspecified pronoun.)

(89) He is easy to please.

Step1: [$_{vP}$ PRO v [$_{VP}$ $V_{[uCase]}$ $HE_{[iPhi][uCase][Op]}$]]

V Case-agrees with HE and null Case is valued.

Step2: [$_{vP}$ $HE_{[iPhi][Op]}$ PRO v [$_{VP}$ $HE_{[null]}$ $V_{[null]}$ $HE_{[iPhi][uCase][Op]}$]]

Feature-Splitting: V attracts only the MATCHING [uCase] and v attracts the rest, crucially, [Op] and [iPhi].

Step3: [$_{CP}$ $HE_{[iphi][Op]}$ [$_{TP}$ PRO $T_{[uCase]}$ [$_{vP}$ $HE_{[iPhi][Op]}$ PRO v […]]]]

EF on C attracts HE bearing [iPhi] and [Op] to Edge-CP.

Step4: [$_{TP}$ $he_{[iPhi]}$ $T_{[uPhi]}$ [$_{CP}$ $HE_{[iPhi][Op]}$ C […]]]]

Matrix T[uPhi] phi-agrees with HE in Edge-CP.
HE is revised to nominative he.

(89) is the derivation for *tough*-constructions. Notice in Step2 that feature-splitting takes place in a different way from the ungrammatical improper movement case (85). Here, the only feature which participated in Agree with "please" $V_{[uCase]}$ is [uCase] on "HE". That is, [iPhi] and [Op] are not involved in this relation. This is why V attracts only [uCase] to Edge-VP and the rest of the features are attracted by v. This makes a crucial featural distinction between the ungrammatical improper movement case and the grammatical

(also improper movement) *tough*-construction. Only in the latter does, "HE" in Edge-CP retain its [iPhi], thereby allowing the matrix finite T to avoid crash by agreeing with it. Another important point here is that in Step3, notice that $T_{[uCase]}$ Case-agrees with subject PRO across the intervening "HE". How is this possible? Assuming with Chomsky (2001) that each feature functions as an independent prober, $T_{[uCase]}$ can indeed agree with $PRO_{[uCase]}$ across the intervening HE, and this is possible precisely because the [uCase] feature of "HE" has already been split off and was moved to Edge-VP. Thus, there is no Case-feature intervention effect in such derivations. (Also see Section 3.3.6 and 3.4.3.3 for discussion of other predictions regarding intervention/opacity under the feature-splitting analysis.) On the other hand, the following object improper movement case with an embedded finite clause is ruled out precisely because of the intervention effect at issue:

(90) *John seems that Mary likes.

If the $V_{[uCase]}$ form is selected for the embedded verb "likes", then "John" can move keeping [iPhi], and it would seem as if [iPhi] on "John" in embedded edge-CP can then value [uPhi] on the matrix T overgenerating (90). However, consider the configuration right after features on "John" are split and T and C are introduced:

(91) $[_{CP}$ C $[_{TP}$ $T_{[uPhi]}$ $[_{vP}$ $\textbf{John}_{\textbf{[iPhi][Op]}}$ $\text{Mary}_{[iPhi][uCase]}$ v […]]]]

Recall, [uCase] on "John" has already been split off having moved to Edge-VP. [iPhi] and [Op] on "John" are attracted by v and land at the outer edge-vP position. Then, T and then C are introduced. After [uPhi] on C is inherited by T, T attempts to phi-agree with "Mary" across the intervening "John", which recall also has [iPhi]. However, the intervening [iPhi] on "John" causes an intervention effect, so that phi-agreement between T and "Mary" is NOT allowed. As a result, [uCase] on "Mary" remains unvalued causing crash of the derivation. This configuration contrasts with Step3 in (89), where T Case-agrees with "PRO" across "HE" lacking [Case]. The proposed analysis can allow *tough*-constructions but also exclude ungrammatical object improper movement.

Notice the proposed analysis predicts that if the downstairs verb “likes” has a full phi-feature set, then [iPhi] on “John” will value these features and will split off and move to Edge-VP. As a result, John in outer edge-vP lacks [Phi] (and [Case]) thereby allowing the embedded T to agree with "Mary" across “John”. However, now [uPhi] on the matrix T cannot be valued by “John”, since “John” lacks phi-features. But now two different scenarios require consideration: [1] Suppose an expletive, not John, is used to value [uPhi] on the matrix T, or [2] Suppose the matrix T is the phi-less “raising-type T”. Case [1] is simply the grammatical: “It seems that Mary likes John”. Case [2] is an ungrammatical example such as “*John to seem that Mary likes _ is unfortunate”, (it means “for it to seem (e.g. to Sam) that Mary likes John is unfortunate”). This important possible overgeneration needs to be further investigated but I leave it for future discussion. Note under Epstein and Seely's (2006) analysis (different from the principles assumed here), the structure is not generable, precisely because "pure-EPP movement" i.e. movement to Edge of phi-less T never occurs (since there is no EPP). (This and other possible landing sites for "John" in such a derivation require further investigation.)

As demonstrated in this section, the feature-splitting system can derive *tough*-constructions (by "proper improper movement" but concomitantly excludes "classic" improper movement. Now, I can answer the last question "what kind of mechanism allows us to derive *tough*-constructions improperly but rule out classic improper movement?" These two constructions select different types of T and V (in accordance with their selectional restrictions), so that features are split in different ways depending on the feature content of the probes. Only in *tough*-constructions, (with a null-Case valued Object) does the agreeing downstairs V ("please") bear only [uCase]. This in turn makes it possible for the phonologically unlicensed moving direct object to keep its phi-features, carrying them to the edge of the embedded CP. This in turn permits a higher $T_{[uPhi]}$ to undergo agree with this edge-CP, thereby permitting convergence, even though "improper movement"--reanalyzed here as a feature-variant local agreement phenomenon--has occurred. The corresponding derivation in which the tough predicate is replaced by a raising predicate crashes.[45]

A final note concerns the noted absence of PP null operator. The analysis presented here makes no appeal to null operators. Chomsky (1982) suggests that null operator movement takes place in *tough*-constructions. However, Browning (1987) presented the following data to show that null operators cannot replace PP:

(92) a. *With a hammer is dangerous for John to fix the television.
b. *On Tuesday is difficult for John to come to a meeting.
c. *To Bill is difficult for John to give a book

(Browning, 1987)

[45] The current system does not straightforwardly work for the following *tough*-construction involving PRO movement:

(i) "John wants to be easy to please."

Rep. [$_{CP}$ [$_{TP3}$ John$_1$ wants [$_{CP}$ [$_{TP2}$ PRO$_1$ to be easy [$_{CP}$ t$_1$ [$_{TP1}$ PRO$_2$ to please t$_1$]]]]]]

PRO$_1$ is moved from the embedded object position to its surface position Edge-TP$_2$ and is controlled by "John". In this sentence, the final landing site of PRO$_1$ is the edge of the infinitival TP$_2$, in which T bears [uCase] valuing null Case under the current system. However, [uCase] on PRO$_1$ is already valued by "please" and split off to the edge of the lowest VP. If this is a lexical DP, its phonological feature has to be specified later by agreeing with a finite T. In the case at issue, however, the moving element is PRO, which has no phonological features. Therefore, as mentioned in Section 3.6.2.2, it does not require specification of phonological features. That is, the problem is that [uCase] on T$_2$ is never valued since PRO at the edge of embedded CP no longer bears matching [uCase], but rather had its Case valued as null Case by "pleasse" and removed in the previous phase causing featural crash, contra the fact. At this point, I have no answer to this problem unless some stipulation is made. I leave this problem as a pending problem.

That is, under the null operator analysis, it is necessary to lexically stipulate that *tough*-constructions are allowed only for DP but not PP. On the other hand, the analysis proposed here (which does not appeal to null operators) excludes the sentences in (92) since PP does not have [iPhi], so that [uPhi] on the matrix T remains unvalued, just like in ungrammatical improper movement cases.[46] This lends further empirical support to the current approach to *tough*-constructions based on feature-splitting.

3.6.5 Discussion and Outstanding Issues

Section 3.6 has proposed the following:

- There are three kinds of C/T and also three corresponding kinds of v/V.
- V (like T) can assign null Case to an object.
- Case-revision in *tough* is explained by saying that a matrix finite T phi-agrees with a moving element whose phonetic features are underspecified because of null Case assigned by $V_{[uCase]}$.
- A-A'-A (so-called "improper") movement is NOT universally excluded.
- Both ungrammatical improper movement and grammatical *tough*-constructions are explicable by feature-splitting internal-merge and the (non-)intervention effects which splitting induces, given individual feature probing.

If the discussion here is on the right track, several hidden problems regarding properties of T and V are clarified while Feature-Splitting Internal Merge gains further theoretical and empirical support. Classic improper movement is re-analyzed as a local, featural agreement failure (i.e. crash) (unrelated to unbounded Condition C application or Chain

[46] I must assume that Agree does not go into PP, so that it cannot "see" DP bearing [iPhi] inside of the PP itself occupying the edge of CP and this too requires further research.

Conditions), as discussed in Section 3.3. The analysis, if successful, is restrictive enough to prohibit classic improper movement while concomitantly allowing "improper movement" derivations of *tough*-constructions. Other aspects of *tough*-constructions not explored here of course require further research, as does the ungrammaticality of subject-*tough* movement and PP-*tough* movement, as (all too) briefly discussed here. One very promising general aspect of this type of Minimalist Analysis, if on track, is that "construction-specificity" might be accurately reducible to an independently motivated, partially parameterized theory of arguably irreducible syntactic features and morpho-syntactic variation.

3.7 Summary

Ch.3 explored aspects of the mechanism of Internal Merge, especially focusing on simultaneous attraction of a single element by multiple heads. In Section 3.2, I proposed Feature-Splitting Internal Merge, which is a new way of structure building and demonstrated that the proposed mechanism can be induced from Richards and Chomsky's argument for Transfer-Valuation simultaneity.

Section 3.3 suggested that the feature-splitting system can give a LOCAL featural account to improper movement based on Chomsky's (2007, 2008) phase-based derivational system. The legitimacy of improper movement is, I claim, parameterized. In some, but not all I-languages, it is excluded as a consequence of the feature-split hypothesis and without appeal to the Activity Condition. If the proposed system is on the right track, improper movement is excluded by agreement failure causing featural crash. I also clarified some consequences obtained from the proposed system including a conjecture that A/A'-properties can be recaptured categorially/featurally (without appeal to stipulated position-types) in terms of the presence/absence of phi-features.

Section 3.4 further extended the system developed in Section 3.3 to two Bantu languages Kilega and Lusaamia which permit "generalized" improper movement unlike in English. I suggested Feature-splitting is parameterized between English-type languages and Kilega-type languages. The parameters concern feature hierarchies (bundling and separability) of functional features universally present in mophosyntactic feature inventories. In addition, I also argued that A'-opacity/transparency effects are parameterized and unified with, hence co-vary with, the possibility/impossibility of improper movement.

Section 3.5 considered Japanese, which permits improper movement like in Kilega/Lusaamia but has neither C-agreement nor rich agreement morphology unlike in Kilega/Lusaamia. I specifically examined two cases: Raising-to-Object constructions and hyper-raising. A close look at the Japanese Case/phi-valuation system provides us with a third type of feature-splitting, which also allows an element moving across a phasal boundary to retain phi-features. In addition, I suggested a possibility that Japanese ECM verbs have a characteristic property of Case-revision: The verbs can "revise" phonological features of DP through phi-agreement.

Section 3.6 discussed *tough*-constructions in English. I demonstrated that also in English, there is a construction which permits a moving DP to retain phi-features and carry them to the edge in addition to constructions in which phi-features are split off. In the latter, improper movement is ruled out by featural crash as discussed in Section 3.3. But, like in Japanese and Kilega/Lusaamia, I explained that *tough*-constructions are regarded as one grammatical improper movement case. Also, some of the unclear issues regarding PRO and the mechanism of null Case assignment were also considered.

The next chapter (Ch.4) applies Richards-Chomsky's value-Transfer simultaneity system to root clause computation and elucidates the mechanisms of root transformations.

Chapter 4

Problems Confronting Root-Clause Transfer: "Solutions" and Predictions

4.1 Framing the Issues

This chapter further extends Chomsky/Richards' Value/Transfer simultaneity analysis to other types of phenomena, specifically considering root clause phenomena.[1] In Chapter 2, I overviewed Richards' (2007) deduction explaining why uninterpretable features on phase heads have to be inherited by lower heads in convergent derivations. If valued uninterpretable features e.g. valued [uPhi] are stranded on phase heads i.e. evade Transfer, the distinction between [iF] and valued [uF] disappears and those stranded valued [uF] cause crash of the derivation at the next phase. This is why valued [uF] can never remain on phase heads but rather must be inherited by lower heads which are included in the phase head complement, i.e. the transferred domain. In Chapter 3, I revealed implications of the analysis regarding elements moved to phase edge positions. While Richards (2007) focuses on [uF] on a phase head, Chapter 3 threw light on non-head phase edge positions, which are also stranded after Transfer of the phase head complement. I proposed Feature-Splitting Internal Merge which renders valued [uCase] on a moving element split off from the phase edge position. I demonstrated that the proposed mechanism enables us to give an entirely new account of improper movement phenomena, which I propose are ungrammatical in some languages but not in others. In a nutshell, why valued [uF] can

[1] Aspects of this chapter are based on Obata (2009).

appear neither at phase heads nor at any phase edge positions is that those positions are transferred as part of the next higher phasal domain, not as part of the phase head complement within which the valued [uF] bearer was externally-merged. That is, this analysis crucially assumes that Transfer applies only to the phase head complement thereby stranding the edge.

This chapter considers how a root edge can ever be transferred under the current Transfer system, which leaves a phase head and its edge behind. A root edge is never included in the domain of any phase since it is not part of a phase head complement. That is, the current Transfer system, which sends only the domain of a phase, does not straightforwardly work for Transfer of the root edge. This is the issue I pursue in this chapter.

The chapter is organized as follows: Section 4.2 reveals the problem confronting Transfer of a root edge and advances a working hypothesis resolving the problem. Section 4.3 investigates some implications obtained from the suggested hypothesis and considers two root transformations in detail: English T-to-C movement and root postposing in Japanese. Section 4.4 provides a summary and conclusion.

4.2 Transfer of Root Clauses

This section addresses aspects of the feature-valuation system and the phasal transfer operation proposed in Chomsky (2007, 2008). Also, I will demonstrate that if Chomsky's system is on the right track, the matrix C and only the matrix should be allowed to keep uninterpretable features [uF], unlike all other phase heads whose phi-features must be inherited by lower heads. From this asymmetry, I seek to deduce aspects of the idiosyncracies of root-clause syntax.[2]

4.2.1 The Transfer Operation

In the phase-based derivational approach, which is first suggested in Chomsky (2000), the linguistic computation occurs only in a limited work space, so that computational complexity can be, by assumption, reduced more than the computation applied to the entire representation at once. This idea makes it possible that the computational system "forgets" the derivation computed in previous phases. Chomsky (2000) argues that "the computational burden is further reduced if the phonological component too can "forget" earlier stages of derivation (Chomsky, 2001:13)". To do this, he assumes that Transfer/Spell-Out to the interfaces applies phase by phase. More precisely, the domain of a phase is transferred to the interfaces.

(1) $[_{HP}\ \text{XP}\ [\text{H}\ \boxed{\begin{array}{c}[_{YP}\ \text{Y}\ \langle\text{XP}\rangle] \\ \text{Transfer}\end{array}}]]$

In (1), H is a phase head. After the narrow syntax computation of HP is completed, Transfer sends YP to the Interfaces. Under current assumptions, YP undergoes Transfer and H and its XP edge remain in narrow syntax without getting transferred. As a result, the syntactic computation can "forget" YP, so that it leads to the reduction of computational complexity. H and its edge positions are still active in the derivation in the next higher phase and are transferred as members of the domain of the next phase. These ideas regarding the phasal transfer system have been basically maintained in Chomsky's (2000, 2001, 2004, 2007, 2008) analyses.

---

[2] Chomsky (2004: 108) points out this issue and suggests that in root clauses, phases are transferred in full although he does not specify the mechanism in detail. I will discuss this problem in Chapter 5.

4.2.2. Puzzles: How to Transfer Root Clauses

As discussed earlier, the phasal Transfer system and the feature-valuation mechanism both lead us to the conclusion that [uF] on a phase head has to be discharged to a lower head. Given the current phasal Transfer system, only the domain of a phase is sent to the interfaces, stranding the edge which is transferred in the next higher phase--but what happens in root clauses? Consider,

(2) What did you buy?
[$_{CP}$ what did (C) [$_{TP}$ you [$_{vP}$ <what> [$_{vP}$ <you> v [$_{VP}$ buy <what>]]]]]
TRANSFER2 TRANSFER1

At the first Transfer, vP is completed and VP is sent to the interfaces. Then, the derivation proceeds to the next higher phase CP. At the second Transfer, CP is completed and TP undergoes Transfer. However, C and its edge position still remain without getting transferred. Remember that Chomsky's phasal Transfer system assumes that only the <u>domain</u> of a phase is transferred. In (2), "what" and "did" are NOT included in the domain of any phase at any point in the derivation. That is, the puzzle here is how the matrix C and its edge are ever transferred under the phasal Transfer system, as defined by Chomsky (2007, 2008).

As long as it is true that a phase head and its edge in a root clause is pronounced and participate in semantic interpretation, the portion at issue MUST be somehow transferred.[3] This implies that something exceptional occurs only to the "last" phase in the derivation. I here advance the following working hypothesis with respect to this issue:

[3] Noam Chomsky (p.c.) suggests a possibility that the root edge is not transferred because it contributes to global computation (rather than cyclic computation) such as sentential intonation which is based on clause-typing by the matrix C. In this thesis, I assume that the root edge is also transferred as lower phases are and that valued [uF] on the matrix edge causes crash of the derivation under Chomsky/Richards' Value-Transfer simultaneity analysis.

(3) Working Hypothesis (to be revised in Chapter 5):
Unlike all other phases, the highest phase head and its edge are transferred "for free" along with the phase head complement.

Under this hypothesis, it is assumed that in (2) the edge "what did" undergoes Transfer along with TP at the second Transfer. One might wonder how the computational system knows which phase is the "highest" or final one. I put aside this problem in this chapter and will discuss in detail in Chapter 5 how root CP and embedded CP are distinguishable.

While the highest phase and its edge are successfully transferred under the working hypothesis in (3), this hypothesis makes an interesting prediction regarding Richards' (2007) argument. Recall that he argues that [uF] cannot be stranded on a phase head because phase heads are not transferred until the domain of the next higher phase is transferred. Given (3), however, the highest phase head undergoes Transfer along with the phase head complement. That is, <u>[uF] can remain on a phase head without getting inherited by a lower head but ONLY in the highest phase.</u> What does this imply? This implies that only the highest phase head can directly probe and enter into an Agree relation, unlike all other phase heads whose phi=features mush be inherited by a lower head. In the next section, I will clarify that this prediction regarding the highest phase head is empirically upheld playing an important role in explaining unique properties of root transformations.

4.3 Explaining Root Transformations

If the working hypothesis (3) is on the right track, the prediction is that only the matrix phase head and its edge can retain valued uninterpretable features, which can trigger Agree, under Chomsky/Richards Value/Transfer simultaneity. This "special" status which is allowed only for the matrix phase seems to comport with the behaviors of some of the root transformations, defined by Emonds (1976) as follows:

(4) Root Sentence
A root S ("sentence") is an S that is not dominated by a node other than S.

