



3 1293 01592 8496

LIBRARY
Michigan State
University

This is to certify that the

dissertation entitled

Localizing Derivational
Economy

presented by

Daehee Lee

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Linguistics

Major professor

Date April 14, 1997

PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.

DATE DUE	DATE DUE	DATE DUE
03-19-20	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

MSU is An Affirmative Action/Equal Opportunity Institution

c:\circ\datedue.pm3-p.1

LOCALIZING DERIVATIONAL ECONOMY IN MINIMALISM

By

Daehee Lee

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Linguistics
and Germanic, Slavic, Asian and African Languages

1997

ABSTRACT

LOCALIZING DERIVATIONAL ECONOMY IN MINIMALISM

By

Daehee Lee

This thesis attempts to uniformly and strictly localize derivational economy, and to investigate the significance and consequence of local economy, providing some natural and unified analyses of the cyclicity in overt derivations, Procrastinate effects, Wh-asymmetries, and Wh-adjunct symmetries in the minimalist program for linguistic theory (Chomsky 1993, 1994, 1995).

First of all, we distinguish between global and local economy, and examine the motivation of global economy on derivations, and its conceptual and empirical problems. Then we pursue local economy in which derivational economy conditions (the Last Resort Condition, the Minimal Link Condition, and the Earliness Principle) should select the most economical operations at each step of derivations. By localizing derivational economy, we get the following desirable results:

- Economy becomes strictly derivational.
- Computational complexity is significantly reduced by generating only a set of optimal derivations.
- The optimality of a derivation is consistent in the course of derivation and at the interface

levels.

- Derivational economy becomes homogeneous in terms of unviolability and locality.

We also propose that the Procrastinate Principle, which is a stipulative and global condition, should be eliminated and replaced with Earliness. Earliness has the following advantages:

- All the derivational economy conditions become localized uniformly.
- The Last Resort Condition becomes strengthened so that it can block 'no operation'.
- The cyclicity of overt computation and Procrastinate effects are derived from one principle, Earliness.

We also hypothesize that multiple features of a target can attract F, and that multiple feature attraction can presumably be parametrized in terms of the number and type of features. Incorporating this with the Minimal Link Condition, multiple feature attraction offers a unified analysis of Wh-asymmetries such as argument-adjunct asymmetries, argument extraction asymmetries, argument-quasi-argument asymmetries, superiority effects, and Wh-adjunct symmetries shown in argument-adjunct asymmetries, pseudo-opacity, and inner island conditions.

Copyright
by
Daehee Lee
1997

Dedicated to my father, Sangwook, my mother, Sochool, my wife, Gemhee, and my three sons, Jaeyoung, Jaegook, and Jaesung.

ACKNOWLEDGMENTS

I am very pleased to express my gratitude to people who have supported me in fulfilling my graduate study and dissertation for five and a half years. Without their support I could not complete my graduate study at all.

First and foremost, I thank all the members of my dissertation committee, Alan Munn (chair), Barbara Abbott, Cristina Schmitt, Alan Beretta, and Mutsuko Endo Hudson, for their strong support and guidance. Especially, I express my deep gratitude to Alan Munn, Barbara Abbott, and Cristina Schmitt.

It was very fortunate that I have been a student of Alan Munn. His consistent guidance, insightful comments, challenging questions and criticisms have been invaluable for my research. In spite of his busy schedule, he has gladly opened his office for me regularly, and very patiently and carefully listened to my little and premature idea for hours, and provided me with very helpful comments on its predictability and the directions of the research. His guidance for reading books and articles, and for organizing the idea has been incredibly helpful. His constant encouragement has also been an invaluable support.

I am grateful to Cristina Schmitt for her excellent comments and questions on my research. By discussing with her, I could make my vague ideas clear. She has never forgotten to encourage me with her good words and confidence in me.

I am greatly indebted to Barbara Abbott for her continuing guidance and strong encouragement. In my early graduate study she provided me with an excellent academic training, and broadened my view of theoretical linguistics. Her comments and questions have been very helpful to improve my research. In classes and seminars she always read for me what she was writing on the blackboard, keeping in mind that I cannot access the blackboard with my blindness. One thing that I could never forget was that she herself read some books and articles on the tape for me.

I am grateful to Seok Choong Song and Kazuhiko Fukushima. They gladly read good books and articles and discussed them together with me.

In addition, I thank all my other professors: Grover Hudson, David Lockwood, Julia Falk, and Carolyn Harford for their support in my early graduate studies.

I am in debt to Jaehyun Han for his guidance in my undergraduate study. By virtue of him I could first open my eyes to theoretical linguistics.

I am also indebted to Office of Programs for Handicapped Students (OPHS) and Tower Guard for their strong

support. Without their reading service I could not have finished my graduate study at all. I give my special gratitude to Michael Hudson, a vision specialist at OPHS, for his friendship, strong confidence in me, constant encouragement, and invaluable information on accessibility on campus.

I also thank my friends at the depart of linguistics for many delightful conversations, good discussions, their friendship and help. I give my special thanks to Laurie Church, Seung-chae Cheong, Ki Yeol Lee, Ok Sook Park and Dennie Hoopingarner for their great assistance.

I am grateful to all the congregations at Korean Lansing United Methodist Church for their fellowship and assistance. My special gratitude goes to Pastor Hyonam Hwang. His good words, prayers and confidence in me have been very encouraging to me.

I am indebted to Deane Blazie and Bryan Blazie at Blazie Engineering for their trust, encouragements, and financial support. They offered me a job in software engineering, while I was a graduate student, and generously allowed me to work remotely from the company. When I was busy with academic research, they kindly assigned some works to someone else. When I was on leave from my position to finish my research, they continued to provide financial support for me.

I am also very truly grateful to my family. My mother,

Sochool, brother, Wonhee, and sisters, Jungok and Sookja, strongly supported me. Especially, their prayers and pride in me have been good encouragements to me.

I never forget Gemhee Lee, my wife's, unselfish love and support for me. She read and scanned a tremendous amount of books and articles for my study, edited my papers, and drew diagrams and figures for me. I interrupted her work and even her sleep at midnight so many times so that I could ask her to look up references for me. Her patience and unselfish love have been and will be the greatest source of support.

Finally, I thank all my three sons, Jaeyoung, Jaegook and Jaesung, who gave up their mother most of time when she helped me. They have been my great pleasure.

TABLE OF CONTENTS

0. Outline of the Thesis	1
1. Introduction to a Minimalist Program for Linguistic Theory	14
1.1 The Minimalist Model	15
1.2 The Lexicon and the Computational System	21
1.2.1 The Lexicon and the Computations: Select	21
1.2.2 Merge	24
1.2.3 The Status of X'-theory	29
1.2.4 Move	32
1.2.5 Delete/Erase	36
1.3 Bare Output Conditions and the Computations	39
1.3.1 Features and Their Interpretability	40
1.3.2 Spell-out, and PF and LF Branching	41
1.3.3 Feature Checking and Move-F	43
1.4 The Economy Principles	57
1.4.1 Last Resort	57
1.4.2 Minimal Link Condition	60
1.4.3 Procrastinate Principle	61
2. Localizing Derivational Economy	63
2.1 Introduction	63
2.2 Global Economy: the Motivations and Problems	65
2.2.1 A Distinction between Global and Local Economy	65
2.2.2 The Last Resort Condition and Global Economy	67
2.2.3 The Minimal Link Condition and Global Economy	69
2.2.4 The Procrastinate Principle and Global Economy	72
2.2.5 Some Problems with Global Economy	73
2.3 Localizing Derivational Economy	81
2.3.1 Localizing the Last Resort Condition	82
2.3.2 Localizing the Minimal Link Condition	85
2.3.3 Earliness as a Local Economy Condition	87
2.4 Summary	92
3. Deriving Strict Cycle and Procrastinate Effects	95
3.1 Introduction	95
3.2 Deriving Strict Cycle	98
3.2.1 Previous Analyses and their Problems	98
3.2.1.1 Extension Condition	98
3.2.1.2 Target- α	105
3.2.1.3 Crossing the Number of Nodes	111
3.2.1.4 Feature Strength and Cyclicity	113

3.2.1.5 Chain Interleaving	114
3.2.2 Earliness and the Strict Cycle	116
3.3 Procrastinate: the Background and Problems	117
3.3.1 Motivations	117
3.3.2 Some Problems	124
3.4 Deriving Procrastinate Effects	127
3.4.1 PF Deletion Analysis	127
3.4.2 Earliness and Procrastinate Effects	134
4. A Unified Analysis of Wh-Asymmetries and Wh-Adjunct Symmetries	139
4.1 Introduction	139
4.2 Some Types of Wh-asymmetries	140
4.2.1 Wh-asymmetries and Pre-minimalist Analyses	140
4.2.2 Some Minimalist Analyses	148
4.3 A Unified Analysis of Wh-asymmetries	155
4.3.1 Feature Specifications of Wh-words and [+Wh] Comps	155
4.3.2 Multiple Feature Attraction	160
4.3.3 Analysis	164
4.3.3.1 Some Basic Assumptions	164
4.3.3.2 Argument-Adjunct Asymmetries	166
4.3.3.3 Argument Extraction Asymmetries	170
4.3.3.4 Argument-Quasi-Argument Asymmetries	171
4.3.3.5 Superiority Effects and Some Residues	175
4.4 Further Consequences: Some Adjunct Symmetries	181
5. Conclusion and Further Research	184
REFERENCES	187

0. Outline of the Thesis

The spirit of economy has long had effect on the theory of generative grammar. In the early theory of grammar it was reflected by simplifying the rule system of language, i.e. reducing complex and superfluous rules into simple universal principles. For example, language-particular and construction-specific phrase structure rules were reduced into a simple X' -theory; construction-specific transformational rules were simplified into one generalized rule, Move- α ; a variety of descriptive islands on extraction were generalized into the Empty Category Principle; and so on.

The minimalist program for linguistic theory (Chomsky 1993, 1994, 1995) pursues the spirit of economy in two different ways. One is that the language system is so perfect that universal principles cannot be overlapped in their effects; if so, some of them may be wrong. In addition, the components of the language system must be virtually conceptually necessary. The existence of D-structure and S-structure was, so to speak, motivated by some theory-internal necessity rather than by virtual conceptual necessity. Thus they have been eliminated from the minimalist program.

The other way economy is used in the minimalist program is that the language system is so perfect that a structural description must be derived in an optimal way. If more than one structural description can be derived by the computations, only an optimal derivation will be selected by the language system, blocking nonoptimal derivations. It indicates that economy considerations should have effect on grammaticality, and hence that some universal principles should reflect economy considerations.

To reflect economy considerations for language analysis in universal grammar, Chomsky (1991, 1993, 1994, 1995) proposes some economy conditions on derivations such as the Greed Principle, the Minimal Link Condition, and the Procrastinate Principle. Since then many studies have been undertaken to discover the properties of derivational economy. Most of the studies are, however, closely related to global economy in which economy conditions apply at the interface levels or representations, which is empirically and conceptually undesirable in language (which we return to later).

The goal of this thesis is to strictly and uniformly localize derivational economy in the minimalist program (Chomsky 1993, 1994, 1995) so that derivational economy should apply at each point of a derivation. Furthermore, the thesis investigates the significance and consequences of local economy, giving some natural and unified explanations

of cyclic derivations in overt syntax, Procrastinate effects, some Wh-asymmetries, and some Wh-adjunct symmetries under local economy considerations. The thesis is organized as follows:

Chapter 1 gives a concise introduction to the minimalist program (Chomsky 1993, 1994, 1995) for the nonspecialist. It describes some basic concepts and machineries of the minimalist model. The first section (section 1.1) makes a brief sketch of the general model of minimalism from a bird's eye point of view, and the subsequent sections then describe the basic modules of the model in more detail. Section 1.2 introduces bare phrase structures and computational operations (i.e. Select, Merge, Move, and Delete/Erase), and explains how the computational system accesses lexical items and constructs phrase structures by means of the computational operations without reference to X'-formats. Section 1.3 explains the relationship between derivations and bare output conditions in two respects: (i) how the computational system maps phrase structures to PF and LF, and (ii) which interface properties motivate the computations. More specifically, we discuss types of features, feature interpretability at the interface levels, Spell-out and Move-F/Attract-F. Section 1.4 describes economy principles on derivations, i.e. the Last Resort Condition, the Minimal Link Condition, and the Procrastinate Principle, and explains how they apply for

optimal derivations.

Chapter 2 sets up a theoretical basis for local economy on derivations. First of all, section 2.2 distinguishes global economy from local economy, as defined in (1) and (2), respectively:

(1) Global Economy

Derivational economy should apply at the interface levels so that it selects a derivation (among convergent derivations) that takes the most economical operations.

(2) Local Economy

Derivational economy should apply at each point of a derivation so that it selects the most economical operation to affect the target at that point.

Then this section discusses the motivation of global economy, and its empirical and conceptual problems in comparison with local economy, as listed in the table below:

	Global	Local
i	It is a kind of representational condition.	It is a strictly derivational condition.
ii	It allows the computational system to generate an explosive or exponential number of derivations at the interface levels.	It allows the computational system to generate only a set of optimal derivations at the interface levels.
iii	Some derivations which were optimal during the time of derivation may become nonoptimal at the interface levels.	Some derivations which are optimal during the time of derivation are always optimal at the interface levels.
iv	It makes economy conditions heterogeneous in terms of unviolability and locality.	It makes economy conditions homogeneous in terms of unviolability and locality.

Localizing derivational economy means that measuring the cost of the computational operations should be done at each point of a derivation. Section 2.3 explores how to measure the cost of the computational operations in a local way. Section 2.3 claims that derivational economy should adopt (4) and (6) for the measurement of the cost of an operation rather than (3) and (5) which motivate global economy.

- (3) The more operations a derivation takes, the more costly it is.
- (4) The more superfluous operations a derivation takes, the more costly it is.
- (5) Merge is costfree, and Move is costly.

(6) Merge and Move are both equal in terms of cost.

As a consequence of (4), "no operation" cannot any longer block necessary (or last resort) operations. Rather, a last resort operation blocks "no operation", incorporating with the Earliness Principle (which we return to later).

In addition, proposal (4) naturally leads us to proposal (6). Since Merge and Move are equal in cost, they are both costly if they perform in a superfluous way; otherwise they are both considered costfree. This proposal implies that Merge and Move cannot compete with each other under economy considerations, since their functions are fundamentally different. An operation, Merge, applies to reduce the number of partial phrase structures into one larger phrase structure for a derivation to converge at PF and LF; otherwise it would crash, since partial phrase structures which are not related in terms of dominance and c-command cannot be interpreted, i.e. not linearized by Linear Correspondence Axiom (Kayne 1994) at PF, nor semantically interpreted by composition at LF.

On the other hand, an operation, Move, functions to satisfy morphological properties at the interface levels, providing a local checking relation in a phrase structure; otherwise a derivation would crash, since some morphological formal features of a derivation cannot be interpreted at the interface levels.

In this section we propose a timing principle, the Earliness Principle, as defined in (7), and claim that Earliness should replace the Procrastinate Principle which is a global condition in nature. We consider how Earliness applies locally along with Attract-F.

(7) Satisfy bare output condition as early as possible.

More specifically, we formulate Attract-F to reflect Earliness in it:

(8) K attracts F early only if a sublabel of K is an uninterpretable feature at the interface level that Attract-F affects.

For a derivation to be optimal at the interface levels, it should satisfy all three types of derivational economy conditions at each point of a derivation: the Earliness principle as in (8), the Last Resort Condition and the Minimal Link Condition, as Chomsky (1995) defines in the following:

(9) The Last Resort Condition

K attracts F if F can enter into a checking relation with a sublabel of K.

(10) The Minimal Link Condition

- (i) K attracts F if F is the closest feature to K.
- (ii) X is closer to K than y if K c-commands x and x c-commands y.

At each point of a derivation the computational system now selects the most economical operation in a strictly local way, generating only a set of optimal derivations regardless of whether they converge at an interface level.

The subsequent chapters (chapter 3 and 4) attempt to offer some unified analyses of the cyclicity in overt derivations, Procrastinate effects, Wh-asymmetries, and Wh-adjunct symmetries under local derivational economy.

In chapter 3 we apply Earliness to some phenomena of the cyclic derivations in overt syntax, and Procrastinate effects, demonstrating that the cyclicity of the computations and Procrastinate are reducible to one timing principle, Earliness. Section 3.2.1 makes a brief sketch on previous efforts to derive the cyclicity of overt computations from some economy principles. Their analyses are done under the global economy considerations and also the assumption that Procrastinate should exist. In section 3.2.2 we derive the cyclic computations in overt syntax from Earliness.

Section 3.4 demonstrates how Procrastinate effects are derived from Earliness in a local way. If we can eliminate the Procrastinate Principle, which is global in nature, then

we can uniformly localize all derivational economy conditions.

In order to eliminate Procrastinate, section 3.3 discusses the motivations of Procrastinate and its problems. First of all, Procrastinate has two stipulations in comparison with other universal principles:

One is that Procrastinate is violable for convergence, while no other universal principles including other economy principles such as the Last Resort Condition and the Minimal Link Condition can be violated for any reason. Its violability is not consistent with the general assumption that all universal principles should be observed for convergence, and that if a derivation violates any principle it should yield some deviance.

The other stipulation is that only Procrastinate is global in nature, while other economy principles can be localized. Its global characteristic is undesirable (as will be described in section 2.2).

In addition, Procrastinate has the following problems: (i) As a timing principle it cannot explain the timing of the computations in overt syntax; (ii) Its conceptual motivation is based upon some characteristic of the sensory-motor system rather than on linguistic properties.

In section 3.4 Earliness derives Procrastinate effects such as English verb movement and object shift and French object shift without reference to Procrastinate at all.

Chapter 4 attempts to give a unified analysis of some asymmetries of Wh-movement and some symmetries of Wh-adjunct movement under the Minimal Link Condition and multiple feature attraction in a local way. We make the following proposals relating to multiple feature attraction:

- (11) K attracts F where the number of features F and types of F are parametrized.

Parametrizing the features F of Attract-F completely fits minimalism in which only lexical items and their morphological properties must be idiosyncratic language to language, and all universal principles must be invariant. In this sense parameters (or options) for a language must be specified in terms of formal features.

As a consequence we reduce the Wh-asymmetries and Wh-adjunct symmetries to the Minimal Link Condition and multiple feature attraction without reference to the non-formal features of categories such as referential/nonreferential θ -role, etc.

In section 4.2 we introduce some asymmetries of Wh-movement such as argument-adjunct asymmetries, argument-quasi-argument asymmetries, argument extraction asymmetries, and superiority effects, and discuss their previous (pre-minimalist and minimalist) analyses and problems. The previous analyses could not treat the Wh-asymmetries and Wh-

adjunct symmetries in a unified way, and needed to refer to some semantic information such as thematic roles, referentiality/nonreferentiality, etc. which are undesirable to refer to during the derivation in the minimalist model.

Section 4.3 makes a unified analysis of those asymmetries under the Minimal Link Condition and Attract-F which are independently motivated in language. In Section 4.3.1, first of all, we elaborate the feature specification of Wh-words and a Comp. In this section we classify Wh-words into three types of categories: Wh-DP operators, Wh-adverbial operators, and Wh-pronominal variables, and their differences are specified in terms of formal features:

- (12) a. Wh-DP operators: $\{D, Op_Q\}$
- b. WH-adverbial operators: $\{Adv, Op_Q\}$
- c. Wh-pronominals: $\{D_{Pro}\}$.

Regarding the formal features of a [+Wh] Comp, we propose the following:

- (13) A Comp attracts F where F is either an Operator Op feature or a pair of features $\langle D, Op_Q \rangle$.