(5) Root Transformation
A transformation (or a transformational operation, in the case of a transformation performing several operations) that moves, copies, or inserts a node C into a position in which C is immediately dominated by a root S in derived structure is a "root transformation" (or a root transformational operation).
(Emonds 1976: 2-3)

The root transformations I focus on here are: (i) T-to-C movement and (ii) root postposing in Japanese. These two cases appear to be syntactic operations triggered only by the matrix C, not by embedded C.

4.3.1 A Preliminary Study of T-to-C Movement

Some of the root transformations Emonds (1976) discusses in fact involve C as a landing site of movement. For example, T-to-C movement in question sentences and negative inversion:

(6) Wh-questions
a. What **do** you think John bought?
b. *You wondered what **did** John buy.

(7) Negative Inversion
a. <u>In not many years</u> **will** Christmas fall on Sunday.
b. *The employees are happy (that) <u>in not many years</u> **will** Christmas fall on Sunday.
(Emonds 1976: 28-29)

Only in the English matrix clause, T-to-C movement occurs as exemplified in the above examples. Similarly, there are Germanic V2 phenomena involving matrix C but not embedded C. These behaviors can be explained under the current system by saying that these operations are triggered as a consequence of direct Agree by some uninterpretable feature that remains on matrix C which itself acts as a probe. That is, the T-to-C movement in (6) and (7) takes place as a result of direct Agree with the matrix C functioning as a probe. Consider, e.g. (6) suppose that C bearing [uQ] triggers Q-agreement with the wh-phrase “what”.

(8) ***Step1:*** Matrix $C_{[uQ]}$ do(T) “$\text{what}_{[iQ]}$”

Q-agreement

Step2: what do(C) t_{do} t_{what}

EF on C attracts "what" and also "do" as a consequence of Q-agreement.[4]

[uQ] on the matrix C triggers Q-agreement with [iQ] on “what”. Then, as a reflex of the Q-agreement, EF on the matrix C attracts “do” as well as “what”. But notice that C-to-T auxiliary movement is a reflex of Q-agreement. Therefore, the head movement does not take place in embedded CPs whose heads are not allowed to retain valued [uF] under Chomsky/Richards’ deduction predicting that there is neither Q-agreement nor subsequent T-to-C movement in embedded clauses. Chapter 5 further examines T-to-C movement in terms of clause-typing by C.

4.3.2 Root-Landing Movement (Postposing) in Japanese

4.3.2.1 Terminological Issues

Before going into discussion on postposing in Japanese, which is another type of root transformation, this section clarifies some terminological issues arising in the following sections. The data I will mainly focus on in this section is postposing in Japanese as

[4] In Chapter 5, I propose that T-to-C movement is a type of PF-movement as a consequence of Q-agreement triggered by the matrix C in accordance with Boeckx and Stjepanovic (2001) and Chomsky (2001).

illustrated by the following:

(9) Taro-ga kono hon-o kat-ta yo.
Taro-Nom this book-Acc buy-Past Part
"Taro bought this book."

(10) Taro-ga **t** kat-ta yo, **kono hon-o**

In Japanese, which is an SOV language, the object "kono hon-o" can undergo rightward movement as in (10), in which the object is pronounced in sentence-final position. Because of this, this type of movement has been called "postposing" or "right dislocation" etc. Although it does indeed look like (10) has undergone movement "rightward", in some of the previous analyses concerning this phenomena, this type of sentence is instead derived by undergoing "leftward" movement (and subsequent deletion), not rightward movement as we will see in Section 4.3.2.4 in detail. That is, so-called "postposing" in Japanese might be "preposing". Taking into account these issues, it seems better not to refer to "directions" such as right or left in referring to this phenomenon. Therefore, I hereafter call it "root-landing movement" in Japanese to avoid potential confusion. Later in Section 4.3.2.4, I will overview some of the previous analyses of this phenomenon and show some implications for those existing analyses obtained from the analysis I propose below.

4.3.2.2 Basics

Japanese root-landing movement behaves uniquely regarding how far phrases can be moved. As Ross (1967) discusses, rightward movement (in general) obeys the Right Roof Constraint, which disallows an element from moving rightward across a CP boundary. In English, for example, rightward movement strictly obeys this rule as follows:

(11) a. That a serious discussion of this topic could arise here was quite unexpected.

b. [$_{CP}$ That a serious discussion *t* could arise here *of this topic*] was quite unexpected.

c. * [$_{CP}$ [$_{CP}$ That a serious discussion *t* could arise here] was quite unexpected *of this topic*] .

The phrase "of this topic" can land 'within' the subject CP as in (11b) but cannot move across the subject CP as in (11c). However, Japanese root-landing movement seems to exhibit behavior that is the "opposite" of the English cases. Moved phrases have to keep moving to (appear at) the matrix edge in Japanese, but are never allowed to occur at right edge positions of embedded clauses. Thus, while the English cases "allow short movement and ban long", the Japanese cases seems to "ban short and require long". Let us consider the following cases:

Root-Landing Movement

(12) ***Original position***
John-ga [Mary-ga kino **<u>ano koen-ni</u>** itta to] kiita.
John-Nom Mary-Nom yesterday that park-to go-Past C hear-Past
"John heard that Mary went to that park yesterday."

(13) ***Edge of the embedded clause***
*John-ga [[Mary-ga kino **t** itta to] ***<u>ano koen-ni</u>***] kiita.
John-Nom Mary-Nom yesterday go-Past C that park-to hear-Past

(14) ***Edge of the matrix clause***
John-ga [Mary-ga kino **t** itta to] kiita(yo) ***<u>ano koen-ni</u>***
John-Nom Mary-Nom yesterday go-Past C hear-Past that park-to

As presented in (12)-(14), the moved phrase "ano koen-ni" can be phonetically realized at the end of the matrix clause, but not in the intermediate position as in (13). Therefore, researchers have regarded root-landing movement in Japanese as a root transformation. Interestingly, this root transformational property is observed only in root-landing movement but not in (leftward) scrambling in Japanese as indicated by the following:

Leftward Scrambling

(15) ***Original position***

Taro-ga [Mary-ga [John-ga **ano koen-ni** itta to] kiita to] sitta]
Taro-Nom Mary-Nom John-Nom that park-to go-Past C hear-Past C know-Past
"I knew that Mary heard that John went to that park."

(16) ***Edge of the lowest clause***

Taro-ga [Mary-ga [**ano koen-ni** John-ga **t** itta to] kiita to] siita]
Taro-Nom Mary-Nom that park-to John-Nom go-Past C hear-Past C know-Past

(17) ***Edge of the second clause***

Taro-ga [**ano koen-ni** Mary-ga [John-ga **t** itta to] kiita to] siita]
Taro-Nom that park-to Mary-Nom John-Nom go-Past C hear-Past C know-Past

(18) ***Edge of the matrix clause***

ano koen-ni Taro-ga [Mary-ga [John-ga **t** itta to] kiita to] sitta]
that park-to Taro-Nom Mary-Nom John-Nom go-Past C hear-Past C know-Past

In leftward (A'-)scrambling in Japanese, the scrambled phrase "ano koen-ni" can be pronounced at any single edge position, even at the edge of embedded clauses as in (17).

Another important aspect of Japanese root-landing movement is that the movement occurs with certain morphemes/particles which appear only at the matrix C. Two of those morphemes are the sentence-final particles "yo" and "ne" which render root-landing movement sentences more natural/acceptable. Those sentence final particles denote speaker-attitude (i.e. confirmation or agreement from the hearer about some shared knowledge, expressed by "ne" and the speaker's conviction or assertion, expressed by "yo"). These particles appear only in the final position of a matrix clause (cf. Masuoka 1991). Also, Q-morphemes such as "no", which also occur only at the matrix C, render root-landing movement more natural.[5] The contrast in (19) is based on relative judgments.

(19) *Sentence final particle: "yo"*

a. ??Hanako-ga kino t_1 tabe-ta ***ano ringo-o***$_1$
Hanako-Nom yesterday eat-Past that apple-Acc
"John ate that apple yesterday."

[5] I thank Satoshi Tomioka for pointing out that Q-morphemes as well as sentence final particles make root-landing movement data more natural.

b. Hanako-ga kino t_1 tabe-ta yo ***ano ringo-o***$_1$
Hanako-Nom yesterday eat-Past Par that apple-Acc

(20) *Q-morpheme: "no"*
Hanako-ga kino t_1 tabe-ta no ***ano ringo-o***$_1$
Hanako-Nom yesterday eat-Past Q that apple-Acc
"Did Hanako eat that apple yesterday?"

Interestingly, the grammaticality/acceptability of leftward scrambling is not influenced by the presence of those particles. That is, the morphemes at the matrix C appear to play a role in the application of root-landing movement.

4.3.2.3 Japanese Root-Landing Movement is Triggered by C-agreement

Two unique behaviors observed in Japanese root-landing movement have been overviewed, so the next question to be asked is how those peculiar properties are explained within the current framework. Since root-landing movement and certain morphemes at the matrix C co-occur, it seems reasonable to think there is relation between them. Let us consider how to explain the interactions between root-landing movement and the matrix morphemes. In order to capture the dependencies, I suggest that the Agree operation enables a moved phrase and a matrix C-morpheme to establish certain dependencies between them. Based on the matrix wh-question cases discussed in Section 4.3.1, the following feature distribution can be postulated for the matrix morphemes and phrases undergoing root-landing movement:

(21) *Matrix Wh-movement*
a. Matrix C: [uQ]
b. Moved wh-phrases: [iQ]

(22) *Root-Landing Movement*
a. Matrix C: [uOp]
b. Moved phrases: [iOp]

Like [iQ] in wh-movement, [iOp] on a moved phrase in root-landing movement is interpreted by undergoing Agree with [uOp], and then EF on C attracts the phrase (YP) to its edge as follows:

(23) Matrix $C_{\text{[uOp][EF]}}$ $YP_{\text{[iOp]}}$

Agree triggered by [uOp]

YP yo/ne/no t *Phonetic specification of C to "yo/ne"*

(C) *EF on C attracts YP to its edge.*

If [iOp] on a phase does not undergo Agree, the output is regarded as *gibberish* as is in the case of [iQ] based on Chomsky (2007, 2008). [uOp] on C can occur accompanying those morphemes, which can be thought of as the phonetic realization of [uOp] on C. Recall that in Section 4.3.1, I suggested that T-to-C movement in a matrix wh-question takes place as a consequence of Q-agreement triggered by the matrix C. In other words, Q-agreement triggered by C determines the phonetic form of the matrix C. (I will discuss this issue more in detail in Chapter 5.) The same mechanism as the matrix wh-question case is applied to the root-landing movement case illustrated in (23). As a consequence of Op-agreement triggered by the matrix C, its phonetic realization is determined as either "yo" or "ne".

Also, remember under Chomsky/Richards deduction coupled with my proposals here, only the highest phase head can keep [uF] which triggers Agree, given the hypothesis (3). That is, Agree for [iOp] takes place only in the matrix clause in a convergent derivation. In embedded clauses, if [uOp] is retained by C, derivations crash because of stranded valued [uOp] on a phase head. In this sense, root-landing movement in Japanese is similar to successive cyclic wh-movement in English-type languages e.g. "What do you think John bought?"—a moving phrase is purely attracted by EF on a phase head until it finds a probe to agree with.

(24) $[_{CP2}$ what do you $[_{vP2}$ t think $[_{CP1}$ t John $[_{vP1}$ t bought t]]]]

In (24), that is, "what" needs to keep moving until the matrix C, which bears [uQ], agrees with, and attracts, “what”. By agreeing with the matrix C, [iQ] on "what" is finally interpreted. On the way to the matrix CP phase, "what" is attracted purely by EF on v_1, C_1 and v_2. In the same way, in the case of root-landing movement, a moved phrase bearing [iOp] keeps moving until to finds and undergoes Agree with C which bears [uOp]. Although wh-phrases are possibly interpreted by s-selecting verbs i.e. interpreted by embedded C selected by verbs like "wonder", phrases undergoing root-landing movement always require C bearing [uOp], which is allowed only at the matrix C, given Chomsky/Richards deduction. This is why this phenomenon in Japanese is a root transformation. (See Chapter 5 for detailed further discussion of s-selection.)

What about leftward long scrambling? As presented in (15)-(18), leftward scrambled phrases can be phonetically realized at any single phase edge. Recall this phenomenon does not show any interactions with particular morphemes/functional heads, unlike root-landing movement. While root-landing movement is similar to wh-movement in a certain sense mentioned above, leftward long scrambling in Japanese behaves differently in that phonetic realization of copies is not as restricted as in root-landing movement. Saito (1985, 2004) suggests that long leftward scrambling is semantically vacuous, so that it has to be analyzed differently from wh-movement, which is semantically substantive. He proposes that scrambling is not something triggered by EPP/EF, while wh-movement is. On the other hand, Hiraiwa (2010) following insights from Miyagawa (1997), suggests that A'-scrambling is EF-driven and targets phasal edge positions, which enables him to induce the proper binding condition from the cyclic spell-out system. (See Hiraiwa 2010 for more detail.) I agree with Saito's insight in that leftward long scrambling behaves differently from English-type wh-movement. Especially in my view, leftward A'-scrambling has to be distinguished from wh-movement and also from root-landing movement in terms of possible pronunciation sites. As mentioned in (23), matrix wh-movement and root-landing movement in Japanese require interpretation of [iQ] and [iOp] by agreeing with probes bearing [uQ] and [uOp], respectively. Therefore, Goals bearing an [iQ] or [iOp] feature "need" to move to positions close-enough to agreeing probes (i.e. matrix C) in convergent derivations, which leads to restricting pronunciation sites. On the other hand, I also agree with Hiraiwa's proposal in that phasal edge positions are the only loophole for successive cyclic movement under the cyclic Spell-Out/Transfer system. Taking into consideration all the insights above, I suggest that Japanese leftward long scrambling is purely driven by EF and targets phasal edges and is distinguished from other types of movement such as wh-movement and root-landing movement in terms of (non)necessity of agreement. That is, leftward long scrambling does not involve any features requiring Agree, in contrast to wh-movement and root-landing movement. Since every phase head bears EF (although its satisfaction is not obligatory), every single phasal edge can be a possible final landing site i.e. possible pronunciation position (in overt movement).[6]

(25) a. Long Leftward scrambling: EF
b. Root-landing movement: EF + [iOp]
c. English wh-movement: EF + [iQ]
(to be discussed in Chapter 5 in detail.)

Given (25), it is explained why root-landing movement is a root transformation but

[6] Here, I put aside Edge-vP as a possible pronunciation site.

leftward long scrambling is not. Also, I would like to again draw the readers' attention to the fact that probing by a phase head can be triggered only by matrix C in a convergent derivation given the hypothesis (3) coupled with the Chomsky/Richards deduction. If on track, the unique behavior of root-landing movement in Japanese can be explained without additional stipulations. Much further research is needed to determine how much of the crosslinguistic uniqueness of root T can be illuminated this way.

4.3.2.3 Implications for Previous Analyses of Root-Landing Movement in Japanese

I have overviewed how the root transformational property in Japanese root-landing movement is explained and distinguished from leftward long scrambling. As I noted in Section 4.3.2.1, there have been various types of previous approaches to root-landing movement phenomena. Among those approaches, the following two are prominent: (i) rightward movement analyses (see Haraguchi 1973, Simon 1989, Takano 2005[7]) and (ii) leftward movement and subsequent deletion analyses (see Kuno 1978, Abe 1999, Tanaka 2001, Kato 2007, Takita to appear[8]).[9] In the following few paragraphs, I briefly review the two competing approaches to root-landing movement phenomena and then conclude that the system proposed here is compatible with Abe's (1999) leftward operator movement analysis, as opposed to Tanaka's (2001) leftward scrambling analysis.

In rightward movement analyses of root-landing movement, phrases undergo movement literally "rightward" as follows:

---

[7] While Haraguchi (1973) and Simon (1989) suggest that root-landing movement is a syntactic rightward movement, Takano (2005) proposes that it is a PF rightward movement.

[8] Although these researchers all propose that root-landing phenomena involve leftward movement, there are some differences between them.

[9] Although I limit my focus only to these two approaches, another analysis worth mentioning is the non-movement analyses suggested by Inoue (1978).

(26) *Rightward movement analysis*
[$_{XP}$ [$_{YP}$ t_1] ZP_1]

This type of analysis is mainly discussed in Haraguchi (1973) and Simon (1989), which demonstrate that ZP in (26) undergoes movement i.e. is not base-generated, based on the fact that root-landing movement is sensitive to island effects. The logic there is as follows: If root-landing movement shows island effects as leftward scrambling does, some sort of A'-movement takes place. Consider the following data testing the coordinate structure constraint:

(27) [[Sofa-to isu-no] aida]-ni table-o oi-ta yo.
Sofa-and chair-Gen between-Loc table-Acc put-Past Part
"(I) put a table between a sofa and a chair."

(28) *Root-Landing Movement*
a. *[[**t** isu-no] aida]-ni table-o oi-ta yo, **sofa-to**
b. *[[Sofa to **t**] aida]-ni table-o oi-ta yo, **isu-no**

(Simon 1989: 124)

(29) *Leftward Scrambling*
***Isu-no** [[Sofa-to **t**] aida]-ni table-o oi-ta yo.

In (28), one of the coordinated elements undergoes root-landing movement. In (29), the coordinated element undergoes leftward scrambling. In each case, the resulting sentence is unacceptable. Since root-landing movement shows island sensitivity as leftward scrambling does, these data indicate that root-landing movement involves movement, not base-generation. This is (i) the right movement analyses.

Kuno (1978), Abe (1999) and Tanaka (2001) propose the other approach to root-landing movement, which involves leftward movement: (ii) the leftward movement and subsequent deletion analyses. Under this type of analysis, root-landing movement sentences are derived as follows:

(30) Leftward movement & deletion analysis

Step1: *Repetition*

Sentence1: [$_{YP1}$ *pro*$_1$] Sentence2: [$_{YP2}$ ZP$_1$]

Step2: *Leftward movement of ZP$_1$ in YP$_2$*

[$_{YP1}$ *pro*$_1$] [$_{XP}$ ZP$_1$ [$_{YP2}$ t$_1$]]

Step3: *Deletion of YP$_2$*

[$_{YP1}$ *pro*$_1$] [$_{XP}$ **ZP$_1$** ~~[$_{YP2}$ t$_1$]~~] *YP$_1$ and ZP$_1$ are pronounced.*

(31) Taro-ga **t** tabe-ta yo, **ano ringo-o**
Taro-Nom eat-Past Part that apple-Acc
"Taro ate that apples."

(32) The derivation for (31):
Sentence1: [Taro-ga pro$_1$ tabe-ta yo].
Sentence2: **Ano ringo-o$_1$** ~~[Taro-ga **t** tabe-ta yo]~~.

Under this analysis, the identical two sentences YP1 and YP2 are generated as in Step1 in (30). ZP never undergoes rightward movement in contrast to the previous approach. Rather, it undergoes leftward movement as in Step 2 and then YP$_2$ out of which ZP is moved is deleted observing the identity requirement governing deletion as in Step 3. The entire sentence YP1 and the remnant from deletion i.e. ZP are pronounced as a single sentence.[10] (32) is the actual derivation of (31) under the leftward movement and subsequent deletion approach. The phrase "ano ringo-o", which is co-indexed with *pro* in Sentence 1, undergoes leftward movement. After that, based on the identity requirement of deletion, TP in Sentence 2 is deleted. According to Tanaka (2001), in a sense, this approach is similar to sluicing, which is TP-deletion after wh-movement as discussed in e.g. Merchant (2001):

[10] It is unclear how the connection between the two identical sentences is constructed. In the case of (32), "ano ringo-o (that apple-Acc)" is a member of Sentence 2 but at some point of the derivation, the phrase behaves like a member of the other sentence. Another obscure part of this approach is the “repetition” mechanisms assumed to derive “multiple copies” of exactly the same sentence.

(33) John bought something. I do not know what.
(Rep: John bought something. I do not know what ~~[John bought t]~~.
(Tanaka 2001: 559)

The identity requirement of deletion is satisfied inter-clausally in (33). Then, after leftward/wh-movement applies to the wh-phrase "what" in the latter sentences, TP-deletion is applied.

As supporting evidence for this type of analyses, Tanaka (2001) focuses on some behavioral differences between English rightward movement and Japanese root-landing movement. As briefly discussed in 4.3.2.2, English rightward movement obeys the right roof constraint, so that no element can move rightward across a clause boundary. On the other hand, Japanese root-landing movement is upward unbounded since the landing site of a moved phrase is limited to the matrix edge. More evidence for this approach comes from the status of a gap left behind in root-landing movement. Kuno (1978) shows that a gap created in root-landing movement is not a trace but an empty proform. (34) is the output configuration under the leftward movement analysis (discussed in (30)) and (35) is the output configuration from the rightward movement analysis (discussed in (26)).

(34) *Leftward movement and deletion analysis*
Sentence 1: **[$_{\text{YP1}}$ *pro*$_1$]**
Sentence 2: [$_{\text{XP}}$ **ZP$_1$** ~~[$_{\text{YP2}}$ t$_1$]~~]
(Ex.) Taro-ga ***pro*$_1$** tabe-ta yo, **ano ringo-o$_1$** (ZP).
Taro-Nom eat-Past Part that apple-Acc
"Taro ate that apple."

(35) *Rightward movement analysis*
[$_{\text{XP}}$ [$_{\text{YP}}$ **t$_1$**] **ZP$_1$**]
(Ex.) Taro-ga **t$_1$** tabe-ta yo, **ano ringo-o$_1$** (ZP).
Taro-Nom eat-Past Part that apple-Acc
"Taro ate that apple."

Notice that in (34), there are two identical but independent sentences. Although ZP is moved leftward in Sentence 2, YP$_2$ including the trace of ZP is deleted. That is, the gap in the object position is not a trace but a proform.[11] On the other hand, the rightward movement analysis has a different output configuration shown in (35). The object gap in this case is a trace/copy which is left behind after rightward movement of "ano ringo-o". That is, these two approaches are distinguished in terms of what empty category occupies the object gap: in (34) *pro*, in (35) a trace. Keeping this discrepancy in mind, consider the following cases:

(36) John-ga Mary-ga LGB-o yonda to itta.
John-Nom Mary-Nom LGB-Acc read-Past C say-Past
"John said that Mary read LGB."

(37) *Leftward scrambling*

a. **LGB-o** [John-ga [Mary-ga **t** yonda to] it-ta].
LGB-o John-Nom Mary-Nom read-Past C say-Past
"LGB, John read LGB."

b. ***LGB-o** John-ga Mary-ga **LGB-o** yonda to itta.
LGB-o John-Nom Mary-Nom LGB-Acc read-Past C say-Past
"LGB, John read LGB."

(38) *Root-landing movement*

a. John-ga Mary-ga **t** yon-da to itta yo, **LGB-o**.
John-Nom Mary-Nom read-Past C say-Past Part LGB-Acc
"John read LGB, LGB."

b. John-ga Mary-ga **LGB-o** yonda to itta yo, **LGB-o**.
John-Nom Mary-Nom LGB-Acc read-Past C say-Past Part LGB-Acc
"John read LGB, LGB."

(Tanaka 2001: 552)

In (37a), the object "LGB-o" undergoes leftward scrambling. In (37b), the overt lexical item "LGB-o" is inserted into the object gap/original position of the moved "LGB-o", which renders the sentence unacceptable. In (38a), the phrase "LGB-o" undergoes root-landing movement. Again, in (38b), the overt lexical item "LGB-o" is inserted into the original position of the moved phrase, in which case the sentence is acceptable. The contrast between (37b) and (38b) implies that there is some difference with respect to

[11] Takita (to appear) argues against the claim that the gap is *pro* but suggests that the gap is created as a consequence of ellipsis.

what is left behind after each of the movements: one allows the presence of the overt lexical item while the other does not. One possible account of this contrast is to attribute different properties to the (object) gaps. In (38a), the gap is, in fact, filled with a *pro*, not a trace, as the leftward movement analysis predicts. In (37a), on the other hand, a trace left behind by scrambling occupies the gap position. In other words, since the gap in (38a) is filled not with a trace but with a null pronoun, it is not unreasonable to think that the overt counterpart can also appear in the same position. Meanwhile, since the gap in (37a) is a trace, the insertion of the overt lexical item nullifies a trajectory of movement of "LGB-o". This is why (38b) allows the occurrence of the overt lexical item while (37b) does not. What is crucial to this analysis is that the object "gaps" differ. If the rightward movement analysis is adopted, the explanation by appeal to the difference between pro and a *trace* does not go through because the gap position under this analysis is filled by a trace left behind by rightward movement as depicted in (35). Therefore, the contrast at issue is not straightforwardly explained by the rightward movement analysis.

In addition, even when the overt lexical item occupies the gap position in root-landing movement, Tanaka (2001) shows that island sensitivity is still observed as follows:

(39) ?*John-ga [[Mary-ga **Bill-ni** age-ta] hon]-o nusun-da yo, **Bill-ni**
John-Nom Mary-Nom Bill-Dat give-Past book-Acc stole Part Bill-Dat
"John stole the book which Mary gave to Bill, to Bill."
(Tanaka 2001: 556)

In order to explain the paradigms in (39) (including the contrast between (37) and (38)), the analysis of leftward movement and subsequent deletion works straightforwardly. That is, there is movement causing island violation in Sentence 2 but the gap position in Sentence 1 is filled with an overt lexical item as follows:

(40) The derivation for (39)
Sentence 1:
John-ga [[Mary-ga **pro**$_1$ age-ta] hon]-o nusun-da yo
Pro is replaced by "Bill-ni" after satisfying the identity requirement of deletion.

Sentence 2:
Bill-ni$_1$ ~~John-ga [[Mary-ga **t**$_1$ age-ta] hon]-o nusun-da yo.~~
Island violation[12]

Why does an island violation show up in (39)? The answer is that leftward scrambling of "Bill-ni" move across the complex NP island. Why can the gap position be replaced by the overt lexical item in (39)? The answer is that the gap position is filled with a null pronoun, not with a trace, so that the overt counterpart can also appear in the same place. This is a brief overview of the leftward movement approach to the root-landing movement discussed in Kuno (1978), Abe (1999) and Tanaka (2001).

Based on the above discussion, the leftward movement and subsequent deletion analysis appears to obtain more empirical support than the rightward movement analysis, although there are some unclear technical issues (see fn. 9).

The analysis I presented in Section 43.2.3 appears to support the leftward movement analysis. A piece of evidence to support this position is that rightward movement in general does not involve feature agreement since it is regarded as PF-movement or stylistic movement, not syntactic movement. Consider the following case:

(41) Many books with stories were sold that I wanted to read.
(Rep: [$_{NP}$ Many books with [$_{PP}$ stories t$_1$]] were sold [that I wanted to read]$_1$.)
(Chomsky 1986b: 40)

According to Chomsky (1986b), the extraposed CP moves across the barriers PP and (by inheritance) NP on the way to the final landing site, which causes a subjacency violation.

[12] This derivation implies that there is no Complex NP island repair by deletion in Japanese. (cf. Lasnik 2001)

If extraposition is a syntactic movement, the sentence in (41) should be ungrammatical by subjacency. But, this is not the case. Therefore, Chomsky (1986b) argues that extraposition is a PF movement, not a syntactic movement. The analysis of root-landing movement in Japanese proposed in this thesis is that moved phrases undergo agreement with the matrix C. That is, if root-landing movement in Japanese is also a type of rightward movement as extraposition in English is, unvalued [uOp] on the matrix C is never valued within narrow syntax and also [iOp] on a moved element is never interpreted. The prediction is that root-landing movement is never derived under the proposed system. This is not a welcome result. If root-landing movement is in fact derived by leftward movement as depicted above in this section, however, it makes sense why moved phases need to undergo agreement with a certain phase head, as wh-movement does. In this sense, the proposed analysis is compatible with the leftward movement analysis of root-landing movement, not the rightward movement analysis.

Pushing further forward the discussion here, it is controversial that leftward movement involved in root-landing movement is operator movement (cf. Abe 1999) or scrambling (cf. Tanaka 2001). The proposed system supports the former view in the sense that [iOp] on moved phrases needs to agree with a certain head, namely the matrix C, but not in leftward scrambling. I also demonstrated that root-landing movement and leftward scrambling behave differently in terms of possible pronunciation sites, so that scrambling is a pure EF movement in contrast to root-landing movement as matrix C-agreement and EF movement. In addition, in the leftward scrambling analysis, there is no reason why the entire representation except a moved phrase has to be always deleted. For example, consider the following case:

(42) a. John-ga Mary-ga **ano koen-ni** itta to kiita yo.
John-Nom Mary-ga that park-to go-Past C hear-Past Par
"John heard that Mary went the park."
b. John-ga Mary-ga __ itta to kiita yo. **ano koen-ni**
John-Nom Mary-Nom go-Past C hear-Past Par that park-to

(43) **Derivation 1**
Sentence1: John-ga [Mary-ga *pro* itta to] kiita yo.
Sentence2: **ano koen-ni**$_1$ ~~[John-ga [Mary-ga t$_1$ itta to] kiita yo]~~

Based on the leftward movement analysis illustrated in (30), after "ano koen-ni" undergoes leftward movement, the entire TP within Sentence 2 is deleted just like sluicing. As a result, (42b) is generated. If the leftward movement is leftward "scrambling", however, there should be another derivational option because leftward-scrambled phrases are phonetically realizable at any single phase edge in contrast to root-landing movement, which requires Agree. Therefore, there is no reason why "ano koen-ni" always moves to the edge of the highest clause. Rather, it can move and stay at the edge of the intermediate clause as follows, which is a perfect sentence involving leftward scrambling:

(44) John-ga **ano koen-ni**$_1$ [Mary-ga t$_1$ itta to] kiita yo.
John-Nom that park-to Mary-ga go-Past C hear-Past Par
"John heard that Mary went the park."

That is, there should be a derivational possibility that deletion applies to the representation in (44) as follows:

(45) **Derivation 2**
Sentence1: John-ga [Mary-ga *pro* itta to] kiita yo.
Sentence2: [John-ga **ano koen-ni**$_1$ ~~[Mary-ga t$_1$ itta to]~~ kiita yo]

(46) *John-ga Mary-ga itta to kiita yo, John-ga **ano koen-ni** kiita yo
John-Nom Mary-Nom go-Past C hear-Past Par

(46) is the resulting output from Derivation 2. After the phrase "ano koen-ni" undergoes

scrambling to the edge of the embedded CP, TP in Sentence 2 is deleted satisfying the identity requirement of deletion. If leftward "scrambling" is applied, that is, it wrongly predicts that (46) is grammatical.

The analysis I proposed in Section 4.3.2.3 does not face this problem because [iOp] on the phrase "ano koen-ni" has to agree with [uOp] on the matrix C in a convergent derivation, so that the phrase is attracted by EF to the matrix edge. Therefore, the phrase always moves to the matrix edge in a convergent derivation because of undergoing Agree triggered by [uOp] on the matrix C excluding the possibility in (45) i.e. avoiding overgeneration. In this sense, as I mentioned earlier, root-landing movement is similar to wh-movement, especially similar to a case like "What do you think John bought?". Therefore, the proposed system supports the operator movement approach discussed in Abe (1999), not the scrambling approach discussed in Tanaka (2001).

4.4 Summary

This chapter has further discussed Chomsky/Richards Value-Transfer Simultaneity analysis especially focusing on root clauses. In Section 4.2, I pointed out a potential problem with the current Transfer system, which applies only to a phase head complement: The root edge is never included in the domain of any phase. If the phasal complement Transfer system is strictly maintained, there is no way to ever transfer the root edge. In order to solve this problem, I advanced a working hypothesis in (3) regarding the root edge, repeated as (47) below:

(47) Working Hypothesis (to be revised in Chapter 5):
Unlike all other phases, the highest phase head and its edge are transferred "for free" along with the phase head complement.

I also demonstrated that (47) makes an interesting prediction in conjunction with the Chomsky/Richards analysis: valued [uF] can be stranded on the edge but only at the matrix phase head in a convergent derivation since it is transferred “for free” along with the phase head complement. In Section 4.3, I showed that the prediction born out of (47) may help explain the idiosyncracies root transformations, as represented by T-to-C movement and root-landing movement in Japanese. In addition, I revealed some important theoretical and empirical implications obtained from the proposed analysis.

However, there is one clear problem regarding the hypothesis in (47). Even though it is strongly empirically motivated, since matrix edges DO undergo phonetic and semantic interpretations, it is not at all clear why such a special formal treatment applies only in the matrix phase i.e. (47) is ad hoc. In other words, it is not clear how the computational system "knows" which CP phase is matrix or embedded without look-ahead. I will address this very issue in the next chapter. In Chapter 5, I reformulate a Transfer system which enables us to accommodate the effects of hypothesis (47) but without stipulating it.

Chapter 5

A Generalized Theory of Feature Inheritance: How does Transfer Send What?

5.1 Framing the Issues:

This chapter further investigates aspects of the Chomsky/Richards Value-Transfer simultaneity analysis, focusing more closely on the formal mechanisms of the Transfer operation. One of the key assumptions in the Chomsky/Richards' deduction is that Transfer is applied only to a phase head complement while the phasal edges are stranded after Transfer. Given that, a phase head cannot retain valued [uF] in a convergent derivation (because valued [uF] becomes indistinguishable from [iF] in the eyes of Transfer, hence is not removed by Transfer thereby causing crash of the derivation at CI.) This is why Richards deduces that [uF] has to be discharged from a phase head to a lower head in a convergent derivation. That is, the assumption that Transfer applies only to a phase head complement plays a central role in the Chomsky/Richards' deduction.

However, one obvious question is WHY Transfer is applied only to a phase head complement. Is there any logical reason which forces Transfer to work that way? Or is it a mere albeit empirically motivated stipulation which is in effect a construction or category specific stipulation? Recall one of the clear reasons for the assumption seems to be that phase edge positions are indeed used for escape hatches of successive cyclic movement. Therefore, the edges have to be available for Internal Merge triggered by EF on phase heads. On the other hand, it is obvious that phase edges are not used as escape hatches in each and every derivation. If providing an escape hatche is the only reason for limiting Transfer to a phase head complement, then, in derivations without wh-movement there is no motivation to leave phasal edges behind after Transfer. And recall if an operation is not necessary, then under SMT, it is prohibited.

In this chapter, I give close consideration to the mechanisms of Transfer answering the following three questions: (i) Why does Transfer send exactly the phasal complement to the interfaces? (ii) What is left behind in the narrow syntax representation after Transfer? and (iii) How are transferred pieces correctly reassembled in the phonological and semantic components as is necessary for "full sentential" representation? Especially in addressing (i), I also discuss problem of root clause Transfer, which I pointed out in Chapter 4, in which I advanced a working hypothesis which states that <u>a root edge is sent to the interfaces along with the phase head complement by Transfer.</u> As noted, under the standard assumption that phase head complements are transferred, leaving the edge behind, there is no way to ever transfer the root edge to the interfaces. I examined how this hypothesis in effect necessitated by the theory explains the special status (predicted by the theory) of root transformations, to which "special" Transfer application must apply. This chapter attempts to explain the working hypothesis for root clause Transfer without stipulating anything special about root clauses.

The chapter is organized as follows: Section 5.2 discusses the formal mechanisms of the Transfer operation and demonstrates that Transfer need not be stipulated to apply only to the phase head complement but instead can be applied to any XP (“Freely Transfer”) a simplification. The satisfaction of s(semantic)-selectional requirements and clause-typing will be shown to play important roles in the Transfer system reformulated in this section. Section 5.3 investigates another formal property of Transfer and considers what exactly Transfer sends to the interfaces and what is left behind in narrow syntax after Transfer. I propose that only label copies are left behind after Transfer, so that their contents (i.e. internal structures) are "lost" from narrow syntax truly explaining the phase impenetrability condition/PIC effects i.e. the syntactic objects within the transferred domain are absent from the narrow syntax, thereby explaining their syntactic inertness. The label copy system is argued to be a natural implementation of Transfer observing several conditions imposed on narrow syntax such as the No-Tampering condition and the recoverability condition. In addition, I show that the proposed system enables us to explain, without a stipulation, the phase impenetrability condition/stipulation. Section 5.4 examines a thus far unnoticed problem confronting phse-based derivation: How can phrases bigger than phases ever undergo Internal Merge? I propose a solution which gives independent support for my label-copy Transfer system which in turn provides an approach to the very general problem of how transferred pieces are reassembled for global computation. Section 5.5 summarizes and concludes this chapter.

5.2 The Formal Mechanisms of the Transfer Operation

This section clarifies some problems the current Transfer system confronts and presents reformulated mechanisms of Transfer also taking into account the issues raised in Chapter 4 regarding Transfer of root clauses.

5.2.1 Why Transfer the Phase Head Complement? Why Not the Entire Phase?

Since Chomsky (2000), it has been assumed that the representations constructed in narrow syntax are sent to the interfaces phase-by-phase by iterated application of Transfer. The portion Transfer sends to the interfaces is the complement of a phase head, so that only a phase edge remains after Transfer. Chomsky (2004) says:

"A natural condition, which permits spell-out of root phrases and allows for meaningful cyclic computation, is that β [= a phase head complement, MO] must be spelled-out at PH [=phase, MO], but not the edge: that allows for head-raising, raising of Predicate-internal subject to Spec-T, and an "escape hatch" for successive-cyclic movement through the edge."

(Chomsky 2004: 108)

As empirical motivations for Transfer of a phase head complement, Chomsky raises the following three kinds of inter-phasal dependencies: head-raising, raising of Predicate-internal subject to Spec-T (which will be discussed later in Section 5.2.2) and successive-cyclic movement. However, head-raising and successive cyclic movement seem not to be strong evidence to support the view that phase edges have to be left behind by Transfer because neither of them take place in every derivation. For example, V-raising to T does not take place if T is filled with auxiliaries such as "can" or "will". That is, head-raising does not fully motivate the current Transfer system. Also with respect to successive-cyclic movement, providing escape hatches seem not to be a strong reason because, again, escape hatches are not in fact necessary in each and every derivation. In other words, the question is whether there is any evidence that edge positions are left behind after Transfer when successive movement does <u>not</u> apply. Consider the following sentences:

(1) What do you think John bought?
(2) You think John bought the book.

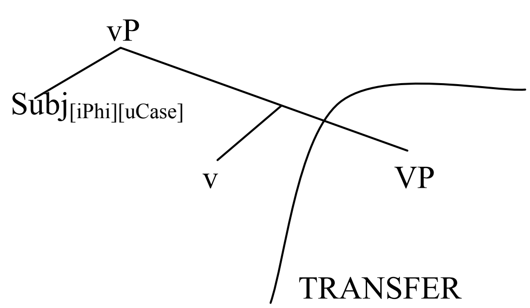
In (1), "what" moves from the embedded clause to the matrix clause (i.e. inter-phasally), so that the edges of the phases need to be available as escape hatches given PIC. In (2), however, no element moves across any phase boundary. If an escape hatch is a reason why Transfer is limited to a phase head complement, then sentences such as (2) raise a question: What is transferred and why? What happens if the entire embedded CP undergoes Transfer in (2)? One clear problem is that there is no way for the matrix verb "think" and the complement CP to be in a sister (Merger) relation since the complement CP does not exist within narrow syntax when "think" is selected from the Lexical Array. (The issue of what is left behind after Transfer will be discussed in Section 5.3 below.) That is, the matrix verb ("think") and the embedded CP ("John bought the book") are no longer recognized as a single constituent but rather are recognized as two independent syntactic objects since there is no relation constructed by Merge between them. In addition, given Epstein's (1999) derivational c-command, the verb does not c-command the embedded CP because of no Merge relation between them. Obviously, this is not a welcome result e.g. in the case of inter-phasal binding relations such as Condition C effects.