Section 4.3.2 considers how Attract-F and the Minimal Link Condition interact with each other in minimalism. Specifically, we demonstrate that the Minimal Link Condition

determines optimal derivations relative to features to be attracted.

Section 4.3.3 demonstrates how Attract-F and the Minimal Link Condition provide a unified analysis of the Wh-asymmetries. Under our analysis the Wh-asymmetries are due to the asymmetries of the availability of the D and Op_0 feature of a Wh-word under multiple feature attraction and the Minimal Link Condition. That is, a [+Wh] Comp attracts F where F is Op or $\langle D, Op_0 \rangle$. A feature Op can attract any category with an Op feature, (i.e. Wh-adjuncts, Quasi-arguments, Wh-NPs), while $\langle D, Op_0 \rangle$ can attract only a category with both a D feature and an Op_0 feature (i.e. Wh-NPs). Under the Minimal Link Condition, however, an Op feature cannot attract another Op feature across an intervening Op or $\langle D, Op_0 \rangle$, while $\langle D, Op_0 \rangle$ cannot attract another $\langle D, Op_0 \rangle$ across an intervening $\langle D, Op_0 \rangle$ but can attract it across Op. The former is a typical case where Wh-adjuncts cannot move across an intervening operators, and the latter a case where Wh-NPs cannot move across an intervening Wh-NP.

Section 4.4 extends this analysis to Wh-adjunct symmetries, as shown in argument-adjunct asymmetries, pseudo-opacity, and inner islands where Wh-adjuncts cannot move across any other intervening operator.

To conclude, local economy is empirically and conceptually superior to global economy. Conceptually, we

can reduce computational complexity, and keep homogeneous derivational economy conditions under local economy. We can also derive the cyclicity of overt derivations and the Procrastinate Principle from one timing principle, the Earliness Principle. Under the Minimal Link Condition and multiple feature attraction, on the other hand, we can uniformly treat some Wh-asymmetries and Wh-adjunct symmetries.

1. Introduction to a Minimalist Program for Linguistic Theory

This chapter will briefly review the framework of the minimalist program for linguistic theory (Chomsky 1993, 1994, 1995). Although the main chapters of this dissertation discuss in more detail some concepts of the minimalist program relating to each chapter's topic when necessary, the introductory review here will provide some theoretical background for understanding basic concepts and machineries of minimalism.

In section 1.1 the minimalist model of grammar will be described in the general sense of conceptual necessity. The subsequent sections will describe the components of grammar in more detail. Section 1.2 introduces bare phrase structures and computational operations (i.e. Select, Merge, Move, and Delete/Erase), and explains how the computational system selects lexical items and constructs (or derives) phrase structures by means of the computational operations without reference to X'-formats. Section 1.3 describes the linguistic representational levels, bare output conditions, and the relationships between bare output conditions and derivations in two respects: (i) how a phrase structure is mapped to the interface levels, PF and LF, and (ii) which

interface properties motivate the computations. More specifically, it discusses types of features, feature interpretability at the interface levels, Spell-out, and move-F/Attract-F. Section 1.4 introduces some economy conditions on derivations such as the Last Resort Condition, the Minimal Link Condition, and the Procrastinate Principle.

1.1 The Minimalist Model

Chomsky (1993, 1994, 1995) proposes the minimalist program as the principles-and-parameters model in which particular languages are assumed to be determined by a finite set of universal principles and parameters. Universal principles are invariant and common to all human language faculties, and parameters (or options) are "restricted to functional elements and general properties of the lexicon" (Chomsky 1994 p.4), and determined by very limited linguistic experience only.

The minimalist program has been designated to accommodate only conceptually necessary or minimally required concepts for a theory of grammar. What elements are conceptually necessary and minimally required for linguistic theory, then?

First of all, one of the minimal theoretical requirements is the large repository which stores the lexical items with idiosyncratic properties including

phonological, morphological, (sub)categorial, and semantic specifications. For example, in English the word "tree" means a tree, not a car; it is pronounced as [tri:], not [ka:]; the verb "buy" obligatorily requires an object, and the verb "arrive" does not; etc. Such arbitrariness of lexical items cannot be computed at all, and must be somehow specified in a storage--what we may call a lexicon. Any theory of grammar must thus have a lexicon.

The second requirement for linguistic theory is a computational system. Since the lexicon itself is a storage device with some morphological processes, the theory requires the computational system to construct larger units such as phrases and clauses. The lexicon and the computational system belong to the generative or computational procedure of language faculty.

The computational system selects an array of lexical items from the lexicon and generates structural derivations. The derivations which the computational system generates affect the sound and meaning. For example, the sentence "John kissed Mary." does not mean "Mary kissed John."; the sound pattern, e.g. intonation of "green house" is different from that of "greenhouse". For this reason Chomsky (1993, 1994, 1995) proposes that the output of the computational system should be interpreted at two interface levels, PF and LF, for sound and meaning, respectively. In addition, only the information relating to sound is interpreted at PF, and

only the information relating to meaning is interpreted at LF, but not vice versa. In this sense the computational system should take on two responsibilities: one is that it should keep the history of a derivation for the interface levels; the other that it should generate only the elements which are interpretable at the interface levels. It is a computational effort to satisfy the output conditions.

The above discussion implies that the theory of grammar requires at least two representational levels, PF and LF. PF and LF are assumed to be further fed to two external systems, an articulatory-perceptual external system, and a conceptual-intentional external system, respectively. In addition, Chomsky (1993, 1994, 1995) argues that only PF and LF are minimally required for linguistic theory, and that D-structure and S-structure, which were assumed in traditional generative grammar, can be eliminated if we can reduce the conditions on D-structure and S-structure to the ones on PF, LF, and derivations. So the minimalist model no longer takes D-structure and S-structure for granted.

Further, the computational system should not arbitrarily derive phrase structures to simply satisfy the output conditions. Rather, it follows some conditions on derivations in the computational process. For example, sentence (1) is grammatical, and sentence (2) is not, although they are both derived from the same lexical choices, and presumably satisfy Full Interpretability (FI)

at the interfaces such as Case theory, θ -theory, the Uniform Chain Condition, the Extended Projection Principle, etc.

The ungrammaticality of (2) is presumed to be due to violating some condition on derivations. Thus derivations must satisfy some conditions on derivations and some output conditions at the same time, in order to be grammatical.

(1) It seems that John_i is believed t_i.

(2) *John_i seems that it is believed t_i.

Universal grammar (UG) will take the following computational procedure to map a phrase structure to PF and LF: First of all, an array of lexical items are chosen from the lexicon. Then the computational system selects the lexical items from the array freely at any point of a derivation before PF and LF branches, and constructs phrase structures, satisfying some derivational conditions. At any point of the course of the derivation, the computational system switches them into PF, which is what we may call the Spell-Out operation. Then the computation maps the structures into a component of Morphology, and further into PF. The computation which maps the phrase structures to the PF representation after the Spell-Out may be called the PF computation.

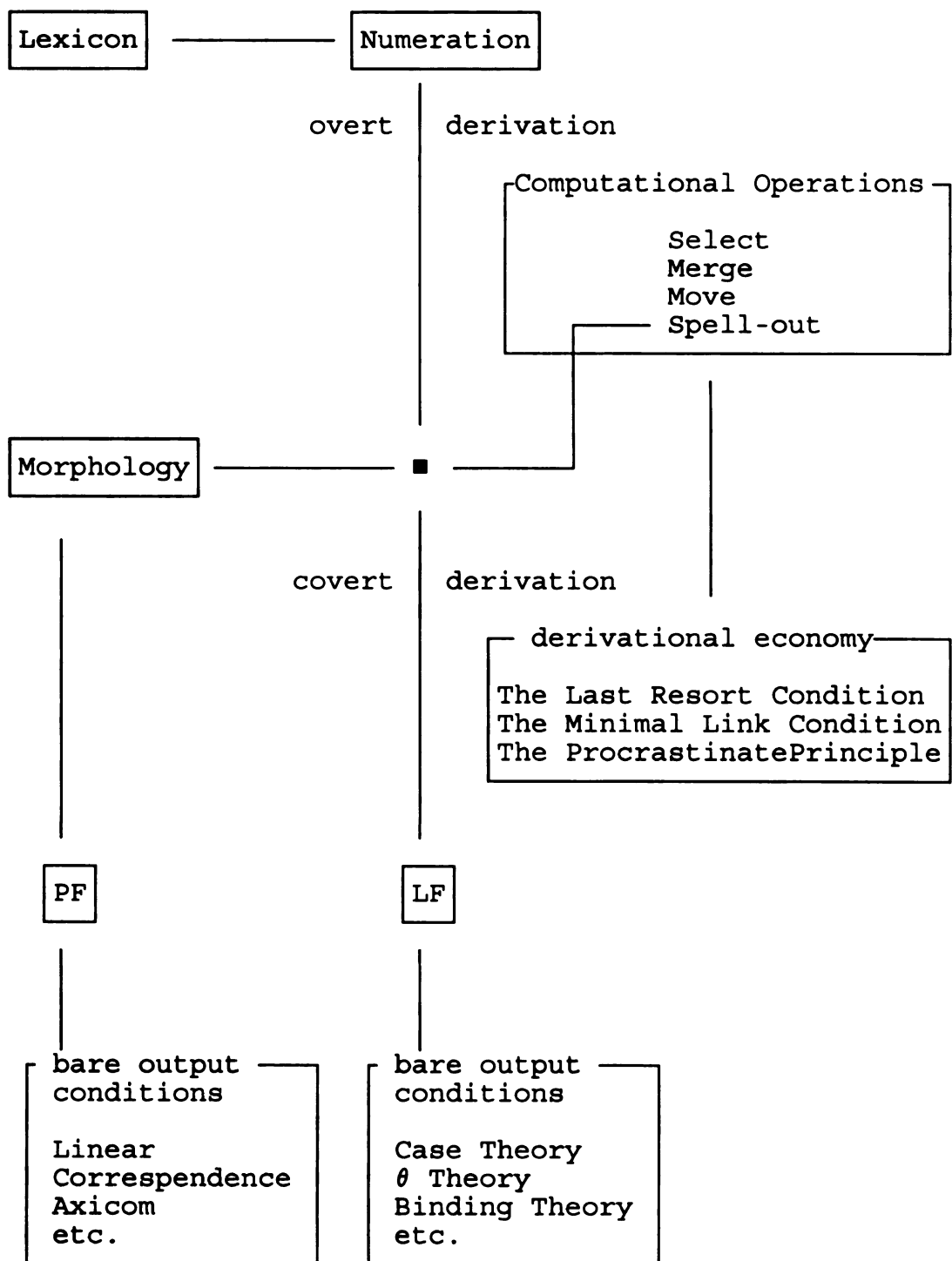
On the other hand, independent of the PF computation, the computational system continues to further modify the

phrase structures and map them into LF. This may be called the LF computation. Chomsky (1993, 1994) claims that the PF and LF computations cannot further access the array of lexical item or the lexicon¹. The computation before Spell-Out may be called an overt syntactic computation, and the LF computation may be called the covert computation, since the syntactic structures modified by the LF computation are not reflected in pronunciation. Note that the overt computation and LF computation (or covert computation) are a single uniform computational system, and hence there is no difference between the overt and covert computations at all except for whether the results of the computation are perceptual or not.

In sum, the minimalist model consists of a lexicon, a computational system, two levels of linguistic representations (PF and LF), and some principles on derivations and representations, which can be diagrammed in (3).

¹Chomsky (1995) claims that phonetically null lexical items may be accessed and merged to the root of a phrase structure even after Spell-out.

(3)



Thus Chomsky (1994) proposes that a language should be specified in terms of "... the nature of the computational procedure; ... the properties of bare output conditions and

the functional component of the lexicon; and ... principles and concepts." (p.5)

In the subsequent sections let us take a look at some of these properties in a little more detail.

1.2 The Lexicon and the Computational System

In this section we will consider the mechanisms and properties of four computational operations: Select, Merge, Move, and Delete/Erase, which are all assumed to be conceptually necessary for the language faculty, and discuss bare phrase structures (Chomsky 1994, 1995), deriving X'-theory from other principles and therefore eliminating it from the grammar.

1.2.1 The Lexicon and the Computations: Select

The computational system generates a linguistic expression $\langle P, L \rangle$ where P refers to an expression for PF and L refers to an expression for LF. P and L are assumed to be constructed from the same lexical choices, since, for example, the sound of a sentence John kissed Mary does not mean that a dog chased a cat. The lexical choices are assumed to be done at two levels. At one level lexical items are selected into an array from the lexicon. This is done all at once before the computations proceed to

construct a phrase structure. At the other level, in the course of deriving a phrase structure, the computational system selects lexical items from the array rather than from the lexicon. In this section let us consider the computational operations to retrieve lexical items from the lexicon to an array, and to retrieve them from the array to a derivation, respectively. These operations are conceptually necessary to interface the lexicon and the computational system.

Retrieving lexical items from the lexicon forms a set of a pair $\langle \text{LI}, i \rangle$ in an array where LI is a lexical item and i is the number of times that LI has been retrieved from the lexicon. The array is called a numeration of lexical items. For example, for a sentence John saw Mary the numeration n is as follows:

- (4) $n = \{ \langle C, 1 \rangle, \langle T, 1 \rangle, \langle \text{John}, 1 \rangle, \langle \text{Mary}, 1 \rangle, \langle \text{saw}, 1 \rangle \}$ (where C and T are functional categories for Comp and Tense, respectively.)

It is important to note that the numeration n must be finished all at once before it is mapped to PF and LF. This condition can be defined in (5).

- (5) Inclusiveness Condition: (=Chomsky(1995) p.228)

Any structure formed by the computation is

constructed of elements already present in the lexical items selected for n .

After the numeration is done, the computational system starts to build a phrase structure, selecting the lexical items from n , and introducing them to a derivation. This operation can be called Select. The computational system selects an lexical item, LI , (i.e. accesses $\langle LI, i \rangle$ in the numeration, and reduces i by 1), and performs permissible computations for derivations.

For example, suppose that a numeration n is completed as in (4), and that the computational system derives a partial phrase structure $[_{VP}John [_V.saw Mary]]$. First, an operation Select accesses $\langle Mary, 1 \rangle$ and $\langle saw, 1 \rangle$, reduces each index by 1, and performs a computation to construct $[_V.saw Mary]$. After this process, the numeration n looks like (6):

$$(6) \ n = \{ \langle C, 1 \rangle, \langle T, 1 \rangle, \langle John, 1 \rangle, \langle Mary, 0 \rangle, \langle saw, 0 \rangle \}$$

After that, Select accesses $\langle John, 1 \rangle$, and reduces i by 1, and a further computation constructs $[_{VP}John [_V.saw Mary]]$. Then the numeration n looks like (7).

$$(7) \ n = \{ \langle C, 1 \rangle, \langle T, 1 \rangle, \langle John, 0 \rangle, \langle Mary, 0 \rangle, \langle saw, 0 \rangle \}.$$

If i is a zero in $\langle LI, i \rangle$, then Select can no longer access that LI . Furthermore, unless all i 's in N are exhausted and so become 0, a derivation cannot be done nor generated.

Note that the Select operation is assumed (by Chomsky (1995)) to be costless in the sense of the economy conditions which we will discuss later.

1.2.2 Merge

When the computational system retrieves the lexical items from a numeration n by Select, it concatenates or merges them into a larger unit. This operation is called Merge. This operation is conceptually necessary to build a unit larger than a word. An operation Merge can be defined as in (8).

(8) Merge: (= Chomsky (1995) p. 243)

- a. take only two syntactic objects, x and y ;
- b. form a larger syntactic object, $z = \{w, \{x, y\}\}$;
- c. eliminate x and y .

(8.a) defines Merge to be a binary operation². It follows from this that a non-branching projection $X \rightarrow X'$ is no longer a valid operation, although it was permissible

²Kayne (1994), Collins (1995) and Watanabe (1995) argue that the binary property of Merge can be reduced instead of being defined in it.

under X'-theory. (8.b) indicates that it creates a new larger category (by projecting one of the two merging categories.) (8.c) means that it deletes two merging categories from partial phrase structures after creating a new category. So Merge is understood as an operation to reduce the number of phrase markers (or syntactic objects)³. Actually, Merge iterates until a single syntactic object is left⁴. We will see an example of this in detail later.

The syntactic objects can be defined as in (9).

(9) (CF. (5), Chomsky 1995 p.243)

a. lexical items

b. $z = \{w, \{x, y\}\}$, where x , y , and z are objects
and w is the label of z .

First of all, let us look at the form, $z = \{w, \{x, y\}\}$. Z is a set which is constituted of x and y , and understood as a phrase marker. x and y are called terms. W is the label of z , representing the type of z . W is determined by projecting either x or y exclusively or asymmetrically. For any structure K the terms can be defined as in (10).

³Bobaljik (1995) takes a different position for Merge. According to him, Merge does not eliminate the merging categories, but simply creates a new category. Thus all partial phrase structures are accessible by Merge for further computations, although they are once merged and contained in a larger category. See Bobaljik (1995) for detail.

⁴This fact can be reduced from Kayne's (1994) Linear Correspondence Axiom (LCA). See Kayne (1994) for details.

(10) (= (10), Chomsky 1995 p.247)

- a. K is a term of K.
- b. If L is a term of K, then the members of the members of L are terms of K.

Let us take some examples of Merge. Let $x = [_v\text{saw}]$, $y = [_N\text{Mary}]$, and x be projected. Then $z = \{w, [x, y]\}$ can be computed as in (11), and informally diagrammed as in (12)⁵.

- (11) a. Merge $[_v\text{saw}]$ and $[_N\text{Mary}]$. $\rightarrow$ (8.a)
- b. Create $V' = \{V, \{[_v\text{saw}], [_N\text{Mary}]\}\}$. $\rightarrow$ (8.b)
- c. Eliminate $[_v\text{saw}]$ and $[_N\text{Mary}]$. $\rightarrow$ (8.c)

(12) V' (=V-type)

$$\begin{array}{cc} / & \backslash \\ [_v\text{saw}] & [_N\text{Mary}] \end{array}$$

As defined in (9) $V' = \{V, \{[_v\text{saw}], [_N\text{Mary}]\}\}$, $[_v\text{saw}]$, and $[_N\text{Mary}]$ are all syntactic objects, and V' has a label of V (i.e. a V-type syntactic object). Also, as defined in (10), V' , $[_v\text{saw}]$, and $[_N\text{Mary}]$ are the terms of V' . In (11) V' is a simple object, since the terms $[_v\text{saw}]$ and $[_N\text{Mary}]$, are terminal strings or lexical items. However, if $x = [_N\text{John}]$, $y = V'$ in (11), and y is projected, then the form

⁵Following Longobardi (1994) proper nouns should be treated as a DP like $[_{DP}D [_{NP}\text{Mary}]]$. But we will here assume proper nouns to be NP's just for simplicity, because it is irrelevant for our discussion.

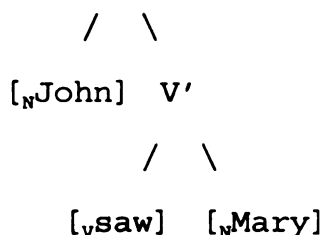
$z = \{w, \{x, y\}\}$ can be a complex form as in (13).

(13)⁶ a. Merge [_NJohn] and V' in (11). -> (8.a)

b. Create VP = {V, {[_NJohn], V'}}. -> (8.b)

c. Eliminate [_NJohn] and V'. -> (8.c)

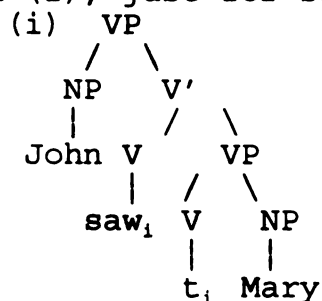
(14) VP (=V-type)



In (13) V' is a term of VP, and so all the terms of V' are also the terms of VP, as defined in (10).