In a nutshell, escape hatches are necessary at least for some of the cases such as (1). On the other hand, there are some cases such as (2), which do not require escape hatches but still need a phase edge to construct a Merge-created sister relation between a matrix verb and its complement CP. Theses observations imply that there is another reason (besides "escape") why Transfer must leave phase edges behind. However, embedded clauses require Transfer to send (only) phase head complements, while matrix clauses require Transfer to send the entire phase, as discussed in Chapter 4. That is, we need a Transfer system such that: (i) embedded phase edges are left behind after Transfer and (ii) matrix phase edges are sent to the interfaces along with the phase head complement. In addition, ideally, (i) and (ii) are implemented without any matrix vs. embedded stipulation--i.e. we seek to induce the asymmetry from higher principles or conditions, which are independently motivated. In the next three sections (5.2.2 to 5.2.4), I propose that (i) and (ii) are both induced from the s-selectional requirement and clause-typing mechanisms. I also demonstrate that under the proposed system, Transfer can apply freely--it is no longer limited to a phase head complement.

5.2.2 Embedded CP/vP: S-Selection and the Phonological Specification of C

This section concentrates on the issues of vP and embedded CP. Let us first consider Transfer of vP:

(3)



In the case of the vP phase, in order to yield convergence, Transfer must apply to VP stranding the subject which bears [uCase] to be "subsequently" valued by T. If the entire phase i.e. vP is transferred, unvalued [uCase] causes crash of the derivation. At least for the vP phase, that is, there is no need to stipulate that Transfer is limited to a phasal complement because VP is the only possibility in a convergent derivation, otherwise causing crash.[1]

Next, consider embedded CP. As mentioned earlier, providing escape hatches does not fully explain that Transfer strands the edge of the embedded CP since there are cases which do not need escape hatches in (2). Why do phase edges of embedded CP always have to be left behind? The key to answering this question is the relationship between a verb (or any head: the belief [$_{CP}$ that [$_{TP}$]]) and its complement CP. There are at least two kinds of relations between them: (i) s-selection and (ii) phonetic realization of C. Let us examine both.

The first relation between a verb and its complement CP concerns the s(emantic)-selectional requirement of a verb. As Chomsky (1986a) discusses, s-selectional properties of verb heads have to be specified in the lexicon, as opposed to c(ategorial)-selection which is deduced from Case Theory, based on Pesetsky (1982). Each verb has its selectional properties, e.g. verbs such as "ask" and "wonder" select interrogative propositions as their complements as follows:

(4) a. I asked what time it is.
 b. *I asked (that) John bought the book.
(5) a. I wondered what time it is.
 b. *I wondered (that) John bought the book.

(Pesetsky 1982: 183)

In the (a)-sentences, verbal s-selection is satisfied by Merge of the interrogative propositions to the verb but in the (b)-sentences, the requirement is not met, so that the sentences are regarded as ungrammatical. These s-slectional properties, however, do not bear a one-to-one correspondence to specific syntactic categories as Grimshaw (1979) discusses.

[1] One might wonder why v' is not an option of the unit of Transfer. I will discuss this issue in Section 5.2.5 below.

(6) I asked the time.
(7) *I wondered the time.

(Pesetsky 1982: 183)

In the case of "ask", its s-selectional requirement can be satisfied by DP but the requirement of "wonder" is not satisfied by DP. Since a single type of s-selection ranges over various categories, Grimshaw (1979) concludes that s-selection has to be assumed independently and c-selection can be subsumed by s-selection, given canonical structural realization:

(8) If a verb (or other head) s-selects a semantic category C, then it c-selects a syntactic category that is the "Canonical Structural Realization of C" (CSR (C)).

(Chomsky 1986a: 87)

In the case of "ask" and "wonder", for example, each s-selects interrogative propositions, then it c-selects CP or N/DP. That is, CSR(proposition) is CP or N/DP. Under (8), the lexicon needs to specify only s-selection because s-selection specified in the lexicon determine what categories are required to occur in certain contexts based on CSR. However, this scenario is not sufficient to explain why "wonder" which is CSR(proposition) does not allow N/DP but "ask" allows it. Pesetsky (1982) further pushes forward Grimshaw's insight and suggests: while s-selection is independently assumed, c-selection restrictions can be induced from Case Theory, i.e. eliminated from the lexicon. That is, (7) is ungrammatical because "wonder" (as an "intransitive verb") does not have the ability to assign Case to the object N/DP. On the other hand, a "transitive verb" such as "ask" can assign Case to the object NP. Therefore, (6) observes Case Theory. The following paradigm also supports this view:

(9) It was asked what time it is.
(10) *It was wondered what time it is.

(Chomsky 1986a: 89)

Passives are allowed only with transitive verbs, but not intransitive verbs. This data indicates that "ask" can assign accusative Case, in contrast to "wonder", as also motivated by the argument concerning the ability of accusative Case-assignment in (6) and (7). Therefore, s-selection of "ask" and "wonder" requires interrogative propositions realized by CP or N/DP but Case Theory rules out the N/DP option for "wonder". That is, by assuming s-selection and Case Theory, the verb-complement co-occurrence presented in (4)-(7) can be explained. In other words, s-selection of a verb has to be specified in the lexicon and plays an important role in complementation. As for the current concern with Transfer, the discussion presented here implies that the s-selectional requirement of a verb must be somehow "transmitted" to its complement CP to construct a certain dependency between them. But to do this, at least some piece of a "soon to be" complement CP has to remain within narrow syntax, so that the verb can be externally-merged to it within the derivation and establish the dependency. It seems plausible then that phase edges of embedded CP are left behind after Transfer to satisfy s-selection imposed by a higher verb.

The second case illustrating interactions between a verb and its CP complement concerns the phonetic realization of C. There are several cases in which the phonetic realization of C is determined by a verb selecting it. It appears that these data also indicate that a relation between a verb and its complement CP is somehow constructed within narrow syntax (i.e. before morpho-phonetic component). First, consider the following case:

(11) *Cape Verdean Creole*

a. Joao pensa ki/*Ø Maria kunpra libru.
John think C Mary bought book
"John thinks Mary bought the book."

b. Joao **fra**-m ma/*ki/*Ø Maria kunpra libru
Joao **told**+me **C** Maria bought book
"John told me Mary bought the book."

(Obata and Baptista, 2009)

In Cape Verdean Creole, phonetic realization of embedded C differs depending on what kind of verb selects it. While most verbs select "ki" as the phonetic realization of C as in (11a), illocutionary verbs such as "tell" require "ma", not "ki", as the phonetic realization of C as in (11b). That is, there is something more than s-selectional relations between a verb and its complement CP. In addition to s-selection, that is, V plays a role in determining the phonetic realization of C. This is not limited only to Cape Verdean Creole. Rather, many languages behave in the same way. In English, for example, when a verb such as "think" takes a CP complement, phonetic realization of C is optional i.e. either null or "that", as in (12a). On the other hand, a verb like "whisper" and also N require C to be overt, as illustrated in (12b) and (13).

(12) a. John thinks (that) Mary bought the book.
b. John whispers ??(that) Mary bought the book.
(13) John knows the fact *(that) Mary bought the book.

These data also show that some kind of dependencies need to be constructed between a verb and its complement CP and some instructions with respect to phonetic realization of C need to be sent to the morpho-phonetic component. In some cases, furthermore, phonetic realization of C is dependent on s-selection. That is, s-selection and phonetic realization of C are not independent issues but rather are closely related:

(14) *English*
a. John wonders whether/if Mary bought the book.
b. John wonders what Ø Mary bought.
(15) *Japanese*
a. John-ga [Mary-ga hon-o kat-ta to/*ka] omot-teiru.
John-Nom Mary-Nom book-Acc buy-Past C think-Prog
"John thinks Mary bought books."

b. John-ga [Mary-ga hon-o kat-ta *to/ka] Sam-ni tazune-ta.
John-Nom Mary-Nom book-Acc buy-Past C Sam-to ask-Past
"John asked Sam whether Mary bought books."

In English (14), a verb such as "wonder" takes Q-complement CP whose head is phonetically realized as "whether" or "if". Also, if wh-movement takes place within the CP complement, C has to be null (i.e. one manifestation of the "doubly-filled COMP filter" in English). In Japanese (15), C selected by a verb like "think" has to be marked with a declarative marker "to" while C selected by a verb like "ask" has to be marked with a question marker "ka".[2]

The discussion in this section indicates that a verb and its complement CP exhibit s-selection and phonetic-realization-of-C dependencies. This is sufficient motivation for Transfer stranding phasal edges of embedded CP (and also vP) have to be left behind after Transfer in convergent derivations. Given the proposed mechanism, providing escape hatches is not the fundamental reason for the phase-head-complement Transfer but rather is only a by-product of satisfaction of s-selection requirements and the phonetic specification of C.

The next step to take is how to formalize the observations in this section.

5.2.3 V- to-C Feature-Inheritance and Subsequent Phonetic Specification of C

This section provides a formal analysis of the data discussed in the last section. In Chapter 2 through 4, I specifically discussed aspects of the C-to-T feature-inheritance system: features on T (and V) are inherited from C (and v). As Chomsky (2008) mentions, one motivation for inheritance is that there is no need to distinguish phi-agreeing T and

[2] With respect to phonetic realization of C, as is well-known matrix C behaves differently from embedded C. I discuss this issue in Section 5.2.4.

raising T in the lexicon. Rather, they are predictably differentiated by appeal to whether C is merged or not. If C is merged, that is, T inherits [uPhi] from C, which triggers Phi-Agree and Case-valuation. If C is not merged, on the other hand, T simply does not receive any agree features, so that neither agreement nor Case-valuation takes place. Depending on what selects TP, T behaves differently. Therefore, the features of T are contextually determined under this system, and so need not specified stipulatively in the lexicon.

I propose that the same formal analysis can be extended to the relation between V and C since phonetic realization of C is dependent on what kind of head selects the CP. Also, the concept of the s-selectional requirement fits the essence of "inheritance" well in that a verb imposes a certain clause-type on its complement. Let us see how V to C feature inheritance is carried out. First, based on Chomsky (1986b), an s-selectional property of V is specified in the lexicon along with its categorial feature:

(16) a. **think** $\begin{bmatrix} \text{V} \\ \text{Dec} \end{bmatrix}$ b. **wonder** $\begin{bmatrix} \text{V} \\ \text{Q} \end{bmatrix}$

Each of the verbs has the above specification in the lexicon. [Dec] stands for s-selection requiring a declarative complement and [Q] stands for s-selection requiring an interrogative complement. Notice that verbs have to be "neutral" with respect to clause-typing such as declarative/interrogative because every verb can appear in any kind of clause: "think" and "wonder" both can appear in question sentences and declarative sentences as follows:

(17) a. John wondered what Mary bought.
b. Did John wonder what Mary bought?

Suppose then that [Dec] and [Q] on V are features which MUST be discharged to lower heads before reaching the CI interface. Otherwise, the verbs retaining these s-selectional features cause “gibberish”. Gibberish here refers to compositional featural interpretability. As in a famous example “colorless green ideas sleep furiously” in Chomsky (1957), each of the words composing the sentence has a certain CI legitimate meaning but once they are composed this way, word meanings are mutually exclusive e.g. in “colorless green”. Here, the same thing can be applied to feature-compositional interpretability. Verbs themselves do not clause-type sentences as exemplified in (17) so that a V feature and a Q/Dec feature are mutually exclusive and cannot stay together when the verb enters semantic computation at the CI interface. If they stay together as is throughout the entire computation, the outputs are regarded as gibberish like “colorless green” are. i.e. a single verb cannot interpreted as a declarative, nor as a question. (See Epstein, Kitahara and Seely 2008 for more detailed discussion of compositional feature interpretability concerning valued phi-features and inherent interpretable tense features on T.) Therefore, s-selectional features have to be discharged (i.e. cannot stay on V) to avoid semantically anomalous interpretations. In order to give formal status to this idea, I propose that the discharge of s-selectional features is executed through feature inheritance from a verb to a lower head as follows:

(18) ~~V~~ C
$\boxed{[\quad\;]}$ [Dec/Q]
Inheritance

The V-to-C feature inheritance system illustrated above ensures s-selectional relations between a verb and the head of its complement CP. So far, I have limited my focus to the relation between V and its complement but the argument also applies to other categories such as N and their complements as in a sentence like (13), in which an N such as “fact” requires a declarative CP complement.

What happens to C after inheritance, illustrated in (18), takes place? After C receives an s-selectional feature from a verb, the embedded CP is clause-typed either as declarative or as interrogative. For example, consider (19):

(19) $[_{CP}$ what Mary bought$]$
(20) a. wondered $[_{CP}$ $]$
b. thinks $[_{CP}$ $]$

Suppose that (19) is the embedded CP which is in the process of the derivation.[3] The object "what" is attracted by EF on C to its edge position. If the CP is merged with "wonder" as in (20a), s-selection [iQ] is inherited from the verb to the embedded C. Then, the embedded CP is clause-typed as an interrogative sentence since its head C received [iQ] from the verb. Since C is marked with [iQ], the wh-phrase "what" is interpreted (at the CI interface) at the edge of the embedded CP. Therefore, "what" stops moving at this derivational point. If the embedded CP is merged with "think" as in (20b), on the other hand, the embedded C receives [iDec] from the verb. This implies that "what" is not interpreted at this position and needs to keep moving to another edge of CP (i.e. if "what" stops, the representation is regarded as gibberish/semantically anomalous at the CI interface). Therefore, V-to-C inheritance has to take place within narrow syntax to give appropriate instructions to the interfaces regarding semantic interpretation of wh-questions and phonetic interpretation i.e. possible pronunciation sites of wh-copies.

In addition to clause-typing, the other thing which occurs after inheritance in (18) is phonetic specification of C. Remember that phonetic realization of C is dependent on a verb selecting it. That is, by receiving features from a verb, phonetic features on C can be (abstractly) specified. (As discussed in Chapter 3, actual morpho-phonetic realization of

[3] In (19), I ignore the vP phase to make the discussion simpler.

C is computed later in the morphological component.) Consider the Japanese cases presented in (15), repeated as (21) below:

(21) *Japanese*

a. John-ga [Mary-ga hon-o kat-ta to/*ka] omot-teiru.
John-Nom Mary-Nom book-Acc buy-Past C think-Prog
"John thinks Mary bought books."

b. John-ga [Mary-ga hon-o kat-ta *to/ka] Sam-ni tazune-ta.
John-Nom Mary-Nom book-Acc buy-Past C Sam-to ask-Past
"John asked Sam whether Mary bought books."

If the verb taking the complement CP is "think", C is phonetically specified as "to". If the verb is "ask", C is phonetically specified as "ka". Depending on which types of s-selectional feature C receives, that is, phonetic features on C are specified differently. Let us see how the derivation proceeds taking (21b) as an example:

(22) $C_{[Phon/---]}$

(23) V(ask) C
$\boxed{[\quad\quad]}$ [Q] →[Phon/**ka**]
Inheritance

C enters into the derivation with its phonetic features unspecified as illustrated in (22).[4] Then, s-selection [Q] is inherited from "ask" to C as in (23). (If not, feature-compositionally gibberish at the CI interface.) As a consequence of this inheritance, the phonetic feature on C is (abstractly) specified as /ka/. The same procedure also takes place for the declarative case in (21a), in which the phonetic feature of C is specified as /to/. Notice that V-to-C inheritance is now not only motivated for compositional feature interpretability issue of V but also for phonetic specification of C. This analysis can be extended to the cases in (11)-(14) in addition to (21). In Cape Verdean Creole (11), if the s-selectional feature [Dec] is inherited from an illocutionary verb such as "tell", the

[4] I abstract away other features on C such as phi-features which are inherited by T and EF for present purposes.

complement C is phonetically specified as /ma/. If [Dec] is discharged from a verb such as "think", the C is phonetically specified as "ki". Also in English cases, if "think" inherits [Dec] to the head of the complement CP, C can be phonetically specified as either Ø or "that".[5] If the verb is "wonder", C is specified as one of the following three options: "if", "whether" or Ø.[6] The last option Ø is chosen only when a moved wh-phrase exists within the CP. Furthermore, if "whisper" as in (12b) and the N "fact" as in (13) bequeath [Dec] to the head of the complement CP, C is phonetically specified as "that". The system proposed here can capture the relations from a verb to its complement CP specifically based on s-selection and phonetic specification of C. The mechanism proposed here can be seen as a generalized theory of feature inheritance in that all the selectional requirements, not only from C to T and from v to T but also from V to C, are explained by feature inheritance.[7] In this sense, Chomsky's (2007, 2008) feature inheritance system gains more empirical ground from the discussion here.

Now, I have an answer to the question why the phase edges of embedded C are left behind after Transfer. The answer is: If embedded CP edges are not left behind after Transfer, unspecified phonetic features on C cause crash at the SM interface. This is why in a convergent derivation, C has to be stranded after Transfer and needs to wait for phonetic specification through s-selectional feature inheritance from a verb selecting it. In addition, successful execution of feature inheritance has a by-product, gibberish-

[5] The issue of which option (Ø or "that") is chosen is an problem independent of the current discussion. See e.g. Pesetsky (1992), Bošković and Lasnik (2003) and Epstein and Seely (2006) for relevant discussion.
[6] Here, I assume that English interrogative C "if" and "whether" are kinds of Q-morpheme, which are similar to "ka" in Japanese. Kayne (1991) suggests that "whether" is, in fact, a wh-phrase, not a complementizer while "if" is a complementizer. Given his analysis, "whether" is at the edge of CP, not at C. Although the thesis assumes that both "if" and "whether" are at C, even if Kayne's analysis is employed, it does not make any different predictions with respect to the proposed system.
[7] Although I did not discuss selectional relations between T and v, I believe that the same kind of inheritance system as V and C is assumed to prevent data like "*I want to kicks Bill."

avoidance. Under this system, Transfer can in principle send the entire phase (i.e. vP and embedded CP) because those derivations never converge. I will discuss this issue more in detail below in Section 5.2.5, in which I reformulate the Transfer operation eliminating the category-specific stipulation that "Transfer applies only to a phase head complement."

5.2.4 Matrix CP: Clause-Typing by Agree

The last section concentrated on embedded phases, and now the focus shifts to matrix phase issues. In Chapter 4, I discussed the problem of how to transfer the matrix edge positions under the current Transfer system. The problem I pointed out is: Since the matrix edges are never included in the domain of any phase head, there is no way to send these positions to the interfaces. To solve this problem, I advanced the working hypothesis concerning Transfer of matrix edges, which is repeated as (24) below:

(24) Working Hypothesis:
Unlike all other phases, the highest phase head and its edge are transferred "for free" along with the phase head complement.

Given that (24) is on the right track, I discussed in Chapter 4 what implications are available from it. The working hypothesis makes it possible that a matrix phase head retains (un)valued [uF] which can participate as the probe in Agree because matrix phase edges are transferred along with the phase head complements. That is, the problems pointed out by Chomsky/Richards (i.e. syntactically valued [uF] in a lower phase is not distinguishable from inherently [iF] when the next Transfer applies, causing failure of valued [uF] removal by Transfer) are no longer problematic here because the entire root phase is transferred under (24). Also, I demonstrated that some of the root transformations are explained. However, there is a serious problem confronting (24). The working hypothesis (while empirically motivated) is a mere stipulation. If we can induce (24) from other conditions, the data accounted for under (24) could be explained. This is the problem I tackle in this section and in the next section i.e. I will explain, without (24), why the matrix phase edge has to be transferred along with the phase head complement as opposed to vP phases and embedded CP phases, in which phase edges have to be stranded in convergent derivations.

Remember in the last section that satisfaction of s-selection and phonetic specification of C are the reasons why embedded CP edges are stranded after Transfer (putting aside the vP phase issue for the time being). If the edge positions of embedded CP are NOT stranded, unspecified phonetic features on C, which are not readable/legible at the SM interface, causes crash and a stranded s-selectional feature on a verb leads to a gibberish output. Considering the matrix edges, there is no s-selection of matrix C by a higher verb/head. On the other hand, it appears that phonetic specification of C is again determined within narrow syntax depending on clause-typing. When it enters into the derivation, that is, phonetic features on matrix C seem to be unspecified just as phonetic features on embedded C do.

(25) a. John buys the book.
     b. Does John buy the book?
(26) a. John and Mary are professors.
     b. Are John and Mary professors?

As illustrated above, in English yes/no question sentences, phonetic realization of C is dependent on phonetic realization of T, which is determined through Agree with the subject. In (25), after T agrees with the subject "John" and receives its phi-features (i.e. 3rd person, singular), finally T is phonetically specified as "does". Then, C can be also phonetically specified. In (26), "be" is inflected as "are" after "be" agrees with the subject "John and Mary". In addition, since the matrix C in the declarative sentences has to be phonetically null in contrast to the question counterparts, clause-typing affects how C is realized. That is, like in embedded C, phonetic features on the matrix C also enter into the derivation unspecified and the derivation in narrow syntax sends instructions concerning phonetic realization of C (and T) to the SM interface. Now, the next question is how the phonetic specification of the matrix C is carried out. In the case of embedded C, s-selectional features inherited from the selecting verb/head play central roles as discussed in the last section. But apparently, the same scenario is not maintainable for the matrix C since no head s-selects the matrix C. I propose the following generalization with respect to phonetic specification of C:

(27) *Phonetic Feature Specification of C*
The phonetic features on C are specified through either
a. feature-inheritance from V OR
b. Agree triggered by C.

The option (27a) is chosen in embedded C, which I discussed in the last section, and (27b) is chosen in matrix C. (I will discuss in the next section the issue of how which of the two options is appropriately selected.)

Let us see how the option (27b) works. The basic scenario is that the matrix C triggers Agree for clause-typing and as a consequence of the clause-typing, phonetic features on C are specified. In the case of wh-questions, C triggers Q-agreement as follows:

(28) What does John buy?

(29) $C_{[uQ][Phon/\_\_]}$ $WH_{[iQ]}$

Q-agreement/clause-typing

[Phon/**does**]

Phonetic feature specification by copying

C bearing unspecified [Phon] and [uQ] agrees with [iQ] on the wh-phrase and [uQ] is valued. As a consequence of Q-agreement, the phonetic feature on C is specified. Notice that English has complementizer morphemes only for embedded C but not for the matrix C as follows:

(30) a. *What whether John buy?
b. *What if John buy?[8]
c. *That John bought the book.

This seems to be a morphologically accidental gap. Since valued [uQ] cannot specify phonetic features on C by itself, the phonetic feature on T is copied i.e. [Phon] on C is specified as "does". Notice that this copy operation is for phonetic specification of C and its output from narrow syntax works as an encoding for head movement as a PF-movement. (See Boeckx and Stjepanovic 2001 for arguments in favor of head movement as a PF-movement.) If a sentence is a subject wh-question, there seems not to be Subj-Aux inversion. Consider the following paradigm:

(31) Who bought the book?
(32) Who can buy the book?

In (31), features on T "hop" onto the verb (i.e. affix hopping, cf. Chomsky 1957) since the do-support operation is a language-specific operation and is regarded as more costly. Therefore, the operation is applied as last resort as discussed in Chomsky (1993). That is, phonetic specification of T in this case is [Phon/null]. This is why C is also specified as null. On the other hand, T bears auxiliaries such as "can" or "will" in some cases. In (32), the same derivation as in (29) takes place: The phonetic features on C are specified as

[8] A sentence like “what if John bought the book?” is grammatical. I assume that there is no wh-movement of “what” in this sentence. Wh-expressions such as “what if”, “how about” and “how come” can be regarded as something like English Q morphemes. Since they seem to behave differently from wh-questions involving wh-movement, however, I put aside these wh-expressions in this thesis. See Collins (1991) for relevant discussion.

"can" by copying them from T. If a language has PF-realized matrix COMP morphemes, those morphemes are specified as a consequence of Q-agreement. For example, Japanese is such a language.

(33) John-ga nani-o kat-ta no?
John-Nom what-Acc buy-Past Q
"What did John buy?"

Interestingly, the Q-morpheme "no" in Japanese is allowed to appear only at the matrix C, but not at embedded C. That is, we can say that "no" can be specified only by direct Q-agreement with C, not by Q-inheritance to T. The same thing can be applied to English Subj-Aux inversion: The [Phon] copy operation from the closest head can be triggered only by Q-agreement but not by Q-inheritance. This is the derivation for matrix wh-questions under the option (27b).

Next let us consider declarative sentences and yes/no-questions under the option ((27b). In these cases, the matrix C has to clause-type and specifies its phonetic features by Agree. However, what element/features can be goals of Agree triggered by C in these cases? There are no wh-phrases unlike in the last case. I suggest that C agrees with TP (i.e. interpreted as proposition at CI) in these cases since semantically yes/no-questions are "for" asking the truth value of a proposition and declaratives are for stating the truth value of a proposition as follows:

(34) *Declaratives:* "He bought the book."
$\mathrm{C}_{\mathrm{[uDec][phone/\_\_]}}$ $\mathrm{TP}_{\mathrm{proposition}}$

[Phon/**null**]
Phon specification

(35) *Yes/No-Questions:* "Does he buy the book?"

C[uQ][Phon/__] TP proposition

[Phon/**does**]

Phon specification by copying

In (34), [uDec] on C triggers agreement with TP/proposition and is valued. In English and many other languages, the matrix C's phonetic features are specified as null in declarative sentences. That is, its phonetic feature is specified as null. Again, the asymmetry between matrix C and embedded C appears to be explained: While matrix C is always null in declaratives, embedded C in embedded declaratives has more options--either "that" or null. That is, we can say that Dec-agreement specifies phonetic features on C as null while Dec-inheritance specifies phonetic features on C as either null or "that". In (35), [uQ] on C agrees with TP/proposition and then is valued. As a consequence of Q-agreement, not Q-inheritance, phonetic features on T are copied in the same way as in matrix wh-questions. Again, if a language has a Q-morpheme for the matrix C (e.g. in Japanese), the morpheme is specified here. Notice that TP/proposition as a goal of agreement triggered by C is last resort. If e.g. [uQ] finds another goal such as [iQ] on a wh-phrase, the agreement between them is carried out. Only when there are no other goals, C agrees with TP/proposition. This is the mechanism of clause-typing and phonetic specification of the matrix C under the Agree option in (27(27b). The following list is a summary of clause-typing options and subsequent phonetic specification of C in English and Japanese:

(36)

	Inheritance		Agree	
	[Dec]	[Q]	[uDec]	[uQ]
ENG	null or "that"	null, "if", "whether"	null	*No morphemes* COPY of T
JPN	"to"	"ka(doka)"	null, "ne", "yo" etc..	"no" (or "ka" for polite form)

As the above table implies, phonetic specification through inheritance and through direct Agree by C gives rise to different outputs. That is, clause-typing by appeal to two different methods can explain some aspects of the differences in morpho-phonetic realization of C between embedded C and matrix C.

This section has discussed the matrix CP phase. The tentative answer to the question why the matrix CP edge is transferred along with C's complement is: The matrix C has to clause-type and specify phonetic features by triggering Agree by itself (because there is no verb selecting it). Therefore, the matrix C always bears [uF] in a convergent derivation and such phase heads cannot be stranded after Transfer under Chomsky/Richards deduction. This is why the matrix edges have to be transferred along with the phase head complement, as opposed to in vPs and in embedded CPs.

The next question to be answered is how one of the two options (i.e. Inheritance or Agree) is selected. That is, how does the computational system choose Inheritance for embedded CP and direct Agree by matrix C for matrix CP? The next section answers this question and also proposes a reformulated definition of Transfer.

5.2.5 Transfer Freely

Now, it is time to remove a stipulation concerning Transfer. Instead of limiting Transfer only to a phase head complement, I propose the following reformulated Transfer:

(37) *Transfer (reformulated)*
Within a single phase, Transfer can send any XP.

The stipulation at issue no longer exists in (37), but rather Transfer, in principle, freely sends any XP to the interfaces. In other words, the entire phases are sometimes transferred. Let us see how the system works:

(38) $[_{\mathbf{YP}}$ X $[_{\mathbf{ZP}}$ Y]]
(Y is a phase head.)

Under the original version of Transfer, ZP which is a complement of Y is uniformly transferred. On the other hand, the reformulated Transfer in (37), YP and ZP are both the possible transferred units. Among these possibilities, the computational system chooses YP, not ZP. Why? If ZP is chosen, it means that Transfer applies to "part" of the representation i.e. leftover is stranded. The leftover will be either (i) included into the next higher phase (if it is an embedded clause) or (ii) transferred by applying additional Transfer (if it is a matrix clause). In the case of (i), the computational workspace in the next phase is extended causing more computational complexity. In the case of (ii), an extra syntactic operation has to be applied. Either of the cases violates the economy condition: fewer or simpler is better. This is why YP is chosen under the reformulated Transfer system. If Transfer of YP causes crash of the derivation but Transfer of ZP does not, however, the situation is different because the most economical option has to be chosen among convergent derivations. In this case, that is, ZP has to be chosen. The current reformulated Transfer allows any XP to be sent to the interfaces but the economy condition requires Transfer to send as much representation as possible, yielding convergence.

One might wonder why only XP is counted as a possible unit of Transfer and why not bar-level projections. Chomsky (1995a) and Chomsky (1995b) suggest that bar-level projections exist but are invisible for computation:

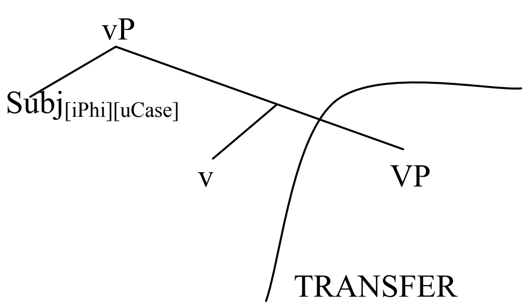
"… Given a phrase marker, a category that does not project any further is a maximal projection XP and one that is not a projection at all is a minimal projection X^0; any other is an X', invisible at the interface and for computation.

(Chomsky 1995b: 396)

Following his suggestion, for the Transfer operation, XP-level projections are visible but bar-level projections are not. That is, bar-level projections cannot be counted as candidates for the unit of Transfer because of their invisibility. With respect to X^0, it can be, in principle, the unit of Transfer. However, the economy condition excludes this option because the representation left behind by X^0-Transfer is included in the workspace of the next higher phase, so that it extends the computational workspace causing more computational complexity. This is why the X^0 is not chosen in a convergent derivation. My reformulated Transfer system is following this assumption.

Let us return to vP phases, embedded CP phases and matrix phases. First, in the case of the vP phase, its edge position always has the subject whose [uCase] is valued by T.

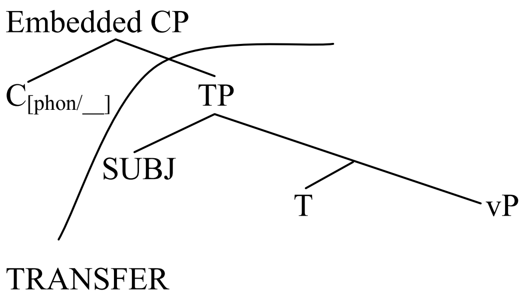
(39)



If the entire vP phase is transferred, the subject bearing unvalued [uCase] causes crash of the derivation. Therefore, in this case, VP is the maximal XP to which Transfer is applied observing the (convergence-seeking) economy condition.

Second, consider the embedded CP phase. As discussed in Section 5.2.3, the embedded C receives an s-selectional feature from a higher verb and subsequently phonetic features on C are specified.

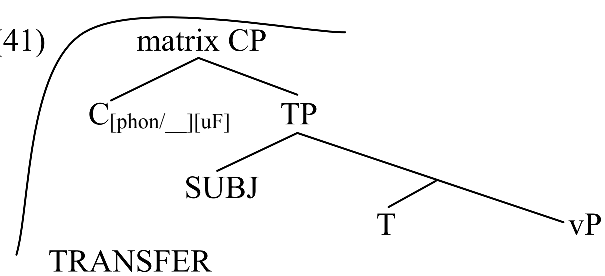
(40)



Before a higher verb enters into the derivation, phonetic features on C are unspecified as in (40). If the entire CP is transferred under the reformulated Transfer system, unspecified [Phon] is not interpretable/legible at the SM interface causing crash. But, if TP is chosen as a transferred unit, C can wait for a higher verb and receive an s-slectional feature from it. Therefore, C is phonetically specified through inheritance and also a higher verb succeeds in discharging its s-selectional feature i.e. no gibberish output is produced. Without stipulating that Transfer is limited to a phase head complement, it can be induced that the phase edge of embedded CP is stranded after Trasnfer in a convergent derivation.

Third, consider the matrix CP phase. As discussed in Section 5.2.4, the matrix C triggers Agree by itself for clause-typing and phonetic specification. Consider the following configuration:

(41)



Since C bears [uF] which is valued for clause-typing, the only convergent way is to transfer the entire CP phase, given the Chomsky/Richards' deduction. Again, the reformulated Transfer system does not specify a certain domain as the transferred unit, unlike the original Transfer operation. Therefore, the most economical convergent option is always chosen. In the case of the matrix CP phase (41), that option is the entire CP phase. Notice that at this point, the working hypothesis of root Transfer I advanced in Chapter 4, which is repeated in (24) in this chapter, is no longer in use. The reformulated Transfer system naturally selects the entire matrix CP as a transferred unit in a convergent derivation. Therefore, there is no need to stipulate the hypothesis (24) only for root edge Transfer.

One might wonder what happens (i) if the inheritance option of C is chosen in the matrix CP and (ii) if the Agree option of C is chosen in the embedded CP. Do we have to stipulate which option appears in which CP? The answer is: The selection of the two options is fully arbitrary. There is no need to block (i) and (ii) with a stipulation because both (i) and (ii) never converge in any case. If (i) happens, unspecified phonetic features on C causes crash because there is no higher verb which can discharge s-selectional features. If (ii) happens, Transfer has to apply to the entire CP phase in (40) under Chomsky/Richards deduction. The transferred CP can converge as the matrix CP does. At the same time, every representation disappears from narrow syntax. (This issue will be discussed more in detail in Section 5.3) What does it predict? Two things seem to happen as a result. One is that a verb is externally merged into the derivation in the next phase but there is no way to discharge s-selectional features causing gibberish outputs (e.g. "*I wonder." Or "*I think."). The second thing is: Since no category is left behind within narrow syntax, it is impossible to construct a relation between a higher verb and the transferred CP by Merge (i.e. no sister relation). That is, the transferred CP and a verb "taking" it can no longer be regarded as elements belonging to a single sentence. Given that the lexical array contains all and only the elements used for a single sentence, this derivation cannot exhaust the lexical array as follows:

(42) LA: {{**C, T**} {**v, I think**,} {~~C, T~~} {~~v, John bought, the, book}~~}
(43) John bought the book.

The entire CP phase in (43) is transferred without using "I", "think", v, T and C in (42). This causes the recoverability condition violation since all and only the elements selected to the lexical array have to undergo interpretation at one of the interfaces, but in this case, the leftovers are stuck in the lexical array and never undergo interpretation at the interfaces. This is why this option (i.e. Agree option in embedded CP) never converges. In conclusion, only when the Agree option is chosen for the matrix CP and only when the Inheritance option is chosen for the embedded CP, can the derivation converge. Accordingly, there is no need to stipulate any mechanism deciding which option (Agree or Inheritance) is chosen but rather they are chosen arbitrarily.[9]

In a nutshell, the analysis proposed in Section 5.2 enables us to throw away a stipulation that Transfer is limited to a phase head complement and explains also some asymmetrical behaviors between embedded CP and matrix CP as exemplified by root transformations. In the next section, we further deepen considerations of Transfer especially focusing on what contents are included in transferred domains.

[9] There is another unselected CP, namely adjunct CP. I believe that the proposed system can also explain adjunct CPs as well but I do not discuss this issue in this thesis.

5.3 What Representation is Left Behind after Transfer?

Section 5.2 mainly discussed what part of the representations is (explicably) sent to the interfaces by Transfer—whether phase edges are left behind as Chomsky proposes or the entire phase is transferred.[10] Then, I explained, by appeal to s-selection and phonetic specification of C, why vP phase edges and embedded CP edges remain while a matrix phase edge is transferred along with the phase head complement. This section further considers the Transfer operation, which plays a central role in Chomsky/Richards deduction. I am specifically concerned with the question of what representation remains in narrow syntax after Transfer has applied--i.e. what is the representational output of transfer-application.

In the following two sections (5.3.1 and 5.3.2), I pursue but ultimately reject two possible answers (and also their entailed problems) to the question of what representation remains in narrow syntax after Transfer applies. In Section 5.3.3, I propose the Label-Copying Transfer system and demonstrate that the proposed system is an optimal way to satisfy all the conditions imposed on syntactic derivations and on representations.

5.3.1 Possibility 1: Everything is Left by Transfer: Full Copy.

The first idea is that all aspects of the NS representation are left behind (fully copied) in narrow syntax, after Transfer applies. In this case, all the representations are (somehow) copied and those copies are sent to the interfaces as illustrated in (44) and (45).

[10] Aspects of this section are based on Obata (2010).

(44) **NS:** **PHON:** [$_{VP}$ bought the book]
[$_{vP}$ John v [$_{VP}$ bought the book]]. **SEM:** [$_{VP}$ bought the book]
[$_{vP}$ John v [$_{VP}$ bought the book]]
Full VP Copy

(45) [$_{CP}$ C [$_{TP}$ John T [$_{vP}$ <John> v [$_{VP}$ bought the book]]]]
Full VP Copy

Under this scenario, as illustrated in (44), narrow syntax retains the complete NS representation, and while identical copies are sent to each interface. Thus, the next higher phase head C is introduced retaining the entire VP contents as in (45).

How is the narrow syntax computation to be limited to phases if the entire representation retains after Transfer? Since VP persists in the post-Transfer narrow syntax representation (e.g. (45)), C (and T) should have access to these domains, which render the computational workspace unlimited. To carry out cyclic computation, an independent condition i.e. the Phase Impenetrability Condition needs to be stipulated.