Turning to (8), we have to make (8.c) clear. For this we have to assume that there should be a set of partial structures for a derivation which is accessible by the computational system. This set is different from a numeration. Select accesses the lexical items in the numeration, and those lexical items are entered into a set of partial phrase structures for a derivation. The

⁶Although Chomsky (1995) assumes a VP shell for transitives, as in (i), just for simplicity we will ignore it for a moment.



computational system manipulates the objects in a set of partial structures. In the case of (11), for example, the computational system accesses $\langle [{}_V\text{saw}], 1 \rangle$ and $\langle [{}_N\text{Mary}], 1 \rangle$ in the numeration, reduces their indices by 1, and puts $[{}_V\text{saw}]$ and $[{}_N\text{Mary}]$ into a set of partial structures for a derivation. Then the set S of partial phrase structures is:

$$(15) S = \{ [{}_V\text{saw}], [{}_N\text{Mary}] \}.$$

If only the suboperations (8.a) and (8.b) of Merge apply to the set S , then the set S changes into S' as in the following:

$$(16) S' = \{ [{}_V\text{saw}], [{}_N\text{Mary}], V' \}.$$

If (8.c) applies to S' , then:

$$(17) S'' = \{ V' \},$$

since it eliminates $[{}_V\text{saw}]$ and $[{}_N\text{Mary}]$.

As a consequence of (8.c), Merge should apply only at the root. If all the indices in the numeration become zeros, and the set of partial phrase structures contains only one object, then the derivation can be a potentially legitimate object for the interface levels. In other words, all lexical items in the numeration must be contained in one

phrase structure in order to be interpreted at PF and LF.

Chomsky (1995) claims that, like Select, Merge should also be costless in terms of economy on derivations.

1.2.3 The Status of X'-theory

Chomsky (1970, 1986), Jackendoff (1977), and others developed the X'-schema, recognizing the endocentricity of syntactic categories (N, V, A, P, I, and C), the inherent properties between a head and its maximal phrase, and the structural parallelism across syntactic categories. As a consequence we could eliminate the redundancy of lexical properties and phrase structure rules, and language-specific construction rules along with the concept of the parameter of headedness of universal grammar, and develop some properties of local domain and relations in syntax. In the minimalist framework, however, Chomsky (1994, 1995) reconsiders X'-theory on the assumption that even the X'-format is derivable from other properties and so is eliminable from the grammar.

Chomsky (1994) argues that categorial projections should be understood as "relational properties of categories, not inherent to them" (Chomsky (1994) p.9). So whether a category is a maximal, minimal or intermediate projection should be determined in the structure where it occurs. Given a phrase marker, maximal and minimal

projections are defined in (18).

(18) (=Chomsky (1994) p.10)

- a. A category that is not any further projected is a maximal projection XP.
- b. A category that is not a projection at all is a minimal projection X^0 .
- c. Any other projection is an intermediate projection X' (which is invisible for the computations and the interface levels).

If a lexical item [_NJohn] is selected from the lexicon, for example, in traditional generative grammar it should always be projected as in (19) by a nonbranching operation in order to satisfy X' -theory.

- (19)
- a. [_NJohn]
 - b. [_{N'} [_NJohn]]
 - c. [_{NP} [_{N'} [_NJohn]]]

However, this is no longer true in the minimalist program. The computations such as Select and Merge do not perform a nonbranching projection at all. As we will see later, Move and Delete/Erase do not render a nonbranching projection, either. It is not defined in the minimalist program.

Now let us take (11) and (13) into consideration again. In (11) the term $[_N\text{Mary}]$ is understood as a maximal and minimal projection at the same time, since it is not further projected and not a projection at all, as defined in (18). The term $[_V\text{saw}]$ is a minimal projection, since it is not a projection at all; but it is not a maximal projection, since it is projected to V' . The projected category V' is not a minimal projection, since it is a projected category, and it is a maximal projection, since it is projected but not further projected. The status of V' is a maximal projection in the minimalist program. Without confusion, (11) can be expressed in (20).

$$(20) \text{ VP} = \{V, \{[_V\text{saw}], [_N\text{Mary}]\}\}.$$

In the case of (13), the term $[_N\text{John}]$ is minimal and maximal like $[_N\text{Mary}]$ in (11). However, the term V' in (13) is not a maximal projection at this time, since it is further projected to VP. It is not minimal, either, since it is a projected category. So V' is understood as an intermediate category which is not visible to the computational system for further access.

As we have seen above, the status of categories is differently interpreted at different stages of the computation, depending upon the categorical relation with other terms in a structure.

1.2.4 Move

An operation Move is also assumed to be conceptually necessary to rearrange the order of phrases. The Move operation can be defined as in (21).

(21) (=Chomsky (1995) p.250)

Suppose the category z with terms x and y . Then:

- a. take x ;
- b. target y ;
- c. raise x ;
- d. form a category $l = \{w, \{x, y\}\}$;
- e. replace y in Z with l ;
- f. form a chain, (x_i, t_i) .

Note that the operations in (21.a-f) are the internal suboperations of Move. So the Move operation itself should be a single operation, and so the suboperations cannot be interrupted, and the intermediate derivations that the suboperations may generate are not accessible by other computations. Note that for Move, the projection (e.g. w for l in (21.d)) is predictable (i.e. a target must be always projected,) while it may be fixed in language l for Merge. (See Chomsky (1995) chapter 4 for details.)

Although it is not yet clear which conditions the Merge operation should satisfy, the Move operation is required to

satisfy some principles of UG. First of all, unlike other operations, it is subject to the economy conditions such as the Last Resort Condition, the Minimal Link Condition, and the Procrastinate Principle, (which will be discussed in section 1.4). Second it should satisfy some conditions on chain formation as in (22).

- (22) a. c-command: a head of a chain must c-command its trace.
- b. uniformity condition: (= (17)) a chain must be uniform with regard to phrase structure status, where the phrase structure status of an element is its relational property of being maximal, minimal or neither.

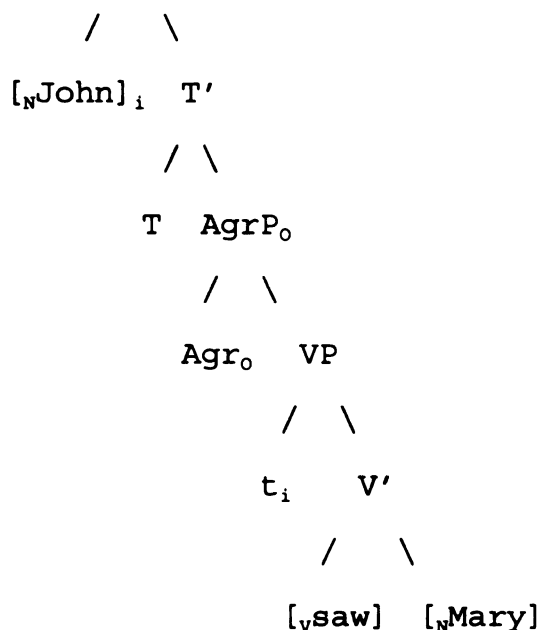
The conditions on chain formation implies two important things for Move: (i) Move must raise (cannot lower) a syntactic object; (ii) Move must project the targeting syntactic object (i.e. it can never project the raising syntactic object).

On the other hand, Move leaves a trace. The trace is understood as an identical copy of the head of the chain. The copy theory of Move accounts for reconstruction effects at LF. (See Chomsky (1993) for the consequences of the copy theory for Move.)

Now let us take object raising for the example of Move.

In the structure of (23), the object [_NMary] raises to target AgrP_o at LF, taking the internal operations of Move in (21), forming the structure of (25), where irrelevant elements and operations are ignored⁷.

(23) a. TP



- b. TP = {T, {John, T'}}
- T' = {T, {T, AgrP_o}}
- AgrP_o = {Agr_o, {Agr_o, VP}}
- VP = {T_i, V'}
- V' = {V, {[_vsaw], [_NMary]}}

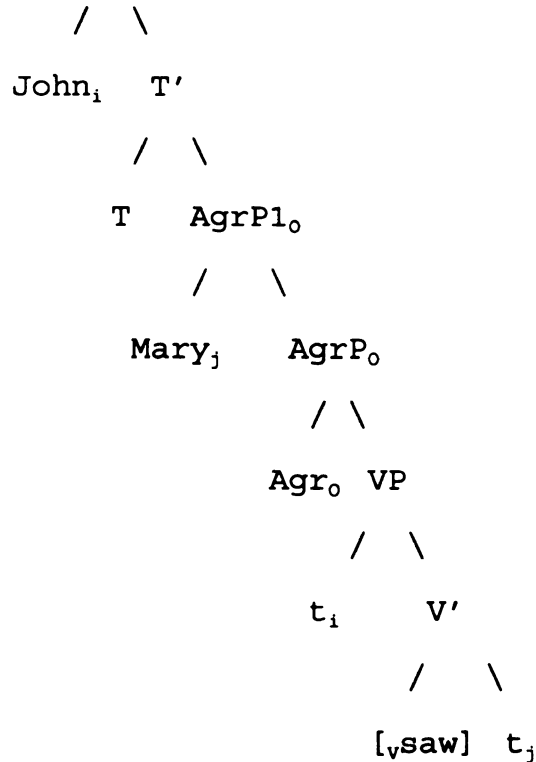
(24) a. Take [_NMary];

b. Target AgrP_o;

c. Raise [_NMary];

⁷Chomsky (1995) no longer takes for granted Agr_s and Agr_o as independent functional categories. See Chomsky (1995), section 4.10, for detail.

- d. Form $\text{AgrP}_{1_0} = \{\text{Agr}_0, \{[_N\text{Mary}], \text{AgrP}_0\}\}$;
 e. Replace AgrP_0 in $\text{TP} = \{\text{T}, \{\text{T}, \text{AgrP}_0\}\}$ with
 AgrP_{1_0} ;
 f. Form a chain $[_{\text{AgrP-}0}\text{Mary}_j \dots [_V[_V\text{saw}][_N\text{Mary}]_j]]$
- (25) a. TP



- b. $\text{TP} = \{\text{T}, \{\text{John}, \text{T}'\}\}$
 $\text{T}' = \{\text{T}, \{\text{T}, \text{AgrP}_{1_0}\}\}$
 $\text{AgrP}_{1_0} = \{\text{Agr}_0, \{\text{Mary}_j, \text{AgrP}_0\}\}$
 $\text{AgrP}_0 = \{\text{Agr}_0, \{\text{Agr}_0, \text{VP}\}\}$
 $\text{VP} = \{\text{T}_i, \text{V}'\}$
 $\text{V}' = \{\text{V}, \{[_V\text{saw}], [_N\text{Mary}]\}\}$

As we have mentioned before, Move is costly and hence subject to economy principles. The question arises of why

UG takes Move if it is costly. We will discuss Chomsky's answer to this question in section 1.3. In section 1.4 we will discuss how to minimize the cost of Move, once it is required to take place.

1.2.5 Delete/Erase

Although some lexical items exist during derivation, they seem to be invisible at the interface levels. Consider the following sentences:

- (26) a. It seems that John likes Mary.
 b. John likes Mary.
 c. Who does John like?

If we represent (26.a) for semantic interpretation, it would be seem(like(John,Mary)). The expletive it in (26.a) does not affect semantic interpretation, although it exists in syntax. The Agr feature of the verb likes (i.e. 3rd person and singular) also seems to be invisible to semantic interpretation if like(John,Mary) or likes(John,Mary) does not matter for the semantic representation of (26.b). If it is correct that a trace is the copy of an moved element, the trace of who in (26.c) also seems to be deleted at PF, although it is visible during derivation and at LF. In this sense Delete/Erase is conceptually necessary in language.

t

1

i

i

a

a

c

c

t

a

c

c

c

c

c

c

c

c

c

c

c

c

c

c

c

c

c

c

The Delete and Erase operations are to make invisible the elements that are uninterpretable at the interface levels in order to satisfy the output conditions. Delete leaves the structure unaffected but marks some elements as invisible at the interfaces. Although the deleted elements are invisible to the interface levels, they are still accessible to the computational system, and further computations can manipulate them. On the other hand, the operation Erase marks the elements as completely invisible to the interface levels and the computational system at all, and the computation cannot further access them.

Delete/Erase have some empirical and theoretical consequences along with the copy theory of a trace. Consider the following sentence:

(27) (= (41) Chomsky 1995 p.206)

John wondered [_{CP}[which picture of himself] [_{TP}Bill took t]]

Following Chomsky (1993, 1995), sentence (27) is ambiguous in two respects: one is that the reflexive himself can take either John or Bill as its antecedent; and the other is that the phrase take picture can be interpreted either idiomatically ("photograph") or literally ("take it away"). If himself takes Bill as its antecedent, the idiomatic and literal interpretations are permitted; if

himself takes John as its antecedent, however, only the literal interpretation is permitted, but the idiomatic interpretation is disallowed.

To explain the correlation between reflexive binding and idiom interpretation, Chomsky (1993, 1995) argues that (27) has two LF representations as in (28):

- (28) a. John wondered [_{CP}[which x, x a picture of himself] [_{TP}Bill took x]]
 b. John wondered [_{CP}[which x] [_{TP}Bill took [x picture of himself]]]

In the LF representation (28.a), John, not Bill, can be the antecedent of the reflexive himself by condition (a) of the binding theory, and in the representation (28.b), Bill, not John, can be the antecedent of himself by the same principle.

In addition, Chomsky assumes that an idiom should be present as a unit at LF to undergo idiom interpretation. In the configuration of (28.b) the phrase take a picture can be either literally or idiomatically interpreted, since it is present as a unit at LF, but in (28.a) we have only the literal interpretation of take.

To derive the LF representations in (28), Chomsky claims that the Move of which picture of himself leaves a copy of itself as a trace, as shown in (29). After Spell-

out the computational system deletes part of either the higher or the lower copy of the chain, generating (28) at LF.

(29) John wondered [_{CP}[which picture of himself [_{TP}Bill took [which picture of himself]]]]

In addition, the Delete/Erase operation plays an important role with feature checking in the minimalist framework. We will discuss this in the next section.

1.3 Bare Output Conditions and the Computations

Given an array of lexical items, the computational system starts to construct a derivation. At some point of the derivation Spell-out splits this derivation into a pair of linguistic expressions $\langle P, L \rangle$. P consists of the PF objects, and L consists of the LF objects.

The objects of a derivation should be legitimate objects at the relevant interface level. That is, the PF objects should be interpretable at PF, and the LF objects should be interpretable at LF. If P contains only legitimate objects which are interpretable at PF, P is said to converge at PF; otherwise, it crashes at PF. If L contains only legitimate objects which are interpretable at LF L is said to converge at LF; otherwise, it crashes at LF.

A derivation should thus converge at both the interface levels, PF and LF; otherwise it crashes.

In this section let us consider what "interpretable" means, which elements are interpretable at which interface level, and how the interpretability at the interface levels affects the computations.

1.3.1 Features and Their Interpretability

A lexical item is supposed to be F , a set of features. Selecting a LI indicates that F of that LI is selected. For the features F of LI, Chomsky (1995) distinguishes three types of features: phonological features, semantic features, and formal features. The phonological features are interpretable only at PF, and formal and semantic features are interpretable only at LF, and not vice versa. The set of features interpretable at PF are represented as $PF(LI)$, and those interpretable at LF are represented as $LF(LI)$. Of $LF(LI)$, the formal features $FF(LI)$, not semantic features, are accessible to the computational system, too, and play a crucial role in the minimalist program. $FF(LI)$ contains categorial features such as N, A, V, P, D, Etc., Case features such as Nominative, Accusative, Etc., tense features such as Present and Past, agreement features such as number, gender and person, and presumably other features for binders, controllers, and operators.

Chomsky (1995) argues that some of FF(LI) are interpretable at LF, and some others are not, although all FF(LI) are accessible to the computational system. He descriptively classifies FF(LI) as [+ Interpretable] and [- Interpretable], as in (30).

(30) a. [+ Interpretable]:

(i) all categorial features: N, A, V, P, D,
etc.

(ii) agreement features of nominals (D and N):
number, gender, and person.

b. [- Interpretable]:

(i) sublabel of the target⁸: strong features,
affixal

(ii) all non-nominal agreement features

(iii) all Case features.

In the subsequent sections we will see how interpretability has effects on the computations.

1.3.2 Spell-out, and PF and LF Branching

In order to converge at the interface levels or satisfy

⁸The features associated with the label is called sublabels. Formally speaking:

(i) (= (30) Chomsky (1995) p.268)

A sublabel of a category K is a feature of $H(K)_{0max}$, where $H(K)_{0max}$ is the zero-level projection of the head $H(K)$ of K.

output conditions for FI, a derivation should contain only the features which are interpretable at the interface levels. If it contains some uninterpretable features, they should be eliminated by some computations. Otherwise it would crash. For example, phonological features must be eliminated at some point of mapping a numeration to LF, since they are not interpretable at LF; likewise, formal and semantic features must be eliminated at some point of mapping a numeration to PF, since they are not interpretable at PF; otherwise the derivations would crash.

For this Chomsky (1995) assumes that there is an operation, Spell-Out. At some point of a derivation, Spell-out applies to the structure S already formed, and strips phonological features away from S, leaving the others behind, which the computational system continues to map to LF. Further, he assumes that Spell-Out maps S to the Morphology component, which maps it to PF, eliminating non-phonological features, i.e. formal and semantic features, with one exception: strong features cannot be eliminated by PF computation at all. Hence a strong feature must be eliminated before Spell-out.

Regarding the LF mapping, after Spell-out, the derivation now contains only formal features FF(LI) and semantic features, PF(LI) having been eliminated. The computational system continues to map this derivation to LF. Yet the derivation may contain some FF(LI) which are

uninterpretable at LF. If they are not eliminated by the LF computation, the derivation will crash at LF. The elimination of the uninterpretable features, $FF(LI)$, is closely related to feature checking. In next section we will discuss feature checking and Move-F.

1.3.3 Feature Checking and Move-F

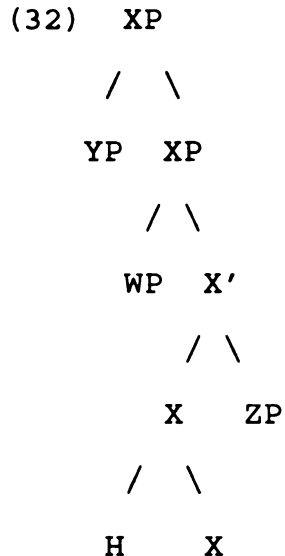
Languages require some formal features of one category to agree with those of other categories. For example, the agreement features of the subject should match with those of a verb of the predicate; the Case feature of the object should match with that of the verb. This feature matching mechanism may be called feature checking.

For feature checking to be successful, it should satisfy two conditions: First, feature checking should happen in some local relation between a checker and a checkee. For example, in (17) the Agr feature of the embedded subject John cannot be checked by that of the matrix verb believes.

(31) *They believes that John kissed Mary.

Chomsky (1995) assumes that a feature checker should be a head or an adjunction to a head, and a checkee should be in a spec of a checker, an adjunction to the maximal

projection of a checker, or an adjunction to a head of a checker. This can be exemplified in (32).



In (32) a head H is adjoined to a head X; ZP is the complement of two segments <X,X>; WP is the specifier of a head X; and YP is an adjunction to the maximal projection XP. In this configuration, the head X is a checker; YP and WP are checkees of X; H can be a checkee of X but can also be a checker of YP and WP. But a category ZP cannot be in checking relations with X and H at all.