(46) Phase-Impenetrability Condition (PIC)
In phase α with head H, the domain of H is not accessible to operations outside α, only H and its edge are accessible to such operations.
(Chomsky 2000: 108)

Under PIC, C and T are prohibited from searching VP or its contents in (45), which already underwent Transfer. In other words, if the entire representation remains in narrow syntax after Transfer, PIC is additionally necessary in order to block "counter-cyclic" computation. If the entire representation remains after Transfer as in (44), narrow syntax needs to retain representations which are by hypothesis never used in the subsequent computation. In other words, superfluous representations which are in fact syntactically inactive are stuck in narrow syntax with no explanation of the (hypothesized) empirical fact (or hypothesis) that they are inert once derivation proceeds to a higher phase.[11]

5.3.2 Possibility 2: Nothing is Left: "Tree Pruning"

Another possibility is that Transfer of a phase head complement leaves nothing behind in the NS representation after Transfer, contra (44). That is, Transfer sends the representation itself to the interfaces, not a copy. If this is on the right track, narrow syntax representations partially disappear as a consequence of Transfer as illustrated below:

(47) **NS:** → **PHON:** [$_{VP}$ bought the book]
[$_{vP}$ John v [$_{VP}$ bought the book]]. → **SEM:** [$_{VP}$ bought the book]
[$_{vP}$ John v]
Representations are pruned

(48) [$_{CP}$ C [$_{TP}$ John T [$_{vP}$ <John> v]]]

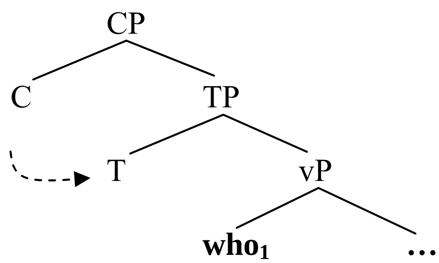
Since, contra (44), VP disappears from narrow syntax as in (48), the higher heads such as C and T have no way to make computational contact with the contents of VP, or with VP itself and this now follows without assuming any independently stipulated principle, i.e. PIC follows, its effects incorporated into the grammatical mechanisms themselves. The reason phase head complements are inaccessible is because they are absent from the NS representations (this arguably being the best way to explain inaccessibility, i.e. absence). Therefore, there is no need to maintain PIC as an independent principle of the grammar to make inter-phasal computation impossible but rather we can have a most natural explanation by saying that syntactic operations or relations can of course involve any

[11] Although it is not clear which view Chomsky adopts, Chomsky (2004) argues based on Nissenbaum (2000) that the distinction between overt and covert movement is made by Internal Merge before or after Transfer. Given this assumption, he implies that at least phonological features are removed from the NS representation as a result of the application of Transfer.

element and only elements existing in the narrow syntactic representation. Furthermore, in (47), unlike the other possibility, narrow syntax does not retain any extra/superfluous representations which are syntactically inert throughout the remainder of the derivation.

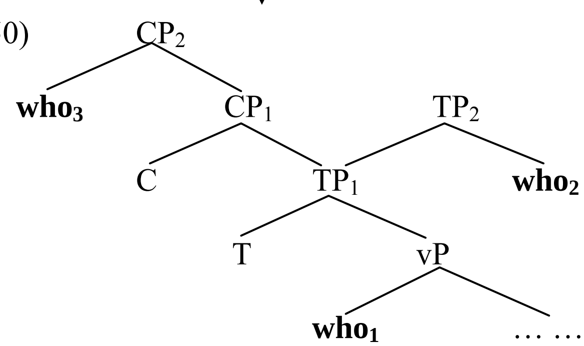
In addition, Epstein, Kitahara and Seely's (2009) analysis implies that this view is supported, (although they do not note the implications of their analysis for the issues under consideration here). According to their argument, Chomsky's (2007, 2008) feature-inheritance operation results ultimately in the creation of a structure with "two peaks"[12] in a derivation for e.g. "I wonder who saw her" as follows:

(49)



Feature-Inheritance from C to T

(50)



Simultaneous Internal Merge of "who" by C and T

Let us focus on the derivation for the embedded CP. Given feature-inheritance, T can work as a probe only after (phi) features are inherited from C as in (49). (See Chapter 2

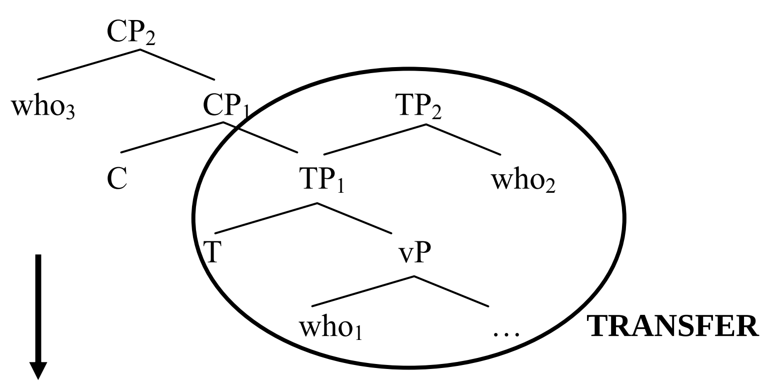
[12] In set theoretic terms, there is set-intersection in (50) i.e. TP1 is a member of >1 set, succinctly, CP1={C, TP_1}, TP2={TP_1, who_2}.

for relevant discussion.) Chomsky (2007, 2008) suggests that in this situation, C and T simultaneously attract the single element "who$_1$" to their edge positions. As illustrated in (50), however, "who$_2$" attracted by T creates a structure with two peaks assuming cyclicity/extension bars "syntactic infixation" of the subject into Edge-TP after C has been merged. Assuming derivational c-command in Epstein et al. (1998) and Epstein (1999), which is defined in (51), neither CP$_2$ nor "who$_3$" c-commands "who$_2$" since there is no derivational relation between them:

(51) *Derivational C-Command:*
X *c-commands* all and only the terms of the category Y with which X was paired/concatenated by Merge or by Move in the course of the derivation.
(Epstein 1999: 329)

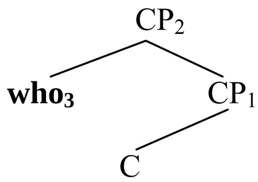
Therefore, it seems to be impossible to decide which projection, CP$_2$ or TP$_2$, is the topmost "root" category necessary for continuing higher derivations (or alternatively they propose a semantic composition failure is indeed by the non-null set interpretation in (50). Epstein, Kitahara and Seely (2009) suggest that by transferring one of the two root projections i.e. TP$_2$ in this case, the "offending" two-peak structure is destroyed and only CP$_2$ survives in narrow syntax as follows, which makes further application of Merge possible.[13] (See Epstein, Kitahara and Seely 2009 for further details.)

(52)



[13] Epstein, Kitahara and Seely (2009) is another analysis to explain why TP, not the entire phase CP, is transferred, although it is not clear to me how two-peak structures undergo reassembly at the interface necessary for global computation such as Condition C.

(53)



If the representation still remains in narrow syntax after Transfer, the two-peak structure still exists and prevents further derivations. That is, their mechanism implies that representations (including one of the two peaks) "entirely disappears" from the narrow syntax by Transfer as illustrated in (53).

For these reasons, it seems to be reasonable to further pursue the possibility that no representation is left behind by Transfer, as in (47). That is, parts of the representation in narrow syntax cyclically "disappear" from narrow syntax. (This would then represent “tree-pruning” as proposed in earlier transformational approaches to deletion.) The next section points out some theoretical problems regarding this possibility and suggests the Transfer Label-Copying system as a general solution.

5.3.3 Label-Copying

If we understand Transfer as a type of "deletion" as discussed in the last section, the elided/transferred part has to be somehow recoverable under the recoverability condition requiring that no information be lost by deletion as discussed most recently in Chomsky and Lasnik (1995). But how is this possible? That is, recoverability seems clearly violated under this eliminative analysis of Transfer (= the nothing-is-left option). Furthermore, there is another problem regarding how the No Tampering Condition (NTC) is satisfied given this approach:

(54) No Tampering Condition
Merge of X and Y leaves the two SOs (= syntactic objects, MO) unchanged.
(Chomsky 2008: 138)

If NTC is extended so that it constrains not only the operation Merge but also the representations derived, a Transfer operation, which obliterates parts of the representation, always breaks up the relation between a phase head and its complement, a relation which is built by Merge. In (47) and (48), the sister relation of v-VP is broken by Transfer by "deleting" VP. Therefore, the current system violates Generalized-NTC in this sense.

How is "Transfer-as-deletion" executable at all in grammars incorporating both Recoverability of Deletion and the Generalized-NTC? To render Transfer-deletion compatible with both of these principles I propose, the following Label Copying Transfer system:

(55) *Label-Copying Transfer*
The transferred phase head complement leaves a copy of only its label when it undergoes Transfer.

Transfer then leaves only the label of the phase head complement behind in the narrow syntax. Given this system, let us see below how the derivation demonstrated in (47) is slightly altered:

(56) **NS:**
[$_{vP}$ John v [$_{VP}$ bought the book]]

PHON: [$_{\mathbf{VP}}$ bought the book]

SEM: [$_{\mathbf{VP}}$ bought the book]

VP-Transfer and Label-Copying
[$_{vP}$ John v [$_{\mathbf{VP}}$]]

(57) [$_{CP}$ C [$_{TP}$ John T [$_{vP}$ <John> v [$_{\mathbf{VP}}$]]]

When VP is transferred in (56), only the copy of its label is left behind in the narrow syntax representation while PHON and SEM each receive the identical copies of the entire VP with its complete internal representation. Notice that PIC is still deduced in this system since the elements within VP do not exist in narrow syntax. Now, let us consider the problems regarding Generalized-NTC and the recoverability condition mentioned earlier. First, how is the recoverability condition satisfied? Since the identical copy at SEM and PHON retains all the information of VP, the identical (but "empty") label VP in narrow syntax can later recover its internal structure at the interfaces as follows:

(58) **NS:** **PHON/SEM**:

Step1: [$_{vP}$ John v [$_{VP}$ bought the book]]

VP-Transfer [$_{\mathbf{VP}}$ bought the book]

Step2: [$_{vP}$ John v [$_{\mathbf{VP}}$]]

Step3: [$_{CP}$ C [$_{TP}$ John T [$_{vP}$ <John> v [$_{\mathbf{VP}}$]]]]

Step 4: *CP-Transfer* [$_{CP}$ C [$_{TP}$ John T [$_{vP}$ <John> v **[$_{\mathbf{VP}}$ bought the book]**]]]]

When the empty label-copy "VP" is transferred as a part of CP (=Step3), the identical copies which were transferred last time are re-inserted into the empty label at the interface. That is, at the interface, by refilling the designated labels, the transferred pieces are re-assembled--as they must be--in order to generate complete sentential (recursive, unbounded) representations at the interface levels. In the above case, the internal structure of VP is re-inserted into the empty VP, which is transferred as a part of TP. Here, the copied labels serve as "guideposts" and the (temporarily) lost information from narrow syntax (i.e. VP-contents) is recovered before reaching the interfaces. What motivates the re-assembly operation? By observing the principle of Full Interpretation, labels lacking internal contents must be filled by its contents, otherwise the representation is uninterpretable. Therefore, the recoverability condition can be satisfied under this approach. The second problem, recall, concerned Generalized-NTC: Transfer as a type of deletion destroys a sister relation created by Merge of a phase head and its complement.

Again, this is no longer problematic for the current system because the copied label is left behind in narrow syntax as in (56), so that the relation of v-VP never changes, and so satisfying Generalized-NTC. Both of the problems are solved under the present approach, while PIC--as an independent stipulation--can be abolished. In this sense, the proposed system based on (55) can be a natural implementation of Transfer observing conditions imposed on syntactic computation. (See Narita 2009 for another approach to the issue of how Transfer affects narrow syntactic representations.)

The derivational procedure illustrated in (58) is the main idea I suggest in this chapter. Reconsidering the two possibilities reviewed in the last two sections (i.e. the everything-is-left approach vs. the nothing-is-left approach), the idea proposed here is a "compromise" located "in between" in the sense that there is a label copy left behind but its contents disappear. With respect to NTC issues, I argued that Transfer breaks sister relations between e.g. v and VP/C and TP violating Generalized-NTC. This is why there is a need to leave a copied label behind. Recall that Section 5.2 tackled the issue of what mechanisms render Transfer of just the phase head complement possible in embedded phases but not in matrix phases. The mechanism can be induced from satisfaction of s-selection/clause-typing by appeal to arbitrary selection of the two derivational options: feature-inheritance or Agree. Under this system, the asymmetries between root phases and embedded phases are explained. Considering these matrix/embedded asymmetries issues in the context of "Label-Copying Transfer", only in vP and embedded CP, copied labels remain in narrow syntax but not in matrix CP. Since the edges of vP and embedded CP are left behind because of s-selection, Transfer sends VP for vP and TP for embedded CP to the interfaces. As mentioned above, Transfer of phase head complements breaks sister relations. To prevent it, the copied label of VP/TP is left behind. On the other hand, matrix CP is not s-selected by any element. This is why the entire CP is allowed to undergo Transfer in this case under the reformulated Transfer system. Since the entire CP is transferred, Transfer does not break any of the sister relations. Therefore, the CP label is NOT left in this case, which observes Generalized-NTC. In other words, matrix CP phases leave neither its edge nor its copied label in narrow syntax leading to no superfluous representations in narrow syntax. The proposed label copy system is compatible with the Transfer system reformulated in Section 5.2.

5.4 On Reassembly: How can Syntactic Objects "Bigger than a Phase" Undergo Internal Merge?

This section further examines the Transfer Label Copy system proposed in Section 5.3. Recall that in the proposed system, Transfer creates a label temporarily in narrow syntax and by re-filling the internal structure with the identical copies (= re-assembly), the representation can come to consist of legible objects at the interfaces satisfying the principle of full interpretation. The cases to be examined here seem to empirically and independently motivate the proposed re-assembly system based on label-copying. Consider e.g.:

(59) a. Mary believed the claim that John bought the book.
b. **Whose claim that John bought the book** did Mary believe _?
(60) a. I think John will buy the book.
b. **John will buy the book**, I think _.
(61) a. It was denied that John bought the book.
b. **That John bought the book** was denied _.

In these cases, the underlined phrases--each bigger than a phase--undergo Internal Merge (IM) which affects both linearization and semantic interpretation. That is, in (59b) it must somehow be the case that the entire DP "whose claim that John bought the book" undergoes successive cyclic movement to the edge of the matrix CP before Transfer applies to it, or any of its sub-terms. But in (59b), for example, at the derivational point at *whose claim [$_{CP}$ that [$_{TP}$ John bought the book]]* is built, the TP is transferred. Now, suppose that there is nothing left behind in narrow syntax after Transfer, not even a label copy. Under that assumption, this TP no longer bears phonological or semantic features, since these have already been transferred, immediately after this TP was merged with C (= "that"). Consequently, by "the time" IM applies to the entire wh-DP underlined in (59b), this TP inside that DP, has already had its phonological and semantic features removed since Transfer already applied to this TP. The resulting output of such wh-DP-fronting at the phonological component is then predicted to be (62): i.e. the TP transported by wh-movement, is incorrectly predicted to lack PHON-features, predicting the incorrect linearization:

(62) The Representation at PHON of (59b)
*__Whose claim that__ did Mary believe **John bought the book**?

Thus, given the standard cyclic Transfer system, the cornerstone of phase-based derivation, it seems impossible to derive any sentences in which constituents bigger than a phase undergo IM, suggesting the possibility of infinite undergeneration.

Regarding this issue, the alternative system proposed in Section 5.3, incorporating label-copying and re-assembly predicts that the wh-DP moved to the edge of the matrix CP contains the empty label TP. After all the representations are transferred from narrow syntax, (63) is obtained:

(63) *PHON&SEM Representations*
[Whose claim that **[$_{TP}$]**] did Mary believe [<whose claim> that
[$_{TP}$ John bought the book]]

Recall, "before" the entire wh-DP underwent IM, TP within this DP was transferred. Based on the Label-Copying Transfer system stated in (55), the empty label is left behind after Transfer, so that the DP at the matrix Spec-CP includes the empty label "TP" whose internal structures are replicated by the identical copies already transferred satisfying the recoverability condition. However, notice that a representation containing an empty node causes violation of the full interpretation principle, (an interface Bare Output Condition), so that (63) is not a legitimate interface representation itself. That is, re-filling/re-assembly is required as a last resort operation as sketched in the last section. The crucial difference between accounts with and without label-copying is that the label-copying system creates an "imperfect"/illegitimate representation such as (63) (presumably crashing, but perhaps gibberish, or both) as the result of the application of Transfer. Therefore, the procedure rendering illegitimate objects legitimate can be invoked as a last resort. In the case of (63), therefore, the empty label TP within the wh-DP occupying the edge of the matrix CP is "re-filled" by copying the identical TP copy of the post-verbal position at the interfaces:

(64) PHON/SEM:
[Whose claim that [$_{TP}$ **John bought the book**]] did Mary believe

COPY

[<whose claim> that [$_{TP}$ **John bought the book**]]

Again, the copied label TP left behind in narrow syntax and the identical copies at PHON/SEM serve as the minimal guideposts necessary for re-assembly at the interface. Of course, re-assembly at the interface is necessary, given phase-based derivation, even independent of the issues raised here. As a result, the representation is regarded as a legitimate object at the interfaces. As mentioned in the previous section, the operations involved in label-copying are not stipulated but rather are induced from independently motivated conditions such as Recoverability, FI and Generalized-NTC. That is, the proposed system is arguably a natural implementation of Transfer satisfying those conditions and it can also explain cases in which phrases bigger than a phase undergo IM. Also, from a cross-linguistic point of view, movement of phrases bigger than phases is not limited to the English cases discussed here but also exists in other I-languages. For example, scrambling in Japanese can also target CP as follows:

(65) a. Taro-ga Jiro-ni [Hanako-ga ringo-o tabe-ta to] it-ta.
Taro-Nom Jiro-to Hanako-Nom apple-Acc eat-Past C tell-Past
"Taro told Jiro that Hanako ate apples."
b. [**Hanako-ga ringo-o tabe-ta to]** Taro-ga **t** Jiro-ni it-ta.
Hanako-Nom apple-Acc eat-Past C Taro-Nom Jiro-to tell-Past
"Taro told Jiro that Hanako ate apples."

In the above example, the entire CP undergoes scrambling to the edge of the matrix CP. That is, this phenomenon also requires some sort of re-assembly procedure of transferred pieces. The system proposed in this section presents one of the possible ways to render phasal reassembly possible in a form observing independently motivated conditions such as (Generalized-)NTC, Recoverability and full interpretation.

Finally, it is noteworthy that the current approach is inconsistent with Collins (2002) analysis suggesting the possible elimination of labels. In the analysis presented in this section, labels play a central role especially in reassembling transferred pieces and also in rendering it possible that Transfer is applied in conformity with Generalized-NTC--without breaking a sister relation.

5.5 Summary

Throughout this chapter, I have specifically examined detailed mechanisms of the

Transfer operation. In the previous chapters (Chapter 3 and 4), I have assumed as is standard, that Transfer sends the phase head complement to the interfaces, which is crucial to maintaining Chomsky/Richards deduction. Then in Chapter 4, I proposed that a matrix CP phase is exceptional in that the edge of the matrix CP is transferred along with the phase head complement and otherwise there is no way to transfer the edge. This exceptional "treatment" of the matrix CP enables us to explain some of the asymmetries between root clauses and embedded clauses such as root transformations, by appeal to independently motivated phase-based derivation coupled with the Chomasky/Richards' analysis which predicts something very special about root edges which have been thus far unexplained.