The second condition for feature checking is that a formal features should commonly exist in FF(LI) of a checker and its checkee. In addition, once they are checked, all the features common to a checker and a checkee should be checked. Suppose that X has a Case feature in (32). In order for X to be in a checking relation with WP, WP should

also have a Case feature. If WP and X have more common features, they should be also checked when their Case features are checked.

Chomsky (1995) assumes that feature checking is a Delete/Erase operation. That is, if a feature is checked and is uninterpretable, it is deleted, and further erased if possible for convergence at the relevant interface levels.

Let us take (33) into consideration for the checking theory.

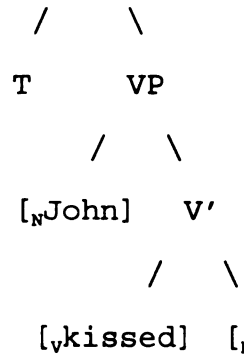
(33) John kissed Mary.

First, let us suppose that (34.a) has been derived by the computations. The head T has a strong D feature which is uninterpretable at PF and at LF, and so it must be presumably eliminated at this point. But T does not have a checkee yet. Move raises John, targeting T, and derives (34.b). At this point John is in the checking domain of T, and hence feature checking is possible if John and T have common features. In this case FF(T) has a strong D feature and a Case feature, and FF(John) also has a D feature and a Case feature. Now FF(John) and FF(T) can be in checking relation. Furthermore, the uninterpretable features (i.e. the D and Case feature of FF(T) and the Case feature of FF(John)) are deleted, once they are checked. However, the D feature of FF(John) remains undeleted because it is an

interpretable feature. The result can be represented in (34.c). Now this derivation can converge at PF, and hence spells out. But this derivation cannot converge at LF yet, since it contains some uninterpretable features, i.e. the Case, Agr and Past feature of FF(kissed) and the Case feature of FF(Mary). To eliminate those features, first, at LF Move raises kissed, targeting T, and derives (34.d). Now FF(kissed) can be a checker of FF(John), and at the same time a checkee of FF(T). The Past feature of FF(kissed) is in the checking relation with that of FF(T), and is deleted. The Agr feature of FF(kissed) is in the checking relation with that of FF(John), and is deleted. The derivation can be represented as in (34.e). After that, FF(Mary) moves to T at LF, deriving (34.f). After the Case feature of FF(kissed) is checked with FF(Mary), the final derivation, (34.g), converges at LF.

(34) (a symbol ** indicates a strong feature, and * a non-strong uninterpretable feature.)

a. TP



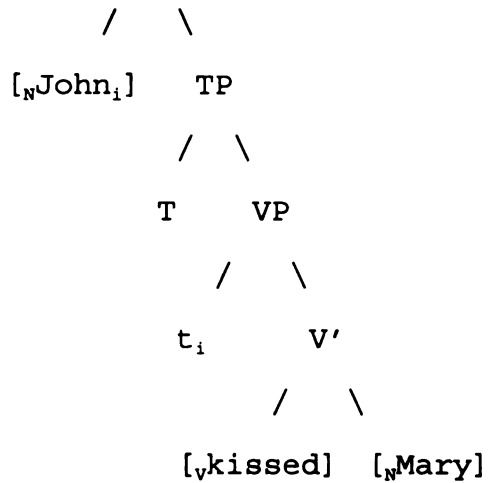
FF(T) = {**D, *Nom, Past}

FF(John) = {D, *Nom, Agr}

FF(kissed) = {V, *Past, *Acc, *Agr}

FF(Mary) = {D, *Acc, Agr}

b. TP1



FF(T) = {**D, *Nom, Past}

FF(John) = {D, *Nom, Agr}

FF(kissed) = {V, Past, *Acc, *Agr}

FF(Mary) = {D, *Acc, Agr}

c. TP1

/ \

[_NJohn_i] TP

/ \

T VP

/ \

t_i V'

/ \

[_vkissed] [_NMary]

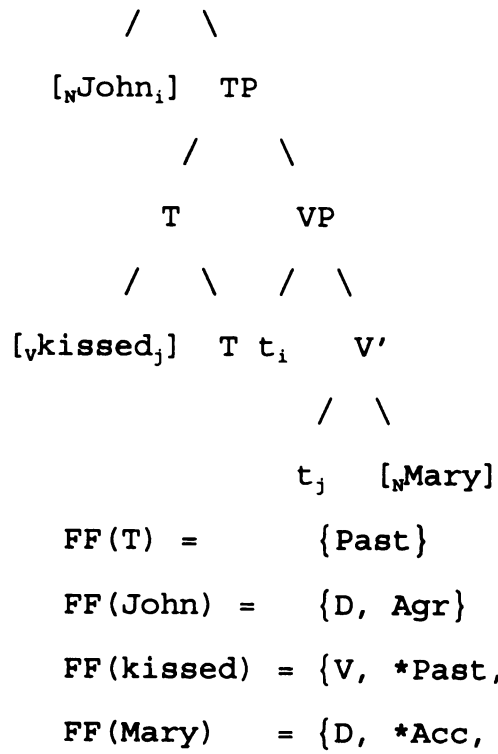
FF(T) = {Past}

FF(John) = {D, Agr}

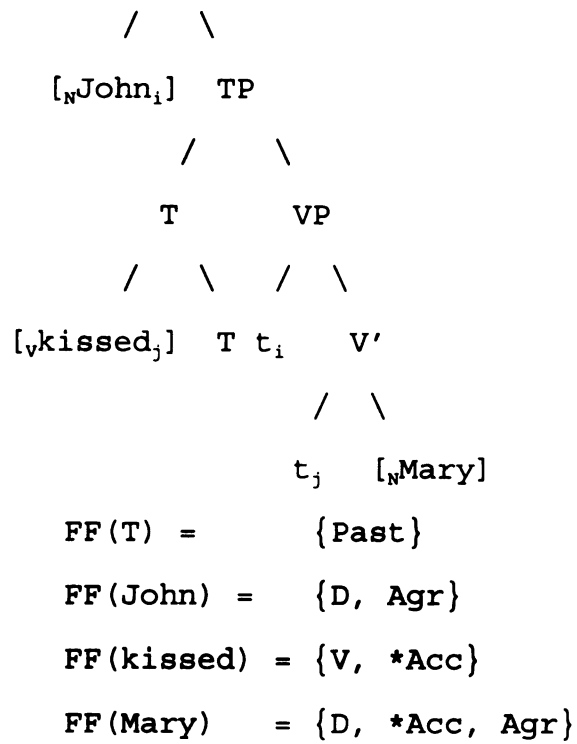
FF(kissed) = {V, *Past, *Acc, *Agr}

FF(Mary) = {D, *Acc, Agr}

d. TP1



e. TP1



f. TP1

/ \

[_NJohn_i] TP

/ \

T VP

/ \ / \

[_NMary_k] T t_i V'

/ \ / \

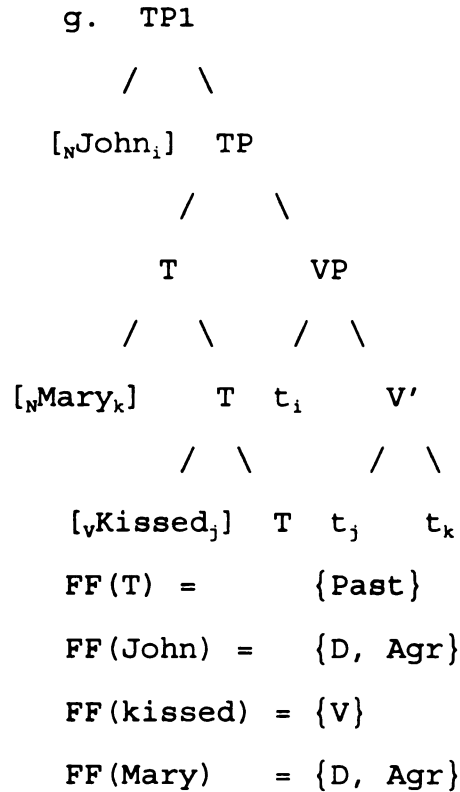
[_vKissed_j] T t_j t_k

FF(T) = {Past}

FF(John) = {D, Agr}

FF(kissed) = {V, *Acc}

FF(Mary) = {D, *Acc, Agr}



Since Move is assumed to be driven only by feature checking, Chomsky (1995) proposes that the minimal operation of Move, then, should move only the feature F to be checked. Move should raise FF(LI) to its target if possible rather than a LI itself. This operation may be called Move-F, which replaces Move- α which raises the whole LI itself. Move-F can be defined as follows:

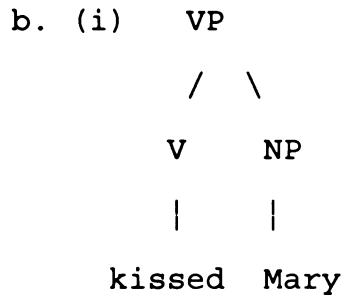
(35) (= (28) Chomsky 1995) p.265)

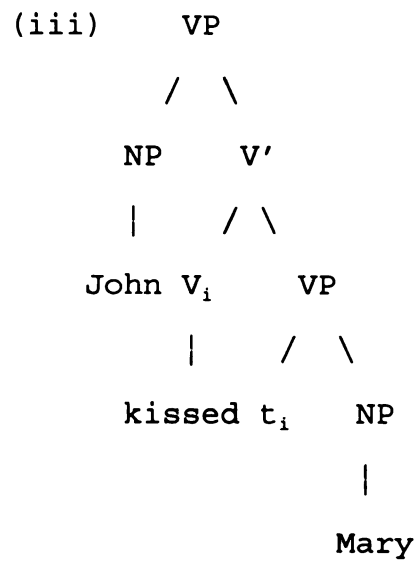
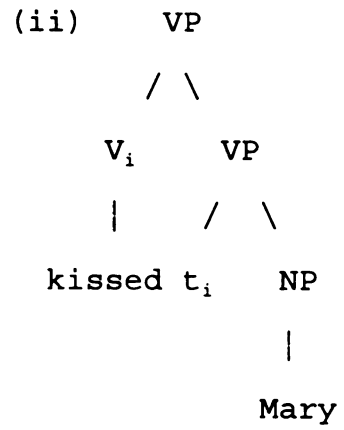
Move-F carries along FF(F), where F is a feature of a lexical item LI, and FF(F) indicates all formal features of LI.

Chomsky (1995) argues that if Move-F raises the formal features of LI overtly, PF convergence requires F to carry along with the whole LI. If Move-F raises F covertly, only FF(F) is raised to a target, leaving its LI behind. Whether FF(F) carries the whole LI is determined presumably by morphological properties, output conditions, and economy principles. Further, covert feature raising adjoins FF(F) to the head of the target, although overt FF(F) raising should target an XP or X of the checker, depending upon the status of the category to be checked.

For example, to derive (36.a) for LF, the computational system takes the derivations as shown in (36.b).

(36) a. John kissed Mary.





(iv) TP

/ \

LI (John)_j T'

/ \

T VP

/ \

NP V'

| / \

t_j V_i VP

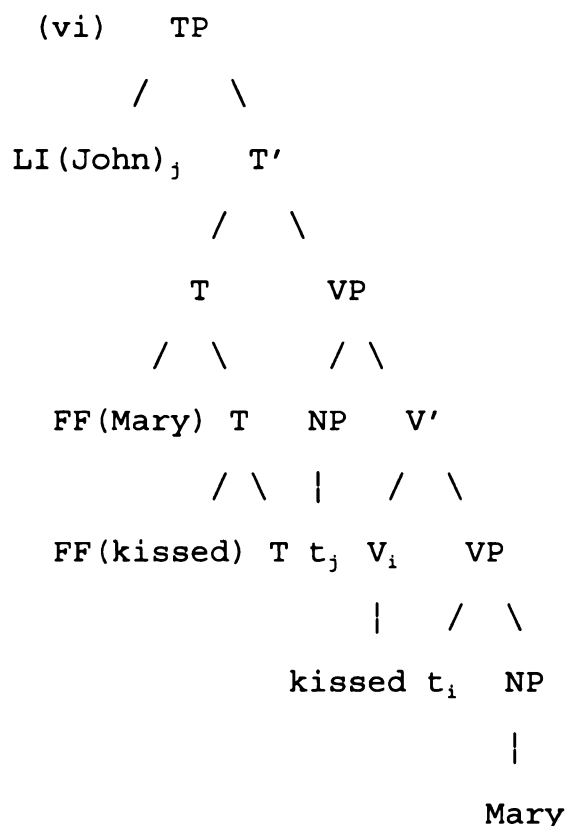
| / \

kissed t_i NP

|

Mary

(v) Spell-out



In the above derivations John is overtly raised to the spec of TP, since an English tense, T, has a strong D feature which is uninterpretable at PF and LF. This overt Move-F carries everything for John, i.e. LI(John). After that, the derivation is spelled out and the covert computations continue to map it to LF. At LF, Move-F raises V, i.e. FF(kissed), to T by adjunction, in order for the tense feature of V to be checked by T, and for the Agr feature of V to check that of John. Then Move-F raises the object, i.e. FF(Mary), to T by adjunction for Case and Agr feature of the object to be checked by those of V.

As we have mentioned in section 1.2, Move is costly,

and thus is subject to economy conditions.

In next section let us consider the relationship between Move and economy conditions.

1.4 The Economy Principles

In the minimalist program a derivation must satisfy bare output conditions for convergence. But satisfying output conditions is necessary but not sufficient for it to be evaluated as syntactically well-formed. It must also be optimal. For a derivation to be optimal, according to Chomsky (1993, 1994, 1995), it must satisfy some economy conditions: the Last Resort Condition, the Minimal Link Condition, and the Procrastinate Principle. In this section let us consider these three economy conditions.

1.4.1 Last Resort

It has long been recognized that an operation Move is a last resort operation: it takes place only when it is forced by some necessity, i.e. to satisfy some conditions. In sentence (37), for example, John moves to the spec of TP; it must take place as a last resort to satisfy the Extended Projection Principle, the Case theory, and presumably other conditions; (37) would be otherwise ungrammatical.

(37) [_{TP}John_i [_{VP}t_i [_V saw Mary]]]

Under the economy considerations, it is natural that a costly computational operation must be driven by some necessity, i.e. to satisfy bare output conditions for FI; otherwise a derivation would fail to converge. In this sense a last resort condition has been understood as an economy condition: the less number of costly operations a derivation takes, the more economical it is. (Chomsky (1993, 1994, 1995), Chomsky and Lasnik (1993))

In the minimalist program, satisfying bare output conditions by movement means eliminating uninterpretable morphological features in checking. So Chomsky (1995) defines the last resort condition for movement in terms of Move-F, as in (38).

(38) (= (51) Chomsky 1995 P.280)

Move-F raises F to target K only if F enters into a checking relation with a sublabel of K.

Now consider (37) again under the definition of (38). Targeting T, Move-F raises the D feature of FF(John) which carries along FF(John). It observes (38), since the D feature of FF(John) that Move-F raises is in the checking

relation with the strong D feature of FF(T)⁹.

Now suppose that Move-F raises FF(John) to target C as in (39). However, no feature of FF(John) is in the checking relation with FF(C). So this raising is superfluous and so violates (38).

(39) *_{CP}John_i [_{TP} t_i [_{VP} t_i [_V saw Mary]]].

Last Resort as defined in (38) can also permit (40) accidentally. Move-F raises the D feature of FF(John) in the spec of the embedded TP to target the matrix T whose FF(T) contains a strong D feature. It observes (38), since the D feature of FF(John) is in the checking relation with the strong D feature of FF(T). But (40) is ungrammatical, not because it violates an economy condition but because it still contains uninterpretable features. That is, although the D feature of FF(John) is in the checking relation with the strong D feature of FF(T) in the matrix sentence, the Case feature of FF(T) cannot be in the checking relation with the Case feature of FF(John), since the Case feature of FF(John) is not available in the spec of the matrix TP: it has been checked and deleted by the Case feature of FF(T) in the embedded sentence.

⁹When checked, the Case feature, a "free rider" of FF(John) is also checked by the Case feature of FF(T).

(40) * $[_{TP}John_i \text{ seems } [_{TP}t'_i [_{VP}t_i [_V \text{ saw mary}]]]]$.

Thus Last Resort in (38) and feature interpretability can successfully block superfluous movement.

1.4.2 Minimal Link Condition

While Last Resort determines whether movement is necessary or not, the Minimal Link Condition (MLC) determines which one should move if more than one category can satisfy Last Resort at the same time.

For example, (41.a) and (41.b) both satisfy FI and Last Resort. However, (41.b) seems to violate some other condition on derivations, while (41.a) satisfies it.

- (41) a. Who did you tell t that John met who?
 b. *who did you tell who that John met t ?

Chomsky (1995) attributes the ungrammaticality of (41.b) to a violation of the MLC. The MLC is defined as follows:

(42) (cf. (110) Chomsky 1995 P.311)

Move-F raises F of x to target K only if there is no y , y closer to K than x , such that y raises to K .

(43) (Chomsky 1995 P.358)

y is closer to the target K than x if y c-commands
x.

To target the matrix CP in (41), two WH-phrases, the complement of tell and that of met, are the competing candidates for Move, since both can satisfy the last resort. But the former is closer to the CP than the latter. Thus (41.b) violates the MLC.

Under economy considerations, the MLC is also an economy condition in terms of the shortest movement: the shortest movement makes the shortest chain links.

1.4.3 Procrastinate Principle

It is well-known that French main verbs are overtly raised to T, and that English ones do not. (Emonds (1978), Pollock (1989), Chomsky (1991))

(44) Jean embrasse souvent Marie.

John kisses often Mary

"John often kisses Mary."

(45) John often kisses Mary.

Following Chomsky (1993), even English main verbs must be raised to T; otherwise the tense feature and Agr feature

of FF(V) would cause (45) to crash at LF, since they are uninterpretable at LF. But it cannot move overtly, as shown in (46), although it observes the Last Resort Condition and the Minimal Link Condition.

(46) *John kisses often Mary.

Chomsky claims that (46) violates an economy condition--the Procrastinate Principle, which is defined in (47).

(47) Minimize overt Move-F.

Under economy considerations, this principle assumes that overt operations cost more than covert operations.

2. Localizing Derivational Economy

2.1 Introduction

In the framework of the minimalist program, all syntactic operations must uniformly satisfy economy conditions. Under the economy considerations syntactic derivations must be optimal. In order to be optimal, a derivation must observe three types of derivational economy conditions (Chomsky 1993, 1994, 1995):

- (1) a. Minimize computational operations.
- b. Minimize chain links.
- c. Minimize overt operations.

Condition (1.a) is the property of greed/last resort of movement; (1.b) adopts the characteristics of Chomsky's (1973) superiority effects and Rizzi's (1990) relativized minimality effects--what may be called the Minimal Link Condition or Shortest Move, and (1.c) is the timing principle of movement--the Procrastinate Principle. The first two conditions are related to the matters of whether to move or not, and which element to move, and the third condition is related to when to move.

In this chapter we will fully consider where the derivational economy conditions in (1) should apply for an optimal derivation. Most of the recent studies of economy principles assume that derivational economy should apply at the interface levels. This may be called global economy. In this chapter we instead propose local economy under which derivational economy should apply locally at each point of a derivation. First of all, section 2.2 discusses the motivation of global economy and its problems in comparison with local economy. Section 2.3 eliminates some assumptions such as (2) and (4) which motivate global economy, and instead makes the following proposals as in (3) and (5) for local economy.

- (2) The more operations a derivation takes, the more costly it is.
- (3) The more superfluous operation a derivation takes, the more costly it is.
- (4) Merge is costfree, and Move is costly.
- (5) Merge and Move are both equal in cost.

Furthermore, we propose the Earliness Principle as a local economy condition on derivations, as stated in (6), and attempts to replace with it the Procrastinate Principle which is a global economy condition in nature.