Based on these analyses made in Chapter 3 and 4, this chapter gave close consideration to the formal nature of the Transfer operation itself by answering the following five questions: (i) Why do edge positions of embedded CP (and vP) have to remain after Transfer?, (ii) Why are edge positions of the matrix CP phase transferred along with the phase head complement?, (iii) How does the computational system find which CP is embedded or matrix?, (iv) What exact contents does Transfer send to the interfaces?/what is the exact representation left in narrow syntax after Transfer applies?, (v) How are transferred pieces reassembled to undergo global computations such as Condition C and sentential intonation? For (i), I focused on s-selection and subsequent phonetic realization of C. To construct a relation between C and a higher V selecting it, s-selectional properties of each verb need to be discharged to its complement CP through inheritance. By means of this inheritance, the verb forces s-selection upon its complement CP leading to clause-typing of the embedded CP. Then, the phonetic realization of C is specified depending on what s-selectional properties are inherited. If the entire embedded CP is transferred, therefore, unspecified phonetic feature on C causes crash of the derivation. With respect to (ii), the matrix C needs to do the same things (i.e. clause-typing and phonetic specification) as embedded C but since there is no s-selecting verb, C triggers Agree by itself in this case. Given Chomsky/Richards deduction, C bearing valued [uF] can never survive after Transfer in a convergent derivation. This is why the edge of the matrix C has to be transferred along with the phase head complement. Otherwise, it causes crash of the derivation. Given this story, the Transfer system can be reformulated as follows: Transfer can send any XP to the interfaces. That is, there is no longer a need to specify which portion is transferred because only when a certain category is chosen, the derivation can converge. Regarding (iii), the distinction between embedded CP and matrix CP can be made freely with the “wrong” choice excluded at the interface (as crash or gibberish). Only when embedded C chooses the inheritance option, the derivation can converge or avoid gibberish outputs. In the same way, only when matrix C selects the Agree option, the derivation can converge. Therefore, the computational system does not need to know and in fact cannot possible know which CP is matrix or embedded. But rather, the difference can be induced from arbitrary selection of either of the two options: Agree or Inheritance. Concerning (iv) and (v), I proposed the Label Copying Transfer system stating that only label copies are left behind in narrow syntax after Transfer, which is induced from the Generalized-No Tampering and Recoverability Conditions. By leaving a copied label, Transfer can be carried out without breaking up a sister relation between e.g. C and TP observing Generalized-NTC. Also, the copied label can ensure that transferred contents/internal structures are not "lost" but rather are "pooled" at the interfaces for reassembly satisfying the recoverability condition. In addition, those copied labels serve as a "guidepost" for reassembly of transferred pieces for global computations. As a case study to demonstrate how reassembly of transferred pieces takes place, I examined movement of phrases bigger than phases, an empirical issue concerning phase-based derivation to my knowledge, not probably raised and addressed.

Chapter 6

Thesis Summary and Architectural Implications

6.1 Introduction

In the previous chapters, I have mainly discussed some aspects of the properties of Internal Merge and the Transfer operation, which both play central roles in Chomsky/Richards deduction. This chapter reflects on the proposals I made in a broader perspective.

Section 6.2 is dedicated to clarifying the assumptions I have made in the previous chapters and discussing what implications are obtained from those assumptions regarding the theory's architecture. Specifically, the system proposed in this thesis is compared with another approach, namely Frampton and Gutmann's (2002) Crash-Proof Syntax. Section 6.3 provides a brief summary of the entire thesis and concludes.

6.2 Architectural Issues: How Strongly/Weakly are Syntactic Operations Constrained?

In this thesis, I have specifically assumed Chomsky's phase-based derivational approach. As Chomsky himself discusses, the "derivational" approach here is the "weak" derivational approach in contrast to the "strong" derivational approach Epstein et al. (1998) and Frampton and Gutmann (2002). Chomsky (2000) says:

"One might construe L [= the language, MO] as a step-by-step procedure for constructing Exps [=Expressions, MO], suggesting that this is how things work as a real property of the brain, not temporally but as part of its structural design. Assumptions of this nature constitute a *derivational approach* to L. The *strong* derivational approach dispenses with the expression altogether, assuming that information is provided to interface systems "dynamically"… The *weak* derivational approach assumes that interface levels exist, allowing "post-cyclic" operations that apply to them in whole or in part (deleting the tail of a chain, imposing metrical structure, determining ellipsis and scope, etc.)…"

(Chomsky 2000: 98)

"I will assume further that L provides information to the performance systems in the form of "levels of representation," in the technical sense. The performance systems access these "interface levels."

(Chomsky 2000: 90)

Under the derivational approach, on which the thesis is based, syntactic representations are constructed by applying operations in a step-by-step manner. In this sense, the derivational procedure is constrained. But how much is it constrained? In the weaker version of the derivational approach, which Chomsky is assuming, there are interface levels, in which interface conditions i.e. Bare OUTPUT Conditions are applied to the resulting representation but not to the derivation itself. This entails that some of the outputs from the syntactic derivation might violate the interface conditions.

"The language faculty must satisfy certain minimal requirements to be usable at all: specifically at least some of the expressions generated by a language must be "legible" to the external systems at the interface."

(Chomsky 1998: 119)

Some of the outputs are legible but others are not. That is, syntax "freely" over-generates these outputs but at the interfaces, they are excluded as “ungrammatical”. This is why, as I demonstrated in the previous chapters, mechanisms under this approach need to be equipped with ways to exclude unacceptable sentences/ungrammatical representations such as improper movement in English by appeal to interface conditions or some aspect of the step-by-step derivational machinery.

On the other hand, the strong derivational approach does not assume any levels of representation i.e. it does not constrain the resulting i.e. output representations at the interface(s). Rather, the derivational procedure itself is highly constrained and the performance/external system directly accesses each “intermediate representation” derived in the syntax. Frampton and Gutmann (2002) characterize their highly constrained syntactic system by comparing it with the opposite system:

"*Free generation and filtering*: Syntactic representations (or derivations) are freely generated. An extensive system of filters assigns status to these representations or to the derivations which produced them."

(Frampton and Gutmann 2002: 91)

"*Highly constrained generation*: Precisely constrained operations iterate to derive a class of representations which are well-formed and interpretable by the interface system. Output filters play no role direct in the generation process."

(Frampton and Gutmann 2002: 91)

The first system is closer to the weak derivational approach, although its syntactic derivation is somehow constrained. The second system is the one they are proposing. Syntax never produces outputs which are NOT interpreted at the interfaces. All of the outputs are well-formed, i.e. satisfy the interface conditions. Since syntactic derivations are constrained fully enough not to generate any ungrammatical outputs, no interface conditions are necessary to pick out ill-formed representation. In this sense, this type of system needs to have all and only the mechanisms to derive "grammatical" outputs. In other words, representations including e.g. improper movement in English do not even exist to begin with, i.e. the narrow syntax fails to generate them.

The weak derivational approach possibly generates many more varieties of outputs from a single lexical array and only one of those possibilities can survive as a legitimate object leading to a grammatical sentence. That is, this system needs to compare all the possible outputs and decides the most economical one. On the other hand, the weak derivational approach, in principle, generates only one grammatical output from a single lexical array. Therefore, this approach does not need to compare possible derivations in contrast to the weak derivational approach. In this sense, the strong derivational approach is less costly in terms of computational complexity than the weak derivational approach is. Also, Brody (2002) argues against the weak derivational approach (which he calls a mixed approach of derivations and representations) in that assuming derivational approaches and representational approaches to be notational variants, an approach employing both approaches has more analytic possibilities (i.e. redundancy) leading to a less restrictive system.[1]

Nevertheless, there still appear to be arguments in favor of the weak derivational system. One of the crucial differences between weak and strong derivational approaches is that one has a mechanism that grammatically characterizes ungrammaticals but the other does not. Epstein (1990, p.c.) suggests that the human faculty of language should be equipped with not only knowledge of grammaticality but also knowledge of types+degrees of ungrammaticality, which makes it possible that we perceive certain sentences as differently ungrammatical. In other words, we know what is ungrammatical as well as what is grammatical.

Chomsky (1965) suggests a relevant idea that a descriptively adequate grammar should not simply "rule out all and only ungrammaticals" but as a model of a cognitive state, a descriptively adequate grammar (an I-language in current terms) should be equipped with some device to differentiate the types/degrees of grammaticality and hence correctly predict natural classes of grammatical deviance. With respect to this issue,

[1] Brody (2002) pushes forward a purely representational approach and rejects a purely derivational approach as well as a mixed approach.

Epstein (1990) shows the following paradigms and claims that those contrasts in degree of (un)grammaticality can be explicated by assuming that the Case Filter is independent of the Theta Criterion (against the spirit of the Visibility Condition discussed in Chomsky (1986a), where the Case Filter is reduced to theta-role assignment).[2]

(1) a. **I hope John to be likely that Bill left.
(*Case Filter, *Theta Criterion)
b. *I want John to be likely that Bill left.
(√Case Filter, *Theta Criterion)

Native speakers of English judge these sentences as deviant. However, the type of ungrammaticality in these two sentences is perceived differently. Loosely speaking, native speakers know that (1a) 'sounds bad at' "John" and in addition the meaning is anomalous. By contrast, (1b) 'sounds fine at' "John" but like (1a) is semantically anomalous. Also, the degree of ungrammaticality is different: (1a) is worse than (1b). If a grammatical theory can account for such empirical phenomena, that theory is preferable to one which cannot. Epstein (1990) working within the GB theory captures this contrast by appealing to how many violations and what kind of filters are involved in excluding these sentences. (1a) violates both the Case Filter and the Theta Criterion. On the other hand, (1b) violates only the Theta Criterion. This difference corresponds to the fact that (1a) is worse than (1b). If the Visibility Condition is adopted reducing the Case Theory to Theta Theory, (1a) violates only the Theta Criterion. That is, each sentence is characterized as a single violation of the exact same filter. The incorrect prediction is that there is no difference in ungrammaticality. Epstein suggests then that these data imply that the Case Filter is not reducible to the Theta Criterion, and so this non-unified theory is empirically preferable.

[2] Epstein (1990) also discusses a third type which violates the Case Filter, but not the Theta Criterion:
(i) *I hope John to think that Bill left.

As long as Epstein's (1990) observation is on the right track, a language should be equipped with mechanisms not only to produce grammaticals but also to characterize ungrammaticals. In this sense, the weak derivational approach is preferred to the strong derivational approach which has only mechanisms to derive grammaticals, although more work is required to fully explain the degree of ungrammaticality even within the weak derivational approach.

Obviously, it is not easy to decide which approach is better or preferable. Also, there are various arguments for and against a derivational approach itself. The attempts made in this thesis pursue only one of many possible approaches.

6.3 Thesis Summary and Concluding Remarks

As clarified in Chapter 1, one of the main goals of this thesis is to present arguments in favor of Chomsky's phase-based derivational approach, specifically Chomsky's (2007, 2008) feature inheritance system.

In Chapter 2, I first discussed historical overviews of generative grammar, especially focusing on how theories had developed from pre-minimalism to current minimalism. By asking *why*-questions, theories are naturally getting minimized and closer to higher levels of explanation. The shift from the P&P approach to minimalism manifests just such an attempt at higher generality. Also, the feature inheritance system discussed in Chomsky (2007, 2008) was summarized mentioning Richards' (2007) deduction of this system from the Value-Transfer simultaneity mechanism. Once an unvalued [uF] is introduced on a phase head, it has to be inherited to a lower head which is included in a transferred domain. Otherwise, syntactically valued [uF] is not distinguishable from [iF] which inherently bears its value in the next higher phase, so that Transfer (of a phase head complement), which cannot detect interpretablity of features, fails to remove/delete valued [uF] from the CI representation causing crash. Therefore, in a convergent derivation, unvalued [uF] has to be inherited to a lower head which is transferred as part of the minimal phase in which the head externally-merged.

Chapter 3 further pushed forward Chomsky/Richards deduction and extended Richards' (2007) observation to phase edge positions. Richards (2007) limits his focus only to phase heads (probes) but I pointed out the same issue can be raised also for elements (goals) moved to phase edge positions. That is, the mechanism of Internal Merge was a matter of concern here. DP moved to phase edge positions bear [iPhi] and valued [uCase]. In standard analyses, when Internal Merge applies to such DPs, valued [uCase] is moved out of the transferred domain to a phase edge, which is stranded after Transfer. If DP undergoes Internal Merge retaining its valued [uCase], the derivation never converges under Chomsky/Richards' deduction. In the case of phase heads, feature inheritance was a possible solution, as Richards (2007) discusses. In the case of an internal-merged DP, however, the same mechanism cannot be applied to get rid of valued [uCase] on the DP occupying the phase edge. Based on Obata (2008) and Obata and Epstein (2008, in press), I proposed Feature-Splitting Internal Merge, in which valued [uCase] is split off from the copy that is moved to the phase edge. Under this mechanism, [iPhi] and [uCase] which are involved in Agree with T/V are attracted by EF on those heads and the rest of the goal's features, such as an operator feature [iOp] is attracted by EF on C/v. As a result, an element moved to a phase edge bears only [iOp] (neither phi-features nor Case-features) which does not cause crash at the interface.

This proposed system may also explain improper movement phenomena by saying that an element at a phase edge (Edge-CP in improper movement) does not have [iPhi] and the element therefore cannot serve as the goal of phi-agreement probing by another head, namely a higher T. Improper movement is then excluded because unvalued [uPhi] on higher T is never valued causing (local) featural crash. (cf. Condition C and Chain Uniformity approaches to improper movement, appealing to global relations.) In addition, by parameterizing how features split, the proposed system can also explain that some languages in fact seem to allow improper movement, e.g. in Kilega/Lusaamia and Japanese. The parameterization is (naturally) based on morphological properties of functional heads (cf. Borer 1984 and Fukui 1986): In contrast to English, Case-valuation and phi-valuation do not co-occur in Kilega/Lusaamia. That is, there are cases in which T phi-agrees with one DP and Case-agrees with another DP. Because of these (learnable) morphological properties, in Kilega/Lusamia, [uCase] and [iPhi] are separable by Feature-Splitting Internal Merge. This makes it possible that an element moved to a phase edge retains [iPhi] in these languages, which is then available for phi-agreement by a higher T, unlike in English. That is, improper movement is then allowed in these languages. In Japanese, on the other hand, phi-agreement is not overtly realized/is “weak”. Whatever kinds of DP T agrees with, a single Case-morpheme "-ga" is simply assigned. In Japanese, that is, phi-features and Case-features are NOT equally involved in agreement by T/V, but rather the involvement of phi-features is much weaker/less than the involvement of Case-features. This is why T/V can attract only [uCase] because of its direct involvement in Agree. As a result, [iPhi] can move to a phase edge along with an operator feature predicting that improper movement is permissible, i.e. a higher T can

Agree with phi-features moved to edge-CP. The proposed system implies that morphological properties (not universals like Condition C, Chain Condition) determine whether the language permits improper movement or not. I also demonstrated that even in English, improper movement can be grammatical in a certain context, namely *tough*-constructions. By focusing on the fact that *tough*-constructions show finiteness sensitivity, types of T/V makes more varieties of feature-splitting possible, namely in a certain context, a moving element in a *tough*-construction can retain and carry [iPhi] to the edge which can then serve as a goal of further phi-agreement by a higher T probe. Chapter 3 is dedicated to explaining the issue of how improper movement is allowed in some cases but not in other cases by appeal to parameterized Feature-Splitting Internal Merge assuming that Chomsky/Richards deduction is right. Furthermore, theoretical and empirical consequences obtained from the proposed analysis were also demonstrated including possible elimination of A/A'-position types, in favor of phi-agreement vs. phi-lacking category-types, facilitated by Feature-Splitting.

Section 4 discusses how root CP phases are transferred. Since Chomsky (2000), Transfer/Spell-Out sends only a phase head complement to the interfaces and a phase edge is transferred as a part of the next higher phase head complement. However, it is not clear how a root CP edge is ever sent to the interfaces given that a root edge is not included in the domain of any phase. In this chapter, I advanced a working hypothesis stating that unlike all other phases, the highest phase head and its edge are transferred "for free" along with their domain. This mechanism is subsumed by other for more general conditions later in Chapter 5, and the empirical and theoretical implications of the analysis are explored. In terms of Chomsky/Richards' deduction, if a root edge is transferred along with its complement, the prediction is that only the root phase head can retain a syntactically valued [uF]. That is, what is implied is that only the root C head can trigger Agree, but not embedded C heads. This prediction is the one suitable to explaining the idiosyncracies specific to root transformations such as T-to-C movement. Under this hypothesis, root transformations can be rephrased by syntactic phenomena triggered only by direct C-Agree, which is the only option available for the root head. I also demonstrated that root-landing movement (i.e. rightward movement) in Japanese, which is another root transformation, can be also explained under the advanced hypothesis.

Chapter 5 gave close consideration to the formal mechanism of the Transfer operation, especially answering the following five questions: (i) Why precisely is it that in all derivations edge positions of embedded CP (and vP) have to remain after Transfer?, (ii) Why is the edge position of the matrix CP phase transferred along with its complement?, (iii) How does the computational system determine which CP is embedded or matrix?, (iv) What exact contents does Transfer send to the interfaces?/what is the exact representation left behind in the narrow syntax after Transfer applies?, (v) How are transferred syntactic objects reassembled to yield complete sentential representations which undergo global computations such as Condition C and sentential intonation?

With respect to (i), I specifically mentioned relationships between a CP complement and a higher selecting V in terms of s-selection and subsequent phonetic realization of C. To ensure the relations between C and a higher V selecting it, s-selectional properties of a verb need to be discharged to its complement CP. I argued that s-selection is implemented through the mechanism of feature inheritance. That is, I proposed a generalized theory of feature-inheritance system. The feature inheritance in C-to-T and v-to-V stems from selectional properties, e.g. C selects a finite TP or a control TP but not a raising TP, and v selects a verb which has inherent external argument theta roles such as accusative verbs and ergative verbs. The inheritance proposed in this chapter also recognizes another selectional relation between a verb and its sentential complement. Depending on what verb selects a CP, the phonetic realization of C differs. If the entire embedded CP is transferred, therefore, unspecified phonetic feature on C causes crash of the derivation. This is why embedded CP edges have to remain after Transfer, otherwise the derivation never converges.

Concerning (ii), the matrix C also needs to specify its phonetic realization just as embedded C does. Since there is no s-selecting verb, however, feature-inheritance never happens here. Instead, C triggers Agree by itself, which is allowed only for the root C, given Chomsky/Richards deduction. C bearing valued [uF] can never survive after Transfer in a convergent derivation. This is why the edge of the matrix C has to be transferred along with its complement, as mentioned in Chapter 5. Under this assumption, I proposed the reformulated Transfer system as follows: Transfer can send any XP to the interfaces. That is, the current system no longer needs a category-specific stipulation to specify which syntactic object is transferred (i.e. the phase head complement) because only when “the correct” certain category is chosen, will the derivation converge.

For (iii), the distinction between embedded CP and matrix CP can be made by appeal to arbitrary selection of ways of clause-typing/phonetic specifications of C. Only when embedded C chooses the inheritance option, the derivation can converge or avoid a gibberish output. In the same way, only when matrix C selects the Agree option, the derivation can converge. Therefore, the computational system does not need to know which CP is matrix or embedded. But rather, the difference can be induced from arbitrary selection of either of the two options: Direct C-Agree or V-to-C feature inheritance.

Regarding (iv) and (v), finally, I proposed the Label-Copying Transfer system stating that only label copies are left behind in narrow syntax after Transfer while complete internal structures are sent to the interfaces. This is induced from a generalized No Tampering Condition and Recoverability Condition. Under this system, Transfer no longer breaks up a sister relation between e.g. C and TP , and thereby observes a generalized NTC. Also, the copied label left behind in narrow syntax can ensure that transferred contents/internal structures are not "lost" but rather are "pooled" at the interfaces for reassembly satisfying the recoverability condition. At the same time, internal structures of copied labels are lost from narrow syntax, which explains the (stipulated) phase impenetrability condition. Furthermore, I also demonstrated that copied labels serve as an independently necessary "guidepost" for reassembly of transferred pieces required to generate complete sentential interface representations which must undergo global computations, e.g. in the case of movement of phrases bigger than phases, a thus far unaddressed empirical issue directly bearing on phase-based computational theory.

Finally, the present chapter overviewed the analyses presented and also discussed the architectural issues especially considering the differences between strong and weak derivational approaches. Although future research awaits to decide which approach is preferable, this thesis is one attempt to reveal some attractive aspects of the weak derivational approach.

Bibliography

Abe, J. 1999. On Directionality of Movement: A Case of Japanese Right Dislocation. Ms., Nagoya University.

Baker, M. 1988. *Incorporation: A Theory of Grammatical Function Changing.* Chicago: University of Chicago Press.

Baker, M. and C. Collins. 2006. Linkers and the Internal Structure of vP. *Natural Language and Linguistic Theory* 24: 307-354.

Bejar, S. and D. Massam. 1999. Multiple Case Checking. *Syntax* 2: 65–79.

Boeckx, C. 2006. *Linguistic Minimalism: Origins, Concepts, Methods and Aims*. New York: Oxford University Press.

Boeckx, C. and S. Stjepanovic. 2001. Head-ing Toward PF. *Linguistic Inquiry* 32: 345-355.

Borer, H. 1984. *Parametric Syntax*. Dordrecht: Foris.

Bośković, Ž. 1997. *The Syntax of Nonfinite Complementation: An Economy Approach*. Cambridge: MIT Press.

Bośković, Ž. 2007. On the Locality and Motivation of Move and Agree: An Even More Minimal Theory. *Linguistic Inquiry* 38: 589-644

Brody, M. 1993. Theta Theory and Arguments**.** *Linguistic Inquiry* 24: 1-23

Brody, M. 2002. On the Status of Derivations and Representations. In *Derivation and Explanation in the Minimalist Program,* ed. by S. D. Epstein and T. D. Seely, 19-41. Cambridge: Blackwell.

Browning, B. 1987. *On Null Operator Constructions.* Doctoral dissertation, MIT.

Carstens, V. 2005. Agree and EPP in Bantu. *Natural Language and Linguistic Theory* 23: 219-279.

Carstens, V. 2008. Raising in Bantu. Ms., University of Missouri.

Chomsky, N. 1957. *Syntactic Structures*. The Hague: Mouton.

Chomsky, N. 1964. *Aspects of the Theory of Syntax*. Cambridge: MIT Press.

Chomsky, N. 1965. *Cartesian Linguistics*. New York: Harper and Row.

Chomsky, N. 1973. Conditions on Transformations. In *A Festschrift for Morris Halle*, ed. by S. Anderson and P. Kiparsky, 232-286, New York: Holt, Reinhart and Winston.

Chomsky, N. 1980. *Rules and Representations*. New York: Columbia University Press.

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

Chomsky, N. 1982. *Some Concepts and Consequences of the Theory of Government and Binding*. Cambridge: MIT Press.

Chomsky, N. 1986a. *Knowledge of Language*. New York: Praeger.

Chomsky, N. 1986b. *Barriers*. Cambridge: MIT Press.

Chomsky, N. 1993. A Minimalist Program for Linguistic Theory. In *The View From Building 20*, ed. by S. Keyser and M. Hale, 1-52. Cambridge: MIT Press.

Chomsky, N. 1995a. *The Minimalist Program*. Cambridge: MIT Press.

Chomsky, N. 1995b. Bare Phrase Structure. In *Government and Binding Theory and the Minimalist Program*, ed. by G. Webelhuth, 383–439. Cambridge: Blackwell.

Chomsky, N. 1998. Some Observations on Economy in Generative Grammar. In *Is the Best Good Enough?* ed. by P. Barbosa, D. Fox, P. Hagstrom, M. McGinnis and D. Pesetsky, 115–127. Cambridge: MIT Press.

Chomsky, N. 2000. Minimalist Inquiries: The Framework. In *Step by Step: Essays on Minimalist Syntax in Honor of Howard Lasnik*, ed. by R. Martin, D. Michaels and J. Uriagereka, 89-155. Cambridge: MIT Press.

Chomsky, N. 2001. Derivation by Phase. In *Ken Hale: A Life in Language*, ed. by M. Kenstowicz, 1-52. Cambridge: MIT Press.

Chomsky, N. 2004. Beyond Explanatory Adequacy. In *Structures and Beyond: The Cartography of Syntactic Structures*, ed. by A. Belletti, Volume 3, 104-131, Oxford: Oxford University Press.

Chomsky, N. 2005. Three Factors in Language Design. *Linguistic Inquiry* 35: 1–22.

Chomsky, N. 2007. Approaching UG from Below. In *Interfaces + Recursion = Language?*, ed. by U. Sauerland and H-M. Gärtner, 1-29. New York: Mouton de Gruyter.

Chomsky, N. 2008. On Phases. In *Foundational Issues in Linguistic Theory: Essays in Honor of Jean-Roger Vergnaud*, ed. by R. Freidin, C. P. Otero, and M. L. Zubizarreta, 133-166. Cambridge: MIT Press.

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

Chomsky, N. and H. Lasnik. 1995. The Theory of Principles and Parameters, In *The Minimalist Program*, 13-127. Cambridge: MIT Press.

Collins, C. 1991. Why and How Come. *MIT Working Papers in Linguistics* 15: 31-45.

Collins, C. 2002. Eliminating Labels. In *Derivation and Explanation in the Minimalist Program,* ed. by S. D. Epstein and T. D. Seely, 42-64. Cambridge: Blackwell.

Dirac, P.A.M. 1968. Methods in Theoretical Physics. In *From a Life of Physics; Evening lectures at the International Center for Theoretical Physics*, Trieste, Italy. A special supplement of the International Atomic Energy Agency Bulletin, Austria. [Reprinted in *Unification of Fundamental Forces*, ed. by A. Salam, 125-143. Cambridge: Cambridge University Press.]

Den Dikken, M. 2001. 'Pluringulars', Pronouns and Quirky Agreement. *The Linguistic Review* 18:19-41.

Emonds, J. 1976. *A Transformational Approach to English Syntax.* New York: Academic Press.

Epstein, S. D. 1984. Quantififer-pro and the LF Representation of PROarb. *Linguistic Inquiry* 15: 499-505.

Epstein, S. D. 1989. On the Derivation and Representation of English 'Tough' Constructions, Talk given at University of California - Los Angeles.

Epstein, S. D. 1990. Differentiation and Reduction in Syntactic Theory: A Case Study. *Natural Language and Linguistic Theory* 8: 313-323

Epstein, S. 1999. Un-Principled Syntax and the Derivation of Syntactic Relations. In *Working Minimalism*, ed. by S. Epstein, and N. Hornstein, 317-345. Cambridge: MIT Press.

Epstein, S. D., E. Groat, R. Kawashima and H. Kitahara. 1998. *A Derivational Approach to Syntactic Relations*. Oxford: Oxford University Press.

Epstein, S. D. and T. D. Seely, 2002. Rule Applications as Cycles in a Level-Free Syntax. In *Derivation and Explanation in the Minimalist Program*. ed. by S. D. Epstein and T. D. Seely, 65-89. Cambridge: Blackwell.

Epstein, S. D. and T. D. Seely, 2006. *Derivations in Minimalism*. Cambridge: Cambridge University Press.

Epstein, S. D., T. Seely, and A. Pires. 2005. EPP in T: More Controversial Subjects. *Syntax* 8: 165-80.

Epstein, S. D., H. Kitahara and T. D. Seely. 2008. The Value of Phonological Underspecification in the Narrow Syntax. Paper presented at the Exploring Crash Proof Grammars Conference. Carson-Newman College, Dandridge, TN. [Written version will be published in *Exploring Crash-Proof Grammars*, ed. by M.Putnam. *Language Faculty and Beyond Series*, ed. by K. Grohmann and P. Pica. Amsterdam: John Benjamins Publishing Company.

Epstein, S. D., H. Kitahara and T. D. Seely. 2009. The Necessity, but Invisibility of Counter-cyclic Outputs: Deducing Extraction Constraints and Transfer-Application from 3rd Factor Conditions on Language Design, *Glow Newsletter*: 34-36.

Frampton, J. and S. Gutmann. 2002. Crash-Proof Syntax. In *Derivation and Explanation in the Minimalist Program,* ed. by S.D. Epstein and T. D. Seely, 90–105. Oxford: Blackwell.

Freidin, R. and H. Lasnik. 1981. Disjoint and Reference and WH Trace. *Linguistic Inquiry* 12: 39-53.

Freidin, R. and J-R. Vergnaud. 2001. Exquisite Connections: Some Remarks on the Evolution of Linguistic Theory. *Lingua* 111: 639-666.

Fukui, N. 1986. *A Theory of Category Projection and Its Applications*, Doctoral dissertation, MIT.

Fukui, N. 1993. A Note on Improper Movement. *The Linguistic Review* 10: 111-126.

Grewendorf, G. 2001. Multiple Wh-Fronting. *Linguistic Inquiry* 32: 87-122.

Grimshaw, J. 1979. Complement Selection and the Lexicon. *Linguistic Inquiry* 10: 279-326.

Haraguchi, S. 1973. Remarks on Right Dislocation in Japanese. Ms., MIT.

Hicks, G. 2009. *Tough*-Constructions and their Derivation. *Linguistic Inquiry* 40: 535-566.

Hiraiwa, K. 2002. Multiple Agree. Paper presented at *The 25th GLOW Colloquium: Tools in Linguistic Theory (TiLT),* Utrecht.

Hiraiwa, K. 2010. Scrambling to the Edge. *Syntax* (online, early view).

Hornstein, N. 1999. Movement and Control. *Linguistic Inquiry* 30: 69-96.

Hornstein, N. 2001. *Move! A Minimalist Theory of Construal*. Oxford: Blackwell.

Horvath, Julia 1981. *Aspects of Hungarian Syntax and the Theory of Grammar*. Doctoral dissertation, University of California, Los Angeles.

Inoue, K. 1978. *Nihongo-no Bunpoo-kisoku [Grammatical Rules of Japanese]*. Tokyo: Taishuukan.

Kato, T. 2007. On the Nature of LBC: Syntactic or Phonological? In *Proceedings of Seoul International Conference of Generative Grammar* 9: 39-51.

Kayne, R. 1991. Romance Clitics, Verb Movement and PRO. *Linguistic Inquiry* 22: 647-686.

Kinyalolo, K. K. W. 1991. Syntactic Dependencies and the Spec-Head Agreement Hypothesis in Kilega. Doctoral dissertation, University of California, Los Angeles.

Kiss, K. 1985. Parasitic Gaps and Case. Paper presented at MIT.

Kitahara, H. 2006. Some Notes on the Minimalist Program. In *Minimalist Essays*, ed. by C. Boeckx, 3-15. Amsterdam: John Benjamins.

Koopman, H. and D. Sportiche. 1983. Variables and the Bijection Principle. *The Linguistic Review* 2:139-160.

Kuno, S. 1976. Subject, Theme and Speaker's Empathy: A Reexamination of Relativization Phenomena. In *Subject and topic*, ed. by C. Li, 417-444. New York: Academic Press.

Kuno, S. 1978. *Danwa-no Bunpoo [Grammar of Discourse]*. Tokyo: Taishuukan.

Lasnik, H. 2001. When Can You Save a Structure by Destroying It? *Proceedings of the North East Linguistic Society* 31, ed. by Min-Joo Kim and Uri Strauss, 310-320. Amherst: GLSA.

Lasnik, H. and R. Fiengo. 1974. Complement Object Deletion. *Linguistic Inquiry* 5: 535-571.

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

Lasnik, H. and M. Saito. 1992. *Move α.* Cambridge: MIT Press.

Lasnik, H. and T. Stowell. 1991. Weakest Crossover. *Linguistic Inquiry* 22: 687-720.

Lasnik, H. and J. Uriagereka. 2005. *A Course in Minimalist Syntax.* Oxford: Blackwell.

Legate, J. A. 2008. Morphological and Abstract Case. *Linguistic Inquiry* 39: 55-101.

Masuoka, T. 1991. *Modality-no Bunpo [Grammar of Modality].* Tokyo: Kuroshio Shuppan.

Matushansky, O. 2006. Head-Movement in Linguistic Theory. *Linguistic Inquiry* 37: 69-109.

May, R. 1979. Must COMP-to-COMP Movement Be Stipulated? *Linguistic Inquiry* 10: 719-725.

Miyagawa, S. 1997. Against Optional Scrambling. *Linguistic Inquiry* 28: 1-25.

Miyagawa, S. 2010. *Why Agree? Why Move? Unifying Agreement-based and Discourse Configurational Languages*. Cambridge: MIT Press.

Merchant, J. 2001. *The syntax of silence: Sluicing, islands, and the theory of ellipsis*. Oxford: Oxford University Press.

Narita, H. 2009. Multiple Transfer in Service of Recursive Merge. Paper presented at *The 32nd GLOW Colloquium*, University of Nantes, Nantes, France.

Nevins, A. 2005. Derivations without the Activity Condition. *MIT Working Papers in Linguistics* 49: 283-306.

Nissenbaum, J. 2000. Investigations of Covert Phrase Movement. Ph.D. thesis, MIT.

Obata, M. 2008a. Improper Movement, Intervention and the (Possible) Elimination of A/A'-Position Types. Ms., University of Michigan, Ann Arbor.

Obata, M. 2008b. On Simultaneous Attraction: How are Phonological Features Split? Ms., University of Michigan, Ann Arbor.

Obata, M. 2008c. On the Formal Nature of Attraction and the Suppression of Superiority Effects. In the *Proceedings of the 37th North East Linguistics Society*, Volume 2: 115-126.

Obata, M. 2009. Determining Phonetic Feature Distribution: How to "Pronounce" Three Types of A'-Movement. In the *Proceedings of the 2nd International Conference on East Asian Linguistics*. Simon Fraser University.

Obata, M. 2010. How to Move Syntactic Objects Bigger than a Phase: On the Formal Nature of Transfer and Phasal RE-Assembly. In the *Proceedings of the 27th English Linguistic Society of Japan (JELS).*

Obata, M. and M. Baptista, 2009. Complementizer-Alternation in Cape Verdean Creole: New Evidence for Spec-Head Agreement. Paper presented *at the 83rd Annual Meeting of the Linguistic Society of America.* San Francisco, CA.

Obata, M. and S. D. Epstein. 2008. Deducing Improper Movement from Phase Based C-to-T Phi Transfer: Feature-Splitting Internal Merge. In the *Proceedings of the 27th West Coast Conference on Formal Linguistics*: 353-360. Somerville: Cascadilla Proceedings Project.

Obata, M. and S. D. Epstein. In press. Feature-Splitting Internal Merge: Improper Movement, Intervention and the A/A'-Distinction. *Syntax.* Cambridge: Blackwell Publishing Company.

Obata, M. and S. D. Epstein. To appear. Feature-Splitting Internal Merge: The Case of *Tough*-Constructions. In *Ways of Structure Building*, ed. by V. Valmala and M. Uribe-Etxebarria.

Obata, M. and S. D. Epstein, In prep. "For-To" Revisited. Ms. University of Michigan, Ann Arbor.

O'Neil, J. 1995. Out of Control. In *The Proceedings of the 25 North East Linguistic Society*, 361-171. Amherst: GLSA.

Ouali, H. 2006. *Unifying Agreement Relations: A Minimalist Analysis of Berber*. Doctoral dissertation, University of Michigan, Ann Arbor.

Pesetsky, D. 1982. *Paths and Categories*. Doctoral dissertation, MIT.

Reinhart, T. 1976. *The Syntactic Domain of Anaphora*. Doctoral dissertation, MIT.

Rezac, M. 2003. The Fine Structure of Cyclic Agree. *Syntax* 6: 156-182.

Rezac, M. 2006. On *Tough*-Movement. In *Minimalist Essays*, ed. by C. Boeckx, 288-325. Amsterdam: John Benjamins.

Richards, M. 2004. Object Shift, Scrambling, and Symmetrical Syntax. Doctoral dissertation, University of Cambridge.

Richards, M. 2007. On Feature Inheritance: An Argument from the Phase Impenetrability Condition. *Linguistic Inquiry* 38: 563-572.

Rizzi, L. 1986**.** Null Objects in Italian and the Theory of pro. *Linguistic Inquiry* 17: 501-555.

Ross, J. 1967. *Constraints on Variables in Syntax*. Doctoral dissertation, MIT.

Saito, M. 1983. Comments on the Papers on Generative Syntax. In *Studies in Generative Grammar and Language Acquisition: A Report on Recent Trends in Linguistics*, ed. by Y. Otsu, H. V. Riemsdijk, K. Inoue, A. Kamio, and N. Kawasaki, 79−89. Tokyo: International Christian University.

Saito, M. 1985. *Some Asymmetries in Japanese and their Theoretical Implications*. Doctoral dissertation, MIT.

Saito, M. 2004. Japanese Scrambling in a Comparative Perspective. In *Peripheries: Syntactic Edges and their Effects*, ed. by David Adger, Cécile de Cat, and George Tsoulas, 143-165. Netherlands: Kluwer Academic Publishers.

Saito, M. 2007. Notes on East Asian Argument Ellipsis, *Language Research* 43: 203-227.

Simon, E. 1989. An Analysis of Postposing Construction in Japanese. Doctoral dissertation, University of Michigan, Ann Arbor.

Svenonius, P. 2000. Quantifier Movement in Icelandic. In *The Derivation of VO and OV,* ed. by P. Svenonius, 255-292. Amsterdam: John Benjamins.

Takano, Y. 2005. Postposing in Japanese. Paper presented at *the Linguistic Society of America Summer Institute*, MIT.

Takezawa 2006. Case, Agreement, and Movement to Subject Position: A View from Japanese, Ms., University of Tsukuba.

Takita, K. to appear. Argument Ellipsis in Japanese Right Dislocation. *Japanese/Korean Linguistics* 18, Stanford: CSLI Publications.

Tanaka, H. 2001. Right-Dislocation as Scrambling. *Journal of Linguistics* 37: 551-579.

Tanaka, H. 2002. Raising to Object out of CP. *Linguistic Inquiry* 33: 637-652.

Toyoshima, T. 2000. Head-To-Spec Movement and Dynamic Economy. Ph.D. thesis, Cornell University.

Uchibori, A. 2000. *The Syntax of Subjuncive Complement: Evidence from Japanese*. Doctoral dissertation, University of Connecticut.

Ura, H. 1994. Varieties of Raising and Feature-Based Bare Phrase Structure Theory. *MIT Occasional Papers in Linguistics* 7.

Vergnaud, J-R. 1977. Letter to Noam Chomsky and Howard Lasnik. [Published in *Foundational Issues in Linguistic Theory,* ed. by R. Freidin, C. P. Otero and M. Vernaud, 3-15, Cambridge: MIT Press, 2008]

Weinberg, S. 1976. The Force of Nature. *Bulletin of the American Academy of Arts and Sciences* 29: 13-29.

Weinberg, A. 2001. A Minimalist Theory of Human Sentence Processing. In *Working Minimalism*, eds. S. D. Epstein and N. Hornstein, 283-315. Cambridge: MIT Press.

Zwart, J-W. 2006. Complementizer Agreement and Dependency Marking Typology. *Leiden Working Papers in Linguistics* 3: 53-72.