(6) The Earliness Principle

Satisfy bare output conditions as early as possible.

2.2 Global Economy: the Motivations and Problems

2.2.1 A Distinction between Global and Local Economy

Chomsky (1995 pp.220-221) proposes that economy conditions must hold only of convergent derivations. In other words, the computational system generates three relevant sets of derivations at an interface level: D , D_c , and D_A . The set D is the set of all the possible derivations that the computational system can generate, regardless of whether or not they converge at that interface. The set D_c is the set of convergent derivations among the set of derivations in D which satisfy the interface conditions for Full Interpretation. So D_c is a subset of D . The set D_A is the set of admissible derivations among the set of convergent derivations D_c which satisfy the economy conditions. Thus D_A is a subset of D_c . It indicates that the economy conditions apply at the interface levels to select optimal derivations. This may be called a global economy condition.

We distinguish global economy from local economy as in

the following:¹⁰

(7) Global Economy

Derivational economy should apply at the interface levels or representations so that it selects a derivation (among convergent derivations) that takes the most economical operations.

(8) Local Economy

Derivational economy should apply at each point of a derivation so that it selects the most economical operation to affect the target at that point.

While local economy evaluates the optimality of derivations locally during the course of derivation, global economy applies at the interface levels, and selects a set of optimal derivations, examining the derivational history of convergent derivations.

In general, the following assumptions for measuring the optimal operations motivate global derivational economy.

- (9) a. The fewer operations a derivation takes, the more economical it is.
- b. Merge is more economical than Move.
- c. Covert operations are more economical than overt

¹⁰See also for definitions of local economy: Collins (1995) and Ura (1995).

operations.¹¹

In the subsequent sections we will consider the assumptions in (9) in detail.

2.2.2 The Last Resort Condition and Global Economy

The operation Move has long been assumed to be a last resort operation in language. It should be driven only by some (morphological) necessity. For example, in (10) John moves to the spec of TP; it must take place to check the strong D feature of T, and the Case features of T and John; otherwise (10) would crash.

(10) [_{TP}John_i [_{VP}t_i [_V saw Mary]]]

On the other hand, in (11) it is unnecessary for John to move to CP, since (11) can converge without that movement. In this sense raising John in (11) violates the last resort for movement.

(11) * [_{CP}John_i [_{TP}t'_i [_{VP}t_i [_V saw Mary]]]]

¹¹The minimalist program also assumes (i) for measuring an optimal derivation, but this assumption can apply locally without any further modification.

(i) The shorter movement a derivation takes, the more economical it is.

Chomsky (1993, 1994, 1995) derives this last resort condition from the assumption that Move is a costly operation, and that the computational system must minimize the Move operation as much as possible. Under economy considerations, this can be stated as in (9.a), repeated in (12):

- (12) The fewer operations a derivation takes, the more economical it is.

If we compare (10) with (11) in terms of the movement of John under the assumption, (12), the former takes only one movement operation, while the latter takes two movement operations. Hence (10) blocks (11).

As Chomsky (1995) mentions:

a derivation in which an operation applies is less economical than one that differs only in that the operation does not apply. The most economical derivation, then, applies no operations at all to a collection of lexical choices and thus is sure to crash. If nonconvergent derivations can block others, this derivation will block all others... (pp.220-221)

however, economy should apply at the interface levels under the assumption, (12); otherwise nonconvergent derivations would be always optimal.

Consider (13) for this.

- (13) a. *_{[CP}[_{TP}seems [_{TP}to be likely [_{TP}to [_{VP}John win]]]]].

- b. [_{CP}[_{TP}John_i seems [_{TP}t''_i to be likely [_{TP}t'_i to
[_{VP}t_i win]]]]].

If we compare (13.a) with (13.b) in terms of NP movement under the assumption of (12), the derivation of (13.a) is optimal, since (13.a) takes no movement at all but (13.b) takes three applications of Move. If (12) applies during the course of derivation or applies to all the possible derivations at the interface levels, a nonconvergent derivation would thus block other derivations, and UG would never generate a convergent derivation at all.

If (12) applies only to convergent derivations at the interface levels, however, (13.b) cannot be compared with (13.a), and hence becomes optimal, since (13.a) is not a convergent derivation: roughly speaking, the Case features of John and Tense, and the strong D feature of Tense are uninterpretable at the interfaces for FI.

Hence under the assumption of (12), derivational economy should apply only to convergent derivations at the interface levels; otherwise nonconvergent derivations would become always optimal and block convergent ones during derivation.

2.2.3 The Minimal Link Condition and Global Economy

The assumption (9.b), repeated in (14), makes the

Minimal Link Condition (MLC) apply globally at the interface levels, although the MLC itself is applicable as a local economy condition. This is motivated by the assumption that Merge is costfree and Move is costly.

(14) Merge is more economical than Move.

Consider the following superraising case:

(15) *John seems that it is likely t to win.

Suppose that the computational system has constructed (16) for (15). At this point the computational system has two choices: it can take a Merge operation, concatenating it to the TP as in (17.a), or it can take a Move operation, raising John to the TP as in (17.b).

(16) [_{TP}is likely John to win]

(17) a. [_{TP}it [_Tis likely John to win]]

b. [_{TP}John [_Tis likely t to win]]

If Merge is costfree and Move is costly, and derivational economy applies locally at the point given in (16), then derivational economy will pick (17.a) rather than (17.b) for an optimal derivation, since (17.a) takes no costly Moves but (17.b) takes one costly Move. If we

compare Merge with Move locally in terms of cost, we can never get the derivation (18), since it is derived from (17.b) which is blocked by (17.a).

(18) It seems that John_i is likely t_i to win.

If derivational economy applies only to convergent derivations at the interface levels, on the other hand, the assumption (14) will then not allow (17.a) to block (17.b) at the point of (16). Then, the computational system will generate further derivations from both (17.a) and (17.b), as shown in (19). Now (19.c) will be optimal at the interface levels among the derivations in (19). That is, (19.a) crashes because the Case feature of T cannot enter into a checking relation with the Case feature of FF(it) which has already been deleted by checking by the FF(T) in the embedded sentence; on the other hand, (19.b) and (19.c) are equal in terms of the number of operations, since both take one Move operation and one Merge operation, but (19.c) takes shorter movement than (19.b) under the MLC.

- (19) a. *It_i seems that t_i is likely John to win.
 b. *John_i seems that it is likely t_i to win.
 c. It seems that John_i is likely t_i to win.

Although the MLC itself can be formulated in a local

way, we cannot avoid the global application of the MLC under (14).

2.2.4 The Procrastinate Principle and Global Economy

The assumption of (9.c) is a global and also stipulative concept, which we return to in section 3.3 in chapter 3. We repeat (9.c) in (20).

(20) Covert operations are more economical than overt operations.

Let us consider (21) and (22) under the assumption of (20). If (21.a) competes with (22.a) during the derivation, the former is more economical than the latter, and wrongly blocks it, since the former takes no overt movement but the latter takes one overt movement.

- (21) a. [_{TP}t [_{VP}often [_{VP}John left]]]
 b. Spell-out
 c. [_{TP}John_i [_TT [_{VP}often [_{VP}t_i left]]]]
 (22) a. [_{TP}John_i [_TT [_{VP}often [_{VP}t_i left]]]]
 b. Spell-out

If we apply (20) at the interface levels, the derivation (22) will become optimal, since (21) crashes and

cannot be compared with (22).

In next section we will discuss the problems with global economy in detail, and motivate the adequacy of local economy for universal grammar.

2.2.5 Some Problems with Global Economy

First of all, although derivational economy is a condition on derivations, it cannot block nonoptimal derivations during the derivation, as shown in (13). It must wait until the computational system generates a set of all possible derivations, and the interface conditions select a set of convergent derivations from a set of all the possible derivations. After that, the economy conditions will select some optimal derivation from the set of convergent derivations. In this sense it is hard to look at global economy conditions as derivational conditions; rather, they function like conditions on representations.

Second, the global economy in which economy holds only of convergent derivations cannot apply to all economy conditions in a consistent way. For example, it is problematic with the Minimal Link Condition (MLC). Consider (23).

- (23) a. * $[_{TP}John_i \text{ seems that } [_{TP}it \text{ is likely } [_{TP}t_i \text{ to win}]]]$.

- b. *[what_i did John wonder [_{CP}who_j [_{TP}t_j bought t_i]]].

Both (23.a) and (23.b) are cases that violate the MLC. To explain the ungrammaticality of (23), Chomsky (1993, 1994, 1995) proposes the shortest Move or Minimal Link Condition as an economy condition as in (1.b) which is repeated in (24) for convenience. Following the MLC in (24), a shorter movement is more economical than a longer movement, and hence blocks it.

(24) A derivation must minimize chain links.

The ungrammaticality of (23.a) can be explained with the global MLC as follows: The computational system generates a set of all possible derivations at PF and LF. From this set we would presumably get a set of convergent derivations as in (25).

- (25) D_c = {
 (i) It seems that John_i is likely t_i to win.
 (ii) John_i seems that it is likely t_i to win.}

Then the global MLC would select (i) from D_c in (25), since the movement of John in (i) is shorter than the

movement of John in (ii)¹².

Now take the ungrammaticality of (23.b). The computational system generates a set of all possible derivations for it, and selects a set D_c of convergent derivations from this set. But (23.b) is the only convergent derivation at this time. So it must be the optimal derivation because there is no shorter movement than the movement of what among the convergent derivations. Although there is a derivation which takes a shorter movement than (23.b), it cannot block (23.b) if it crashes:

(26) [_{CP}Who_j did John wonder [_{CP}t'_j [_{TP}t_j saw what]]]

The case of (23.b) makes a strong implication that some economy condition like the MLC must be a local condition on derivations and should not be violated even for convergence¹³.

If this is correct, then the global characteristic can apply to some economy conditions like Procrastinate, and

¹²It would be more desirable to compare (25.ii) with (i) as below, since they are derived from the same partial derivation (ii).

(i) *It seems that t_i is likely John to win.

(ii) [_{TP}seems that it is likely John to win]

But (i) cannot be compared with (25.ii) under global economy, because it cannot converge at the interface levels. If we compare the movement of it in (i) with the movement of John in (25.ii), the former would be optimal, because the movement of it is shorter than the movement of John.

¹³Chomsky (1995) also takes this research line in which the MLC must be a local condition and unviolable.

cannot apply to other economy conditions like the MLC. Hence the economy conditions become heterogeneous in the grammar. This heterogeneusness seems to be arbitrary. It would be conceptually simpler to have only local economy conditions.

Third, for global economy the computational system generates a set of all the possible derivations explosively (or exponentially) and redundantly, regardless of whether the derivations are optimal or not. This set should also include the derivational history of each derivation so that economy conditions can examine the history to select the optimal derivation. If economy is a real condition on derivations, it would be better that economy constrains the computational system to generate only a set of optimal derivations, regardless of whether they are convergent or nonconvergent at the interface levels.

Let us take an example for this.

(27) Who left?

Suppose the computational system derives (27). First, it generates a set D of all possible derivations for the interface levels, as in (28). (We here ignore the possibilities of V-to-C-raising.)

(28) D = {

- (i) $[_{VP}who\ left] \rightarrow [_{TP}[_{VP}who\ left]] \rightarrow$
 $[_{CP}[_{TP}[_{VP}who\ left]]]$ (at PF and at LF),
- (ii) $[_{VP}who\ left] \rightarrow [_{TP}who_i\ [_{VP}t_i\ left]] \rightarrow$
 $[_{CP}[_{TP}who_i\ [_{VP}t_i\ left]]]$ (at PF and at LF),
- (iii) $[_{VP}who\ left] \rightarrow [_{TP}who_i\ [_{VP}t_i\ left]] \rightarrow$
 $[_{CP}who_i\ [_{TP}t'_i\ [_{VP}t_i\ left]]]$ (at PF and at LF),
- (iv) $[_{VP}who\ left] \rightarrow [_{TP}\ [_{VP}who\ left]] \rightarrow [_{CP}who_i$
 $[_{TP}t'_i\ [_{VP}t_i\ left]]]$ (at PF and at LF),
- (v) $[_{VP}who\ left] \rightarrow [_{TP}\ [_{VP}who\ left]] \rightarrow$
 $[_{CP}who_i\ [_{TP}\ [_{VP}t_i\ left]]]$ (at PF and at LF),
- (vi) $[_{VP}who\ left] \rightarrow [_{TP}left_i\ [_{VP}who\ t_i]] \rightarrow$
 $[_{CP}[_{TP}left_i\ [_{VP}who\ t_i]]]$ (at PF and at LF),
- (vii) $[_{VP}who\ left] \rightarrow [_{TP}left_i\ [_{VP}who\ t_i]] \rightarrow$
 $[_{TP1}who_j\ [_{TP}left_i\ [_{VP}t_j\ t_i]]] \rightarrow [_{CP}[_{TP1}who_j$
 $[_{TP}left_i\ [_{VP}t_j\ t_i]]]$ (at PF and at LF),
- (viii) $[_{VP}who\ left] \rightarrow [_{TP}left_i\ [_{VP}who\ t_i]] \rightarrow$
 $[_{TP1}who_j\ [_{TP}left_i\ [_{VP}t_j\ t_i]]] \rightarrow [_{CP}who_j$
 $[_{TP1}t'_j\ [_{TP}left_i\ [_{VP}t_j\ t_i]]]$ (at PF and at LF),
- (ix) $[_{VP}who\ left] \rightarrow [_{TP}left_i\ [_{VP}who\ t_i]] \rightarrow$
 $[_{CP}who_j\ [_{TP1}t'_j\ [_{TP}left_i\ [_{VP}t_j\ t_i]]]]$ (at PF and at LF),
- (x) $[_{VP}who\ left] \rightarrow [_{TP}left_i\ [_{VP}who\ t_i]] \rightarrow$
 $[_{CP}who_j\ [_{TP}left_i\ [_{VP}t_j\ t_i]]]$ (at PF and at LF),

- (xi) $[_{VP}who\ left] \rightarrow [_{TP}[_{VP}who\ left]] \rightarrow$
 $[_{CP}[_{TP}[_{VP}who\ left]]]$ (at PF) $\rightarrow [_{VP}who\ left]$
 $\rightarrow [_{TP}left_i\ [_{VP}who\ t_i]] \rightarrow [_{CP}[_{TP}left_i\ [_{VP}who\ t_i]]]$ (at LF),
- (xii) $[_{VP}who\ left] \rightarrow [_{TP}who_i\ [_{VP}t_i\ left]] \rightarrow$
 $[_{CP}[_{TP}who_i\ [_{VP}t_i\ left]]]$ (at PF) $\rightarrow [_{VP}who\ left]$
 $\rightarrow [_{TP}who_i\ left_j\ [_{VP}t_i\ t_j]] \rightarrow$
 $[_{CP}[_{TP}who_i\ left_j\ [_{VP}t_i\ t_j]]]$ (at LF),
- (xiii) $[_{VP}who\ left] \rightarrow [_{TP}who_i\ [_{VP}t_i\ left]] \rightarrow$
 $[_{CP}who_i\ [_{TP}t'_i\ [_{VP}t_i\ left]]]$ (at PF) $\rightarrow$
 $[_{VP}who\ left] \rightarrow [_{TP}who_i\ left_j\ [_{VP}t_i\ t_j]] \rightarrow$
 $[_{CP}who_i\ [_{TP}t'_i\ left_j\ [_{VP}t_i\ t_j]]]$ (at LF),
- (xiv) $[_{VP}who\ left] \rightarrow [_{TP}\ [_{VP}who\ left]] \rightarrow [_{CP}who_i\$
 $[_{TP}t'_i\ [_{VP}t_i\ left]]]$ (at PF) $\rightarrow [_{VP}who\ left]$
 $\rightarrow [_{TP}left_j\ [_{VP}who\ t_j]] \rightarrow [_{CP}who_i\ [_{TP}t'_i\$
 $left_j\ [_{VP}t_i\ t_j]]]$ (at LF),
- (xv) $[_{VP}who\ left] \rightarrow [_{TP}\ [_{VP}who\ left]] \rightarrow [_{CP}who_i\$
 $[_{TP}\ [_{VP}t_i\ left]]]$ (at PF) $\rightarrow [_{VP}who\ left] \rightarrow$
 $[_{TP}left_j\ [_{VP}who\ t_j]] \rightarrow [_{CP}who_i\ [_{TP}left_j\ [_{VP}t_i\$
 $t_j]]]$ (at LF),
- ...
- ... }

Then the interface conditions select a set D_c of convergent derivations from (28): PF selects convergent derivations as in (29), and LF does as in (30).

$$(29) D_c = \{(iii), (iv), (viii), (ix), (xiii), (xiv)\}$$

$$(30) D_c = \{(viii), (ix), (xiii), (xiv)\}$$

Now we get a set D_c of derivations which converge both at PF and at LF as in (31).

$$(31) D_c = \{(viii), (ix), (xiii), (xiv)\}$$

Then the global economy conditions will select a set D_A of optimal derivations from (31). Procrastinate selects D_c as in (32) from (31).

$$(32) D_A = \{(xiii), (xiv)\}^{14}$$

As we have seen above, like a condition on representations, global economy must allow the computational system to generate an exponential number of derivations to get a set of optimal derivations.

Fourth, as we pointed out in the previous section, some derivations which may be optimal during the time of derivation would become nonoptimal ones at the interface levels if they cannot converge, although this does not occur with local economy.

If we can apply some derivational economy conditions

¹⁴Earliness will further select (xiii) for an optimal derivation in comparison with (xiv). See section 2.3.3 and chapter 3.

in a local way during the time of derivation we can get only optimal derivations to reach the interface levels. For example, suppose (33.a) is derived, and more economical than (33.b).

- (33) a. [_{TP}T [_{VP}who left]]
 b. [_{TP}left_i+T [_{VP}who t_i]]

Then (33.a) will block all derivations in which (33.b) is involved (i.e. (vi)-(x) in (28).)

Next, (34.a) is supposed to be further constructed and optimal. Then it will block all derivations in which (34.b) is involved (i.e. (i), (iv), (v), (xi), (xiv), (xv) in (28)).

- (34) a. [_{TP1}who_i T [_{VP}t_i left]]
 b. [_{CP}C [_{TP}T [_{VP}who left]]]

Next (35.a) is constructed and optimal at this point. Then it will block all derivations containing (35.b) (i.e. (ii), (xii) in (28)).

- (35) a. [_{CP}who_i [_{TP1}t_i T [_{VP}t_i left]]]
 b. [_{CP}[_{TP1}who_i T [_{VP}t_i left]]]

Next, suppose that (36.a) is constructed and optimal.

It will block (36.b).

- (36) a. [_{CP}who_i [_{TP1}t_i left_j+T [_{VP}t_i t_j]]]
 b. [_{CP}who_i [_{TP1}t_i T [_{VP}t_i left]]]

Now the derivation (36.a) is optimal and gets to the interface levels for FI. Bare output conditions can interpret it; hence it converges.

Thus it is more relevant and desirable if economy is a local condition on derivations, and determines the optimality of derivations during the time of derivation and generates only a set of optimal derivations at the interfaces. In next section let us consider local derivational economy in detail.

2.3 Localizing Derivational Economy

As we have mentioned in section 2.2, local economy can be stated as follows:

(37) Local Economy

Derivational economy should apply at each point of a derivation so that it selects the most economical operation to affect the target at that point.

That is, local economy evaluates the optimality of derivations at each point of derivation, and selects the most economical operation at a given point. Thus local economy generates only a set of optimal derivations at the interface levels rather than three sets of derivations, i.e. a set of all possible derivations, a set of convergent derivations, and a set of admissible derivations, which global economy requires at the interface levels.

To pursue local derivational economy, in this section we reconsider the concepts of measuring the cost of operations, as in (9), and remedy the concepts that motivate global economy with alternative local measurements for the optimality of derivations. (9) is repeated in (38) for convenience.

- (38) a. The fewer operations a derivation takes, the more economical it is.
- b. Merge is more economical than Move.
- c. Covert operations are more economical than overt operations.

2.3.1 Localizing the Last Resort Condition

As shown in section 2.2.2, the assumption (9.a), repeated in (39), requires the economy condition to apply at the interface levels; otherwise "no operation" would block

even last resort operations.

- (39) The fewer operations a derivation takes, the more economical it is.

But we should not understand the concept of last resort in such a way that operations are always costly, and therefore that "no operation" is always the most economical. To measure the most economical operation, we should consider the necessity of the operation. In other words, "no operation" should not block a last resort operation under economy considerations. If our assumption is correct, we eliminate (39) and replace it with (40)¹⁵.

- (40) The more superfluous operations a derivation takes, the more costly it is.

The distinction between "no operation" and "no superfluous operation" for measuring the cost of an operation has a desirable consequence for local economy.

Consider (41):

- (41) a. *_{TP}Seems [_{TP}John to leave]].
 b. [_{TP}John_i seems [_{TP}t_i to leave]].

¹⁵Chomsky attempts to derive (40) from (39), applying economy globally at the interface levels.

If we apply (39) to (41.a) and (41.b) during the course of the derivation, then (41.a) will block (41.b): (41.a) takes no movement at all, while (41.b) takes one movement. We will get a undesirable result.

If we apply (39) to (41) at the interface levels, then it will correctly select (41.b), since (41.a) cannot converge at the interfaces. So (39) forces economy to apply only to a set of convergent derivations at the interface levels.

If we take (40) for derivational economy, then the prediction will be different. If we compare (41.a) with (41.b) in terms of (40) during the time of derivation, then (41.a) and (41.b) are both optimal, since both of them take no superfluous movement at all. That is, the movement in (41.b) is not superfluous but necessary movement. So (41.a) and (41.b) cannot block each other in terms of the number of computational operations.

More strongly speaking, we argue in section 2.3.3 that Earliness will select (41.b) for an optimal derivation in comparison with (41.a), although (40) considers (41.a) and (41.b) to be equally optimal. (See section 2.3.3 for more detail.)

For local economy, we adopt Chomsky's (1995) formulation of the Last Resort Condition on movement as in (42).

(42) (cf. (51) Chomsky 1995 P.280)

K attracts F only if F enters into a checking relation with a sublabel of K.

This formulation of last resort can only tell us that movement is legitimate, i.e. whether movement can take place or not. But this will be strengthened by the Earliness Principle in section 2.3.3 so that a last resort operation should be triggered as early as possible.

2.3.2 Localizing the Minimal Link Condition

The assumption of (40) naturally leads us to another way to eliminate one more concept of global economy as in (9.b). We repeat (40) and (9.b) in (43) and (44), respectively:

(43) The fewer operations a derivation takes, the more economical it is.

(44) Merge is more economical than Move.

Chomsky (1995) assumes that Merge is a cost-free operation, and Move is a costly operation¹⁶. From our assumption of (43), however, we can draw the conclusion that

¹⁶I cannot see any specific motivation for such distinction between Merge and Move in terms of cost. It seems to be simply a stipulation.

Move is not necessarily a costly operation. That is, if Move is taken by necessity, it can be thought of as a costfree operation. The same will be true with Merge. If Merge is taken by necessity, it will take no cost at all; otherwise it is also a costly operation.

Thus all syntactic operations, including Merge, Move, and perhaps Delete, are assumed to be last resort operations so that they "must be driven by some condition on representations" to satisfy FI at the interface levels; otherwise they would crash. (Chomsky 1995 p.28) Then, all computations are considered to be equally costly¹⁷. So, which operation, Merge or Move, the computational system will take must completely depend upon some other evidence rather than the economy considerations¹⁸.

If our assumption is correct, we can replace the assumption (44) with (45):

(45) Merge and Move are equal in cost.

Returning to (15)-(17), repeated in (46)-(48), at the point of (47) Merge and Move are both equal in cost, since neither Move nor Merge are superfluous at all. So (48.a)

¹⁷From Watanabe's (1995) Avoid Redefinition, we may have the same conclusion. That is, Merge and cyclic Move do not undergo redefinition, and so are equally economical.

¹⁸Following Collins (1995), Merge is driven by the fact that both of the phrases have the property that they must be integrated into the clause.

and (48.b) are both available to the computational system for further computation.

(46) *John seems that it is likely t to win.

(47) [_{TP}is likely John to win]

(48) a. [_{TP}it [_T'is likely John to win]]

b. [_{TP}John [_T'is likely t to win]]

Now Chomsky's (1995) Minimal Link Condition, as formulated in (49), can apply locally during the course of a derivation.

(49) (cf. (110) Chomsky 1995 p.311)

K attracts F of X only if there is no y, y closer to K than x, such that y raises to K.

(50) (Chomsky 1995 P.358)

y is closer to the target K than x if y c-commands x.

2.3.3 Earliness as a Local Economy Condition

Among the economy principles, only Procrastinate can hardly be maintained for local economy, defined in (9.c) which is repeated in (51). Procrastinate seems to be global in nature.

(51) Covert operations are more economical than overt operations.

To localize all derivational economy conditions uniformly, we attempt to eliminate Procrastinate from derivational economy, and instead we propose an alternative timing principle, the Earliness Principle, which is independently motivated by cyclic computation (which we will discuss in chapter 3.) Putting aside the motivations of Procrastinate and its problems, and a way of reducing Procrastinate to Earliness in chapter 3, let us elaborate the Earliness Principle as a local economy condition to strengthen the Last Resort Condition in this section.

Following Chomsky (1993, 1994, 1995), the computational system constructs a linguistic expression $\langle p, l \rangle$ in an optimal way where p is a PF object and l is an LF object. The concept of optimality can be considered from various points of view¹⁹. The intuitive idea here is that economy is related to how early or fast a derivation can satisfy bare output conditions. This consideration of economy may be called the Earliness Principle. If computation is a process of satisfying bare output conditions at the interface levels, we propose the Earliness Principle as a timing principle and economy condition as in the following:

¹⁹See Chomsky (1993, 1994, 1995), Collins (1994, 1995), Fukui (1993), Kitahara (1994), Oka (1993, 1995), Ura (1995), Watanabe (1995) for different views of economy conditions.

(52) The Earliness Principle

A derivation must satisfy bare output conditions as early as possible.

From our Earliness Principle, (52), we can also derive Pesetsky's (1989) idea of an earliness principle that movement must take place as early in the derivation as possible. An operation Move is motivated by eliminating uninterpretable features to satisfy bare output conditions. In other words, for a derivation to satisfy bare output conditions early, all uninterpretable morphological features in a derivation must be checked as early as possible. So (52) subsumes Pesetsky's (1989) Earliness which can be repeated in (53) in our terms²⁰.

(53) Uninterpretable morphological features must be checked as early as possible.

²⁰We may derive some condition on Merge from our Earliness Principle. Chomsky (1995) claims that Merge should apply only to a root of a phrase structure. For example, the computational system constructs partial phrase structure (i) not as in (ii) but as in (iii):

- (i) [_{TP}T [_{VP}John [_v.met Mary]]]
- (ii) a. [_{VP}met Mary]
- b. [_{TP}T [_{VP}met Mary]]
- c. [_{TP}T [_{VP}John [_v.met Mary]]] -> *Merge
- (iii) a. [_{VP}met Mary]
- b. [_{VP}John [_v.met Mary]]
- c. [_{TP}T [_{VP}John [_v.met Mary]]]

If Merge is greedy in some sense (Collins 1995), (iii) satisfies the Greed Principle earlier than (ii). I will leave this question for further research.

Formally speaking, we formulate Earliness in (54), relating Attract-F:

- (54) K attracts F early only if a sublabel of K is an uninterpretable feature at the interface level that Attract-F affects.

Let us consider how Earliness can help a last resort operation to block "no operation", as discussed in section 2.3.1. Consider the embedded clausal construction in (55):

- (55) A professor knows [_{CP}whether C [_{TP}[_{DPA} student]_i should [_{VP}t_i read this book]]]

Suppose that the D feature of English tense T is strong. To satisfy PF output condition, it must be eliminated by raising a student, targeting TP, for feature checking. The computational system may take either (56) or (57) for (55). Both of them are convergent derivations.

- (56) a. [_Tshould [_{VP}[_{DPA} student] [_Vread this book]]].
 b. [_{TP}[_{DPA} student]_i [_Tshould [_{VP}t_i [_Vread this book]]]]].
 c. [_CC [_{TP}[_{DPA} student]_i [_Tshould [_{VP}t_i [_Vread this book]]]]].
 d. [_{CP}whether [_CC [_{TP}[_{DPA} student]_i [_Tshould [_{VP}t_i

[_v.read this book]]]]].

- (57) a. [_T.should [_{VP}[_{DP}a student] [_v.read this book]]].
 b. [_C.C [_T.should [_{VP}[_{DP}a student] [_v.read this book]]]].
 c. [_{CP}whether [_C.C [_T.should [_{VP}[_{DP}a student] [_v.read this book]]]]].
 d. [_{CP}whether [_C.C [_{TP}[_{DP}a student]_i [_T.should [_{VP}t_i [_v.read this book]]]]]].

Suppose that the computational system has constructed (56.a) (or (57.a)). At this point the computational system can take two choices: the strong D feature of FF(T) attracts a student, as in (56.b), or it can merge C and TP to CP as in (57.b). Derivational economy, Earliness, picks (56.b) for an optimal derivation rather than (57.b), since the strong D of FF(T) is uninterpretable at PF and LF, this Attract-F affects PF and LF at the same time, and (56.b) is the earliest point for Attract-F for TP. Hence a last resort operation becomes optimal in comparison with no movement under Earliness.

As shown above, different from Procrastinate, Earliness selects an optimal derivation in a course of derivation rather than it selects an optimal one among a set of derivations at the interface levels. That is, the derivation (56) blocks (57) at the point of derivation (56.b).

2.4 Summary

In this chapter we have distinguished local economy from global economy as in (58)-(59), and explored three types of local derivational economy conditions: the Last Resort Condition, the Minimal Link Condition, and the Earliness Principle, as defined in (60)-(62), respectively.

(58) Global Economy

Derivational economy should apply at the interface levels or representations so that it selects a derivation (among convergent derivations) that takes the most economical operations.

(59) Local Economy

Derivational economy should apply at each point of derivation so that it selects the most economical operation to affect the target at that point.

(60) The Last Resort Condition (= (51) Chomsky 1995 P.280)

K attracts F only if F can enter into a checking relation with a sublabel of K.

(61) The Minimal Link Condition (cf. (110) Chomsky 1995 P.311)

K attracts F of x only if there is no y, y closer to K than x, such that K attracts y.

(62) The Earliness Principle

K attracts F early only if a sublabel of K is an uninterpretable feature at the interface that Attract-F affects.

Now at each point of a derivation the computational system takes the most economical operation which satisfies all the three types of derivational economy conditions given in (60)-(62), constructing a linguistic expression $\langle P, L \rangle$ for PF and LF. As a result, it generates only a set of optimal derivations at the interface levels.

local economy has following advantages in comparison with global economy:

	Global	Local
i	It is a kind of representational condition.	It is a strictly derivational condition.
ii	It allows the computational system to generate an explosive or exponential number of derivations at the interface levels.	It allows the computational system to generate only a set of optimal derivations at the interface levels.
iii	Some derivations which were optimal during the time of derivation may become nonoptimal at the interface levels.	Some derivations which are optimal during the time of derivation are always optimal at the interface levels.
iv	It makes economy conditions heterogeneous in terms of unviolability and locality.	It makes economy conditions homogeneous in terms of unviolability and locality.

In addition, local economy offers some unified analyses of various phenomena of natural language. In the subsequent sections we will investigate cyclic computation, Procrastinate effects, some Wh-asymmetries and Wh-adjunct symmetries under local economy.

3. Deriving Strict Cycle and Procrastinate Effects

3.1 Introduction

In this chapter we derive two seemingly opposite principles, the Strict Cycle Condition (SCC), a principle for overt computations, and the Procrastinate Principle, a principle for LF computations, from Earliness.

Linguists have long observed that the computational operations, specially, overt operations, apply cyclically. Sentence (1) is a typical example²¹ which shows that overt

²¹Another typical example that the SCC applies to is the case of Wh-island violations. Consider (i).

(i) *_{CP}[How_i did _{TP1}John wonder _{CP}what_j _{TP2}Bill bought t_j t_i]]]

Sentence (i) is the one that violates Subjacency, since how crosses two bounding nodes, i.e. TP1 and TP2, deriving it as in (ii). If the computational system were to derive (i) as in (iii), it would escape the Subjacency violation.

(ii) a. [_{TP2}Bill bought what how]
 b. [_{CP}what_i [_{TP2}Bill bought t_i how]]
 c. [_{TP1}John did wonder [_{CP}what_i [_{TP2}Bill bought t_i how]]]
 d. [_{CP}how_j did [_{TP1}John wonder [_{CP}what_i [_{TP2}Bill bought t_i t_j]]]]]
 (iii) a. [_{TP2}Bill bought what how]
 b. [_{CP}how_i [_{TP2}Bill bought what t_i]]
 c. [_{TP1}John did wonder [_{CP}how_i [_{TP2}Bill bought what t_i]]]
 d. [_{CP}how_i did [_{TP1}John wonder [_{CP}t'_i [_{TP2}Bill bought what t_i]]]]]
 e. [_{CP}how_i did [_{TP1}John wonder [_{CP}what_j [_{TP2}Bill bought t_j t_i]]]]]

If the SCC applies to (iii) at S-structure, then it will prohibit the derivation as expected.

However, Wh-island phenomena are more puzzling than (1). So we will put them aside until chapter 4.

movement must be cyclic.

- (1) * $[_{CP} \text{Who}_i \text{ was } [_{TP} [_{DP} \text{a picture of } t_i]_j \text{ sold } t_j]]$

Sentence (1) is ungrammatical, since it violates Chomsky's (1973) Subject Condition. Roughly speaking, the Subject Condition indicates that nothing can be extracted out of a DP in $[DP, TP]$. (1) is assumed to be derived as in (2).

- (2) a. $[_{CP} \text{e was } [_{TP} \text{e } [_{VP} \text{sold } [_{DP} \text{a picture of who}]]]]$
 b. $[_{CP} \text{e was } [_{TP} [_{DP} \text{a picture of who}]_j [_{VP} \text{sold } t_j]]]$
 c. $[_{CP} \text{who}_i \text{ was } [_{TP} [_{DP} \text{a picture of } t_i]_j [_{VP} \text{sold } t_j]]]$

However, if the computational system constructs the derivation of (3) for (1) rather than (2), it can escape from the Subject Condition.

- (3) a. $[_{CP} \text{e was } [_{TP} \text{e } [_{VP} \text{sold } [_{DP} \text{a picture of who}]]]]$
 b. $[_{CP} \text{who}_i \text{ was } [_{TP} \text{e } [_{VP} \text{sold } [_{DP} \text{a picture of } t_i]]]]$
 c. $[_{CP} \text{who}_i \text{ was } [_{TP} [_{DP} \text{a picture of } t_i]_j [_{VP} \text{sold } t_j]]]$

We cannot constrain derivation (3.b) with Subjacency, since it is possible to extract a category, who, out of $[_{DP} \text{a picture of}]$ if it is a complement of a verb as in (4).

- (4) [_{CP}Who_i did [_{TP}John sell [_{DP}a picture of t_i]]]

Traditionally, the Strict Cyclic Condition (SCC), which was assumed to constrain a derivation at S-structure, forces the computational system to build (2), prohibiting (3).

However, the SCC is untenable in the minimalist program in which D-structure and S-structure are reduced to PF and LF.

In contrast to the SCC, which reflects the timing of earliness in overt derivations, we also have the Procrastinate Principle which prefers covert operations. The SCC reflects Earliness in itself, but Procrastinate is opposite in spirit to Earliness.

This chapter explains the SCC and Procrastinate with Earliness in a local way. In section 3.2.1, we review some previous efforts to reduce the SCC in the minimalist program, and in section 3.2.2 explain the SCC effects with Earliness.

Before deriving Procrastinate effects, in section 3.3 we discuss the motivations of the Procrastinate Principle, and its problems such as (i) that as a timing principle it cannot explain the timing of overt derivations at all; (ii) that its conceptual motivation is based upon some characteristics of the sensory-motor system rather than on some linguistic properties; and (iii) that its violability is not consistent with the unviolability of other economy

conditions such as the Last Resort Condition and the MLC; and (iv) that its global characteristic is also a undesirable property (as described in section 2.2).

In section 3.4 we derive Procrastinate effects from Earliness. In Section 3.4.1 we discuss Kitahara's (1994, 1995) analysis of Procrastinate and its problems, and in section 3.4.2 derive Procrastinate effects from Earliness.

3.2 Deriving Strict Cycle

3.2.1 Previous Analyses and their Problems

3.2.1.1 Extension Condition

Chomsky (1993) proposes two operations, Nonbranching Projection and Generalized Transformation, as syntactic operations. They can be defined in (5) and (6), respectively.

(5) Nonbranching Projection (NBP) (= (18), Chomsky (1993) p.21)

- a. $X \rightarrow X'$
- b. $X' \rightarrow XP$

(6) Generalized Transformation (GT) (Chomsky (1993) p.22)

- a. Target a category x ;

- b. Add an empty category e to x ;
- c. form a new category z ;
- d. Take a category y , and substitute y for e ;
- e. Form the chain $(y_i \ t_i)$ if y is contained in the targeted category x .

Chomsky (1993) assumes that all categories should be projected to a maximal projection even if there is no specifier and complement: e.g. no branching. This requirement of projection to a maximal category necessitates the NBP operation, as exemplified in (7) and (8).

- (7) $[_{NP}[_{N'}[_{N}cats]]]$
- (8) a. $[_{N}cats]$
- b. $[_{N'}[_{N}cats]]$
- c. $[_{NP}[_{N'}[_{N}cats]]]$

Contrary to NBP, a category can be projected to a maximal category or an intermediate category if it has either a specifier or a complement. Thus a branching projection requires Generalized Transformation (GT), as exemplified in (9) and (10).

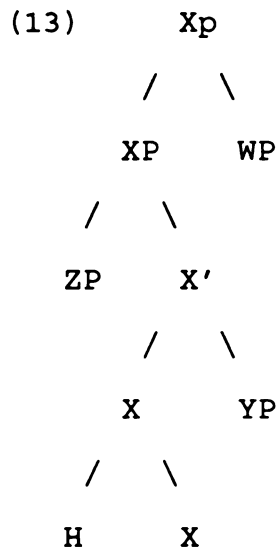
- (9) $[_{VP}[_{NP}dogs] [_{V'}[_{V}chased] [_{NP}cats]]]$
- (10) a-1. $[_{V}Chased]$
- a-2. $[_{V}chased] \ e$

- a-3. [_V [_V chased] e]
- a-4 and b-1. [_V [_V chased] [_{NP} cats]]
- b-2. e [_V [_V chased] [_{NP} cats]]
- b-3. [_{VP} e [_V [_V chased] [_{NP} cats]]]
- b-4. [_{VP} [_{NP} dogs] [_V [_V chased] [_{NP} cats]]]

Following Chomsky (1993), Move is a subcase of GT as in (11) and (12).

- (11) [_{TP} [_{NP} dogs]_i [_T T [_{VP} t_i [_V [_V saw] [_{NP} cats]]]]]
- (12) a. [_T T [_{VP} [_{NP} dogs] [_V [_V chased] [_{NP} cats]]]]
- b. e [_T T [_{VP} [_{NP} dogs] [_V [_V chased] [_{NP} cats]]]]]
- c. [_{TP} e [_T T [_{VP} [_{NP} dogs] [_V [_V chased] [_{NP} cats]]]]]
- d. [_{TP} [_{NP} dogs] [_T T [_{VP} [_{NP} dogs] [_V [_V chased] [_{NP} cats]]]]]
- e. [_{TP} [_{NP} dogs]_i [_T T [_{VP} [_{NP} dogs]_i [_V [_V chased] [_{NP} cats]]]]]

Chomsky (1993) proposes that X'-theory constrains the syntactic operations, NBP and GT. The operations, NBP and GT, should satisfy the X'-format of (13).



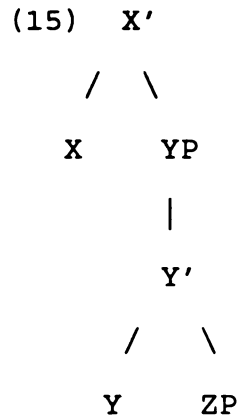
In addition, GT should satisfy the Extension Condition (EC) that "substitution operations always extend their target" (p.23). We may paraphrase the Extension Condition in (14).

(14) Extension Condition

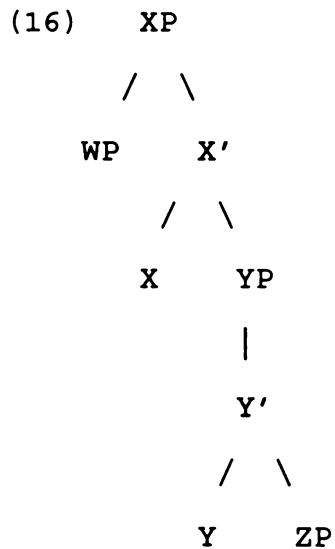
A branching operation (i.e. GT) should form a branching node which dominates (or contains) all the phrase markers in that phrase structure which it targets.

In other words, the branching node which GT creates should be the topmost phrase node in that phrase structure.

Let us consider (15). Assume that the computational system has built the phrase structure (15).

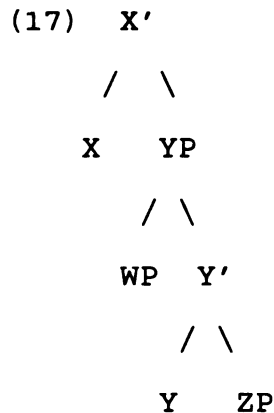


If at the point of the derivation of (15) the computational system targets X', taking a maximal projection, WP, and constructs (16), it will satisfy the EC, since the newly branching node, XP, dominates the whole phrase structure; that is, XP dominates WP, X', X, YP, Y', and ZP.



If at (15) the computational system targets Y', taking

a maximal projection, WP, and builds (17), however, it will violate the EC, since the newly branching node, YP, does not dominate the whole phrase structure; YP does not dominate X' and X.



Returning to the derivation of (1), the EC correctly blocks the derivation of (3) which is rewritten in (18).

- (18) a. $[_{CP} \text{Who}_j \text{ was } [_{TP} \text{e } [_{VP} \text{sold } [_{DP} \text{a picture of } t_j]]]]?$
 b. $[_{CP} \text{Who}_j \text{ was } [_{TP} [_{DP} \text{pictures of } t_j]_i [_{VP} \text{sold } t_i]]]?$
 -> *EC

Instead, the EC permits (2) which violates the Subject Condition (SC), rewritten in (19).

- (19) a. $[_{T'} \text{ was } [_{VP} \text{sold } [_{DP} \text{a picture of who}]]]$
 b. $[_{TP} [_{DP} \text{a picture of who}]_i \text{ was } [_{VP} \text{sold } t_i]]?$
 c. $[_{CP} \text{who}_j \text{ was } [_{TP} [_{DP} \text{a picture of } t_j]_i \text{ sold } t_i]]?$

-> *SC

Although the EC can predict the ungrammaticality of (1), it has a problem which can hardly be accommodated in minimalism.

First of all, Chomsky (1993) stipulates that the EC should not apply to adjunction operations like head movement, although they may be branching operations. For example, in French a verb overtly raises to T as an adjunction, as in (20). This operation extends the head T, but cannot extend T' which is the highest category which dominates all other categories.

(20) a. T'

```

      /  \
     T   VP
      |
    ...V...

```

b. T'

```

      /  \
     T   VP
    /  \  |
   Vi T ...ti...

```

Second, Chomsky stipulates that the EC must apply only

to overt operations--it should not apply to LF operations. For example, in English an object is assumed to raise to the spec of Agr₀ for Accusative, as in (21). This covert movement violates the EC, since it does not extend the highest category CP.²²

- (21) a. [_{CP} [_{TP} John_i [_{AgrPO} [_{Agr'O} [_{VP} t_i [_V saw Mary]]]]]]
 b. [_{CP} [_{TP} John_i [_{AgrPO} Mary_j [_{Agr'O} [_{VP} t_i [_V saw t_j]]]]]]]

To pursue minimalism, the stipulations of conditions should be removed from the theory. Furthermore, the stipulations for the EC cannot be maintained in minimalism. Because DS and SS are not available, it is unclear how the stipulation which distinguishes between overt and covert movement can be stated.

Thus the EC should be replaced with another principle, or its stipulative assumptions should be removed in this sense.

3.2.1.2 Target- α

Kitahara (1994, 1995) argues that Target- α can remove Chomsky's Extension Condition, incorporating it with the

²²In Chomsky's (1995) theory of grammar covert movement is supposed to be an operation like a head movement. Thus the stipulation for covert movement would be the same as that for overt head movement as we have discussed just above.

economy principle. Kitahara (1994) unifies two operations, NBP and GT, into a generalized targeting operation, Target- α . The operation of Target- α is stated in (22).

(22) (= 26 Kitahara 1995)

Target- α : Target a category α , and

- a. Build a new phrase structure β immediately dominating α .
- b. Substitute a category β for a newly created empty e external to α .

In addition to Target- α , he proposes the following economy principle:

(23) If a derivation D1 takes less targeting operations than a derivation D2, then D1 blocks D2.

Consider (24). To derive (24), the computational system can take Target operations in two different ways, as in (25) and (26).

(24) [_{VP} [_{NP} dogs] [_V [_V chased] [_{NP} cats]]]

(25) a. [_V Chased]

b-1. [_V chased] e

b-2. [_V [_V chased] e]

b-3 and c-1. [_V [_V chased] [_{NP} cats]]

- c-2. e [v, [vchased] [NPcats]]
 c-3. [VP e [v, [vchased] [NPcats]]]
 c-4. [VP [NPdogs] [v, [vchased] [NPcats]]]
 (26) a. [vChased]
 b and c-1. [v, [vchased]]
 c-2. e [v, [vchased]]
 c-3. [VP e [v, [vchased]]]
 c-4 and d-1. [VP [NPdogs] [v, [vchased]]]
 d-1. [VP [NPdogs] [v, [vchased] e]]
 d-2. [VP [DPdogs] [v, [vchased] [DPcats]]]

If we compare (25) with (26) in terms of the number of Target operations, (25) takes three targeting operations, and (26) takes four targeting operations. So (25) blocks (26), following the economy principle (23).

Now let us consider (1), repeated in (27), under the Target- α analysis. Again the computational system constructs (1) as either (28) or (29).

- (27) *_{CP}Who_i was _{TP}[_{DP}a picture of t_i]_j sold t_j]]?
 (28) a. [_Twas sold [_{DP}a picture of who]]
 b. [_{TP}[_Twas sold [_{DP}a picture of who]]]
 c. [_C[_{TP}[_Twas sold [_{DP}a picture of who]]]]
 d. [_{CP}who_i [_C[_{TP}[_Twas sold [_{DP}a picture of t_i]]]]]
 e. [_{CP}who_i [_C[_{TP}[_{DP}a picture of t_i]_j [_Twas sold t_j]]]]]

- (29) a. [_T.was sold [_{DP}a picture of who]]
 b. [_{TP}[_{DP}a picture of who]_j [_T.was sold t_j]]
 c. [_C [_{TP}[_{DP}a picture of who]_j [_T.was sold t_j]]]
 d. [_{CP}who_i [_C [_{TP}[_{DP}a picture of t_i]_j [_T.was sold t_j]]]]]

If we compare (28) with (29), the derivation of (29) takes one less target operations than that of (28), and thus (29) blocks (28).

One of the good things about Target- α is that it selects one or more optimal derivations even if they are nonconvergent. For example, (29) cannot converge: who in a picture of who should raise to CP and check the strong Wh-feature of C, but cannot be extracted out of the tensed subject, which would violate the Subject Condition. In spite of this, Target- α considers (29) to be optimal in comparison with (28).

Target- α has some problems, however. One is that the economy principle (23) should still apply globally. Suppose that the computational system constructs up to (28.b) and (29.b). But it cannot block (28.b) at this point, because they have the same number of target operations up to this point. The computational system further constructs up to (28.d) and (29.d), but both of them still have the same number of target operations. The economy principle finally selects (29) at the point of (28.e). So target- α has the

same problems as the global condition, as we have discussed in chapter 2.

Another problem is that target- α is no longer accommodated for Chomsky's (1994) model of the minimalist program. Target- α can be maintained in the model in which a nonbranching projection operation is available. But Chomsky (1994) eliminates a nonbranching operation from UG. Let us consider (1) with Target- α in the model of Chomsky's (1994) Bare Phrase Structures.

- (30) a. [_{TP}was sold [_{DP}a picture of who]]
 b. [_{CP}C [_{TP}was sold [_{DP}a picture of who]]]
 c. [_{CP}who_i [_{C'}C [_{TP}was sold [_{DP}a picture of t_i]]]]
 d. [_{CP}who_i [_{C'}C [_{TP}[_{DP}a picture of t_i]_j [_{T'}was sold t_j]]]]]
 t_j]]]]]
 (31) a. [_{T'}was sold [_{DP}a picture of who]]
 b. [_{TP}[_{DP}a picture of who]_j [_{T'}was sold t_j]]
 c. [_{CP}C [_{TP}[_{DP}a picture of who]_j [_{T'}was sold t_j]]]
 d. [_{CP}who_i [_{C'}C [_{TP}[_{DP}a picture of t_i]_j [_{T'}was sold t_j]]]]]
 t_j]]]]]

As we see in (30) and (31), both of the derivations (30) and (31) have the same number of target operations. Thus Target- α fails to make (31) block (30) in the model of Bare Phrase Structures.

To fix this problem, Kitahara (1996) proposes three

syntactic operations, Merge, Move, and Replace, for phrase structure constructions, and also decomposes Chomsky's Move operation into these three. They can be defined as follows:

(32) Merge: concatenate two elements.

(33) Move: target and raise.

(34) Replace: replace.

If we consider (1) again, two possible derivations are available:

(35) a. $TP = \{T(\text{was}) [\text{sold } [_{DP} \text{a picture of who}]]\}$

-> Merge

b. $CP = \{C, TP\}$ -> Merge

c. $CP_1 = \{\text{who}_i, CP\}$ -> Merge and Move

d. $TP_1 = \{[_{DP} \text{a picture of } t_i]_j, TP\}$ -> Move and Merge

e. $CP = \{C, TP_1\}$ -> Replace

(36) a. $TP = \{T(\text{was}), [\text{sold } [_{DP} \text{a picture of who}]]\}$ -> Merge

b. $TP_1 = \{[_{DP} \text{a picture of who}]_i, TP\}$ -> Move and Merge

c. $CP = \{C, TP_1\}$ -> Merge

d. $CP_1 = \{\text{who}_j, CP\}$ -> Move and Merge

(35) takes four Merge operations, two Move operations,

and one Replace operation, whose total number of operations are seven. (36) takes four Merge operations and two Move operations, whose total number of operations are six. If we compare (35) with (36), (35) takes one more operations than (36). Thus (36) blocks (35), adopting the economy principle in (23).

In this case, too, Kitahara must assume that the economy principle is a global condition, because this time a derivation with no movement always takes less operations than a derivation with movement. This indicates that nonconvergent derivations would always block convergent derivations, as we have discussed in chapter 2.

In addition, Kitahara (1996) splits one single Move operation into independent operations. However, Chomsky explicitly defines Move with raise, merge, and replace, all of which are the internal noninterruptible suboperations of Move, and should not be seen by conditions. (See p.32 in chapter 1. for detail) Kitahara claims that Raise, Merge and Replace are independent operations. Then Move may be interruptible, and so it is not clear they can be counted as Kitahara suggests.

3.2.1.3 Crossing the Number of Nodes

Chomsky (1994) and Collins (1994) consider the number of nodes that movement crosses to be costly under economy

considerations as in the following.

- (37) The less nodes movement crosses, the more economical it is.

Consider (38).

- (38) a. [_Cwas [_{TP}sold [_{DP}a picture of who]]]
 b. [_Cwas [_{TP}[_{DP}a picture of who] sold]

Chomsky (1994) argues that the extraction out of (38.b) is more economical than that out of (38.a), since the WH-phrase who moves across more nodes in (38.a) than (38.b). If we simply counts the maximal categories that who crosses, it crosses DP, VP, and TP in (38.b), but crosses DP, and TP in (38.a). This view of economy has at least two problems. First, it cannot explain why (39.a) cannot block (39.b), although the Wh-phrases in (39.a) crosses less nodes than the Wh-phrase in (39.b).

- (39) a. Which person did John tell to buy which picture?
 b. Which picture did John tell which person to buy?

Second, the economy principle is global again.

3.2.1.4 Feature Strength and Cyclicity

Giving up explaining cyclicity in terms of economy, Chomsky (1995) suggests an alternative. Following his argument, a strong feature has two properties: First, it triggers an overt operation, and second, induces cyclicity. For the first case, the pre-Spell-Out property, a strong feature is assumed to crash at PF and therefore must be removed before Spell-Out. For the second case, cyclicity in overt derivations, a derivation is assumed to be not able to tolerate strength: a strong feature cannot be passed by an operation and later checked by another operation. That is, a derivation D is canceled if D contains a strong feature.

In the case of the derivations for (1), only the derivation (2) is a legitimate derivation, repeated in (41), because, as repeated in (40), (3) contains a strong feature within it.

- (40) a. $[_{CP} \text{Who}_j \text{ was } [_{TP} \text{e } [_{VP} \text{sold } [_{DP} \text{a picture of } t_j]]]]?$
 b. $[_{CP} \text{Who}_j \text{ was } [_{TP} [_{DP} \text{pictures of } t_j]_i [_{VP} \text{sold } t_i]]]?$
- (41) a. $[_{T'} \text{ was } [_{VP} \text{sold } [_{DP} \text{a picture of who}]]]$
 b. $[_{TP} [_{DP} \text{a picture of who}]_i \text{ was } [_{VP} \text{sold } t_i]]?$
 c. $[_{CP} \text{who}_j \text{ was } [_{TP} [_{DP} \text{a picture of } t_j]_i \text{ sold } t_i]]?$
 -> *SC

Again, this statement is completely a stipulation.

Thus we cannot answer the question of why only a strong feature cannot be contained within legitimate derivations.

3.2.1.5 Chain Interleaving

Following Chomsky (1993) and Collins (1994), the formation of chains is one single operation. For example, who raises to the matrix CP, and forms a chain <who, t', t> in (42).

- (42) a. [_{CP}who did [_{TP}Bill think [_{CP}t' that [John sold a picture of t]]]]
 b. <who, t', t>

Then the question is whether the chain formation of (42.b) should be considered as two instances of movement, or as one instance of movement. Chomsky (1993) argues that chain formation should be one instance of movement. If the chain (42.b) is formed as two operations of movement, the economy considerations would block it with another derivation with one movement--which is less costly-- as shown in (43).

- (43) a. [_{CP}who did [_{TP}Bill think [_{CP}that [John sold a picture of t]]]]
 b. <who, t>

Following Chomsky (1993), (42) and (43) are equally costly if we assume that the chain formation is one movement.

Collins (1994) formalizes this characteristics as follows:

(44) A chain must not be interleaved

(45) Two chains, X and Y, are interleaved if during a derivation, part of X is formed, then part of Y is formed, then part of X is formed, and so on.

(p.47)

Following Collins (1994), (44) is derivable from the assumption that chain formation is one single operation.

Then we can apply (44) to explaining (1). Consider the procedures to construct (1).

- (46) a. [_{VP}was sold [_{DP}a picture of who]]
 b. [_{VP}who_i [_{VP}was sold [_{DP}a picture of t_i]]]
 c. [_{TP}[_{DP}a picture of t_i]_j [_{VP}who_i [_{VP}was sold t_j]]]
 d. [_{CP}who_i [_{TP}[_{DP}a picture of t_i]_j [_{VP}t'_i [_{VP}was sold t_j]]]]]

Collins (1994) argues that the derivation of (46) violates (44).

But we still have an alternative to derive (1) as

follows:

- (47) a. [_{CP}[_{TP}was sold [_{DP}a picture of who]]]
 b. [_{CP}who_i [_{TP}was sold [_{DP}a picture of t_i]]]
 c. [_{CP}who_i [_{TP}[_{DP}a picture of t_i]_j was sold t_j]]

This observes (44), forming the chains as one instance of movement. To filter out (47), we still need the equivalent of the SCC we have already seen cannot easily be derived.

3.2.2 Earliness and the Strict Cycle

In this section we will argue that Earliness as a local economy condition can predict that overt derivations are cyclic without any further assumptions and stipulations.

Consider the derivation (48). The computational system constructs up to (48.a). At this point of the derivation the computational system can possibly take one of (49.a) and (49.b). If it takes (49.b), it would violate Earliness, because this point (49.a) is the earliest time to be able to check the DP and the strong D feature of Tense, although their features can be checked later as in (50) if it takes (49.b). Thus (49.a) is optimal in comparison with (49.b) at this point of the derivation, and blocks (49.b) and its subsequent derivations.

- (48) [_{TP}was sold [_{DP}a picture of who]]
- (49) a. [_{TP}[_{DP}a picture of who]_i was sold t_i]
 b. *[_CP C [_{TP}was sold [_{DP}a picture of who]]]
- (50) *[_CP Who_j was [_{TP}[_{DP}a picture of t_j]_i sold t_i?]]]
- (51) [_CP C [_{TP}[_{DP}a picture of who]_i was sold t_i]]]

After that, the computational system further builds a phrase structure (51) from (49.a). At this point the strong +wh feature of C should have attracted any +Wh feature to satisfy bare output conditions. However, the computational system cannot utilize Move any more, since who is in a Subject Condition island, and so is not accessible (or visible) to the computational system²³. If it reaches the interface levels, then it crashes, because the strong feature is not interpretable at PF and at LF.

3.3 Procrastinate: the Background and Problems

3.3.1 Motivations

A well known difference between English and French is the word order of finite verbs relative to a negative morpheme and adverbs. As exemplified in (52) and (53),

²³We assume that Attract-F can only attract some feature within some (minimal) domain, and that the Constraint on Extraction Domain (Huang 1982) including the Subject Condition and the Adjunct Condition should not be in this domain.

French finite main verbs precede the negative pas "not" and adverb souvent "often", while English ones do not precede the negative not and adverb often.

(52) a. Jean embrasse souvent Marie.

John kisses often Mary

"John often kisses Mary."

b. Jean (n') aime pas Marie.

John likes not Mary

"John does not like Mary."

(53) a. John often kisses Mary.

b. John does not like Mary.

More correctly speaking, French inflected main verbs obligatorily precede negation and adverbs, and English ones may not precede them. The following examples will show the impossibility that English main verbs come before a negative and adverbs, and that French ones come after them.

(54) a.* Jean (ne) pas aime Marie.

b.* Jean souvent embrasse Marie.

(55) a.* John likes not Mary.

b.* John kisses often Mary.

According to Emonds (1978), Pollock (1989), and Chomsky (1991), both in French and English the main verbs are

generated in the V-position of the VP as in (56.a) and (57.a), respectively. In overt derivations, however, English finite main verbs remain in the V-position of a VP as in (57.b), while French ones are in the T-position of a TP, raising the V in a VP to the T of TP, as (56.b).²⁴

Since English finite main verbs have overt affixation like the 3rd singular person, we may have a question about how the affixation of finite main verbs are possible if they remain in the V-situ position, not raising to T, in the overt derivation. The Emonds-Pollock-Chomsky analyses suppose that T is overtly lowered to the verb, leaving the T-trace unbound, as in (57.b). The amalgated V-T raises to T at LF as in (57.c), though, to remedy the unbound trace which will violate some condition such as the ECP in that framework.

- (56) a. [_{TP}Jean T [_{VP}souvent [_{VP}embrasse Marie]]].
 b. [_{TP}Jean embrasse_i-T [_{VP}souvent [_{VP}t_i Marie]]].
 (overt raising)
- (57) a. [_{TP}John T [_{VP}often [_{VP}kisses Mary]]].
 b. [_{TP}John t_i [_{VP}often [_{VP}kisses-T_i Mary]]]. (overt
 lowering)
 c. [_{TP}John [kisses-T]_i [_{VP}often [_{VP}t_i Mary]]]. (covert
 raising)

²⁴Their analysis is based on the assumption that a negative like not and adverbs like often are posited between T and VP invariantly across languages.

If we compare the length of the derivation of (56) with that of (57), the former takes one step less than the latter. Hence Chomsky (1991) concludes that raising is less costly than lowering, since lowering always presumably requires raising again to remedy an unbound trace.

With Chomsky (1993) and Chomsky and Lasnik's (1993) introduction of the checking theory, lowering processes become unnecessary. As we have discussed just above, T-lowering was required to explain the visibility of verb affixes (at S-structure). In the checking theory it is assumed that the affixation of a syntactic category e.g. kiss plus the Present 3rd Singular features is done before being drawn from the lexicon, and inserted to a phrase structure rather than that affixation is done syntactically. Syntactic processes simply check the features of a category with a corresponding functional category and determine whether the inflection is correct or not. Thus the base generation of (52.a) and (53.a) can be represented as follows:

(58) [_{TP}Jean T-(3rd sg pres) [_{VP}souvent [_{VP}embrasse-(3rd sg pres) Marie]]].

(59) [_{TP}John T-(3rd sg pres) [_{VP}often [_{VP}kisses-(3rd sg pres) Mary]]].

Then the inflectional features must check and match

with the features of the T, raising the verb to the T. The only difference between French and English is the timing of movement which takes place: French main verbs move overtly or before Spell-out, and English ones move covertly or after Spell-out. So the further derivations of (58) and (59) can be represented as follows:

(60) a. [_{TP}Jean embrasse_i-T-(3rd sg pres) [_{VP}souvent [_{VP}t_i Marie]]].

b. Spell out

(61) a. Spell out

b. [_{TP}John kisses_i-T-(3rd sg pres) [_{VP}often [_{VP}t_i Mary]]].

Chomsky (1993, 1994, 1995) argues that it depends on the strength of the formal features of functional categories whether movement takes place overtly or covertly. That is, if a feature is strong it must be checked and eliminated before Spell-out for convergence at PF and LF, since it is an uninterpretable feature at PF and LF; if it remains after Spell-out, the derivation crashes at PF, although they are eliminated at LF. Weak features can be eliminated after Spell-out because they are invisible at PF; but they must be eliminated at LF for convergence. Furthermore, the strength of formal features is assumed to be parametrizable language to language.

Returning to the case of French and English verb movement, French T is assumed to have a strong V feature, and English T is not. In French a feature of T is strong and must be eliminated before PF by raising V to T for feature checking as in (60); otherwise the derivation would crash at PF. Thus the verb movement must occur overtly. In English the verb does not have to move overtly, since English T does not have a strong feature. But the verb is supposed to have an uninterpretable feature like a tense feature, and so it must be eliminated before LF, and so English main verb moves covertly as in (61).

We have one more question to ask for this analysis. Suppose that kisses raises to T before Spell-out as in (55). It converges at PF and LF, too, because it does not have any feature which is uninterpretable at PF and at LF. Why shouldn't English main verbs raise before Spell-out at all, as exemplified in (55), even if it can converge at PF and LF?

Chomsky (1993, 1994, 1995) claims that this is due to the Procrastinate Principle.

(62) The Procrastinate Principle

Minimize overt operations.

He assumes that covert movement is cheaper than overt movement, since LF operations are "operating mechanically

beyond any directly observable effects." (p.30) For the reason that covert operations are less costly than overt operations the computational system tries to minimize overt operations and maximize covert operations. That is, the computational system prefers covert operations to overt operations unless only overt operations make a derivation converge.

In English, main verbs must not move to T overtly, even if this overt verb movement can prevent the derivation from crashing at PF and at LF, since covert verb movement can also prevent it from crashing.

Then a question may arise why overt movement like verb movement in French is permitted to violate the Procrastinate Principle. Chomsky claims that the economy conditions should apply only to convergent derivations, and among them select an optimal derivation. So the Procrastinate Principle can be violated by the requirement for convergence.

To sum up, covert operations are less costly than overt ones, since the former are not directly observable mechanically. In addition, the Procrastinate Principle is violable for convergence. Therefore the Procrastinate Principle applies only to convergent derivations, and among them selects an optimal derivation which takes the least number of covert operations.

3.3.2 Some Problems

First of all, Procrastinate is too loose to explain all the cases of the timing of movement. It simply considers one point of a derivation--the point of Spell-out-- to be a critical time for movement, so that, among convergent derivations, it selects a derivation which takes the least number of overt operations. This comes from the assumption that Procrastinate prefers covert operations to overt operations. As we have shown in section 3.2, however, there is some evidence that the computational system determines whether it should take movement at each point of a derivation, rather than at the point of Spell-out.

Consider (1) again, repeated in (63). (63.a) is assumed to be represented as (63.b) at LF.

- (63) a. *Who was a picture of sold?
 b. * $[_{CP} \text{Who}_j \text{ was } [_{TP} [_{DP} \text{a picture of } t_j]_i \text{ sold } t_i?]]$

(63) is ungrammatical, since it violates Chomsky's (1973) Subject Condition: nothing can be extracted out of a DP in [DP, TP], roughly speaking. (63) is assumed to be derived as in (64).

- (64) a. $[_{TP} T [_{VP} \text{sold } [_{DP} \text{a picture of who}]]]$?
 b. $[_{TP} [_{DP} \text{a picture of who}]_j [_{T'} T [_{VP} \text{sold } t_j]]]$?

c. [_{CP}who_i was [_{TP}[_{DP}a picture of t_i]_j [_{VP}sold t_j]]]]?

However, if the computational system constructs the derivation of (65) for (63) rather than (64), it can escape from the Subject Condition.

(65) a. [_{CP}e was [_{TP}e [_{VP}sold [_{DP}a picture of who]]]]?

b. [_{CP}who_i was [_{TP}e [_{VP}sold [_{DP}a picture of t_i]]]]?

c. [_{CP}who_i was [_{TP}[_{DP}a picture of t_i]_j [_{VP}sold t_j]]]]?

We cannot explain (65) with Subjacency, since it is possible to extract a category out of [_{DP}a picture of] if it is a complement of a verb as in (66).

(66) [_{CP}Who_i did [_{TP}John sell [_{NP}a picture of t_i]]]?

To explain the ungrammaticality of (63), thus it is far more important when who and a picture of who move, rather than whether they move overtly or covertly. For (63), the computational system must first construct (64.b), raising the DP to the TP, and then (64.c), raising who to the CP, rather than derive (65.a)-(65.c) in that order. That is, the computational system is forced by some principle to move [_{DP}a picture of who] to TP without any delay right after TP is constructed.

Procrastinate simply allows the computational system to

move the Wh-phrase and DP overtly, since in English a D feature of T and a Wh feature of C are both strong, but cannot force the computational system to take the derivation of (64) rather than (65) at all. UG requires some additional principle in order to filter out (63).

To motivate Procrastinate as an economy condition in UG, Chomsky argues:

... LF operations are a kind of "wired-in" reflex, operating mechanically beyond any directly observable effects. They are less costly than overt operations. The system tries to reach PF "as fast as possible," minimizing overt syntax. (PP.30-31)

Yet his intuitive argument is obscure. First, it is not clear why the operations "mechanically beyond any directly observable effect" (p.30) are less costly than any directly observable operations. All syntactic operations are, overt or covert, observable by UG. So the argument that overt operations are observable and that covert operations are unobservable may be related to the sensory-motor system which is not supposed to affect language under Chomsky's (1995) "... speculation that the essential character of C_{HL} is independent of the sensory-motor interface." (p.335) Hence the economy conditions should be considered in the sense of language properties.

As Brody (1995) also points out, Procrastinate may be an unnatural economy condition for UG. Procrastinate implies that the default case of economy is to make the LF

form maximally different from the PF form, which is not a natural expectation. One would expect the LF form to be maximally similar to the PF form as the default case, so that it can be recovered with minimum effort.

The violability of the Procrastinate Principle is also inconsistent with the unviolability of other universal principles. In generative grammar, including the minimalist program, it has been in general assumed that all universal principles should be observed and that some deviance should be yielded if any universal principle is violated. But the Procrastinate Principle is the only exception to this assumption. It can be violated if it is necessary for convergence, yielding no deviance. Rather, it must be violated for convergence.

In addition, undesirably, the Procrastinate Principle has the characteristics of a global condition, although other economy conditions can be formulated in a local way for their application, as we have discussed in chapter 2.

In next section we will derive Procrastinate from a different timing principle, the Earliness Principle, which is independently motivated by overt cyclic derivations.

3.4 Deriving Procrastinate Effects

3.4.1 PF Deletion Analysis

Kitahara (1994, 1995) proposes that Procrastinate is derivable from Target- α . Under his analysis it depends on the number of targeting operations at the interface levels whether an operation is obligatorily covert (Procrastinate) or optionally covert (optional movement). In other words, if movement obligatorily takes place covertly like object shift in English, it is due to the fact that covert movement takes fewer targeting operations than overt movement: it is the case that covert movement is more economical than overt movement. If movement optionally takes place covertly, it comes from the fact that covert movement takes the same number of targeting operations as overt movement: overt and covert movement are equally economical.

Let us take (67), which is the case of English object shift which obligatorily takes place covertly.

(67) John kisses Mary.

According to Kitahara, the computational system will take the steps in (68).

- (68) a. [_{AgRP-O} Agr₀ [_{VP} John kisses Mary]]
 b. [_{AgRP-O} e [_{AgR'-O} Agr₀ [_{VP} John kisses Mary]]]
 c. [_{TP} John_i [_{AgRP-O} e [_{AgR'-O} Agr₀ [_{VP} t_i kisses Mary]]]]
 d. Spell-out
 e. [_{TP} John_i [_{AgRP-O} e [_{AgR'-O} kisses_j + Agr₀ [_{VP} t_i t_j Mary]]]]]

f. [_{TP}John_i [_{AgrP-O}Mary_k [_{Agr'-O}kiss_j [_{VP}t_i t_j t_k]]]]]

If we compare (68) with (69) which is derived for overt object shift, the derivation of (68) takes one more targeting operations than (69). In other words, overt computations take less operations than covert computations under the target- α analysis. This would wrongly predict that there must be no procrastinate effect in natural language.

- (69) a. [_{AgrP-O}Agr₀ [_{VP}John kisses Mary]]
 b. [_{AgrP-O}kisses_j+Agr₀ [_{VP}John t_j Mary]]
 c. [_{AgrP-O}Mary_k [_{Agr'-O}kisses_j+Agr₀ [_{VP}John t_j t_k]]]
 d. [_{TP}John_i [_{AgrP-O}Mary_k [_{Agr'-O}kisses_j+Agr₀ [_{VP}t_i t_j t_k]]]]]
 e. Spell-out

To solve this problem, Kitahara extends the targeting operation to a Delete operation at PF. He argues that under the copy theory of movement a trace of movement is exactly the same copy as the moved category. A trace is, apparently, phonetically null, although it is exactly the same as the moved category. Arguably, it indicates that the copy left by movement, e.g. a trace, deletes at the PF component. (CF. Affect- α in Chomsky and Lasnik (1993), Lasnik and Saito (1984, 1992)) He subsumes PF Delete under a

targeting operation as in the following:

(70) (=76) Kitahara 1994 p.41)

Target- α (targeting a category α)

- a. Build a new category τ by merging α and an empty ϕ
- b. Substitute a category β for ϕ sister to α
- c. Delete α

Now overt computations induce Delete operations at PF, where as Delete operations are not necessary for covert computations.

Under the target as defined in (70), let us consider (68) and (69), which are derived as in (71) and (72), respectively.

- (71) a. [_{AgrP-O}Agr_O [_{VP}John kisses Mary]]
 b. [_{AgrP-O}e [_{Agr'-O}Agr_O [_{VP}John kisses Mary]]]
 c. [_{TP}John_i [_{AgrP-O}e [_{Agr'-O}Agr_O [_{VP}t_i kisses Mary]]]]
 d. Spell-out
 e. [_{TP}John_i [_{AgrP-O}e [_{Agr'-O}kisses_j+Agr_O [_{VP}t_i t_j Mary]]]]
 f. [_{TP}John_i [_{AgrP-O}Mary_k [_{Agr'-O}kiss_j [_{VP}t_i t_j t_k]]]]
 g. Delete t_i at PF
- (72) a. [_{AgrP-O}Agr_O [_{VP}John kisses Mary]]
 b. [_{AgrP-O}kisses_j+Agr_O [_{VP}John t_j Mary]]]
 c. [_{AgrP-O}Mary_k [_{Agr'-O}kisses_j+Agr_O [_{VP}John t_j t_k]]]

- d. [_{TP}John_i [_{AgrP-O}Mary_k [_{Agr'-O}kisses_j+Agr_O [_{VP}t_i t_j t_k]]]]
- e. Spell-out
- f. Delete t_i at PF
- g. Delete t_j at PF.
- h. Delete t_k at PF.

Now we compare (71) with (72) in terms of the number of targeting operations. The derivation (71) now takes one less targeting operations than (72). So English object shift is obligatorily covert, which is a Procrastinate effect.

Let us consider (73) instead of (72) for English object shift.

- (73) a. [_{AgrP-O}Agr_O [_{VP}John kisses Mary]]
- b. [_{AgrP-O}Mary_k [_{Agr'-O}Agr_O [_{VP}John kisses t_k]]]
- c. [_{TP}John_i [_{AgrP-O}Mary_k [_{Agr'-O}Agr_O [_{VP}t_i kisses t_k]]]]
- d. Spell-out
- e. [_{TP}John_i [_{AgrP-O}Mary_k [_{Agr'-O}kisses_j+Agr_O [_{VP}t_i t_j t_k]]]]
- f. Delete t_i at PF
- g. Delete t_k at PF.

If we compare (73) with (71) in terms of the number of targeting operations, they are equally economical, since they take the same number of targeting operations. Kitahara rejects the derivation of (73) for object shift, adopting

Holmberg's (1986) generalization that object shift requires a verb to raise to Agr₀ before it. In other words, for the object Mary to be raised to AgrP₀ as in (71.f) and in (72.c), the verb kisses should be raised to Agr₀ prior to object raising, as in (71.e) and in (72.b). In this sense the derivation of (73) is not legitimate, violating Holmberg's generalization, since the object shift in (73.b) does not induce a verb raising prior to it. Thus object shift indicates that both a verb and an object must move. It means that one object movement induces two movements: a verb movement and an object movement. If it takes place overtly, two Delete operations will be induced. Overt object shift is this more costly than covert object shift.

If we take French object shift, however, we can notice that the target- α analysis will fail. In French verb raising is known to be obligatorily overt, being similar to Icelandic verb raising but contrasting to English main verb raising. But French object shift is obligatorily covert as in (74).

(74) a. [_{TP}Jean_i embrasse_j+T [_{VP}souvent [_{VP}t_i t_j Marie]]]

b. *[_{TP}Jean_i embrasse_j+T Marie_k [_{VP}souvent [_{VP}t_i t_j t_k]]]

(75) a. [_{AgrP-0}Agr₀ [_{VP}souvent [_{VP}Jean embrasse Marie]]]

b. [_{AgrP-0}embrasse_i+Agr₀ [_{VP}souvent [_{VP}Jean t_i Marie]]]

c. [_{AgrP-0}e [_{Agr'-0}embrasse_i+Agr₀ [_{VP}souvent [_{VP}Jean t_i

souvent]]]]

- d. [_{TP}Jean_j [_{AgrP-O}e [_{Agr'-O}embrasse_i+Agr_O [_{VP}souvent
[_{VP}t_j t_i Marie]]]]]]
- e. Spell-out
- f. [_{TP}Jean_j [_{AgrP-O}Marie_k [_{Agr'-O}embrasse_i+Agr_O [_{VP}souvent
[_{VP}t_j t_i t_k]]]]]]
- g. Delete t_j
- h. Delete t_i

- (76) a. [_{AgrP-O}Agr_O [_{VP}souvent [_{VP}Jean embrasse Marie]]]
- b. [_{AgrP-O}embrasse_i+Agr_O [_{VP}souvent [_{VP}Jean t_i Marie]]]
- c. [_{TP}Jean_j [_{AgrP-O}embrasse_i+Agr_O [_{VP}souvent [_{VP}t_j t_i
Marie]]]]]
- d. [_{TP}Jean_j [_{AgrP-O}Marie_k [_{Agr'-O}embrasse_i+Agr_O [_{VP}souvent
[_{VP}t_j t_i t_k]]]]]]
- e. Spellout
- f. Delete t_j
- g. Delete t_i
- h. Delete t_k

Suppose that the computational system constructs (74.a) and (74.b) as in (75) and (76), respectively. If we compare the derivations (75) and (76) in terms of the number of targeting operations, we can see that both of them take the same number of targeting operations. In spite of the equal number of targeting applications, French object shift must be obligatorily covert.

In addition, as we have discussed in chapter 2 (section 2.2.5) and chapter 3 (section 3.2.1.2), Target- α assumes global economy on derivations which is undesirable.

Target- α cannot also be maintained in Chomsky's (1994, 1995) bare phrase structure.

In next section we will derive Procrastinate from Earliness from which overt cyclic derivations are derived.

3.4.2 Earliness and Procrastinate Effects

In this section we will argue that Procrastinate effects should be derived from Earliness. According to our Earliness Principle formulated in (77), a derivation must satisfy bare output conditions as early as possible.

- (77) K attracts F early only if a sublabel of K is an uninterpretable feature at the interface level that Attract-F affects.²⁵

Consider covert verb movement and object shift in

²⁵We assume that there are three types of uninterpretable formal features. The first type is such a feature as a strong feature which is uninterpretable at PF and at LF. In this case, Attract-F should affect both PF and LF. In other words, the movement affects sound and meaning. The second type is such a feature as a Case feature which is uninterpretable only at LF. In this case, Attract-F should affect only LF. It is the case of covert movement which affects meaning, not sound. The third type is a feature which is uninterpretable only at PF. In this case, Attract-F should affect only PF. Scrambling which does not affect meaning may be subject to this case.

English as in (78):

(78) [_{TP}John T [_{VP}often [_{VP}t_i kisses Mary]]]

Suppose that the computational system has constructed (79). At this point the computational system can potentially take two choices: the strong D feature of FF(T) attracts FF(John) as in (80), or C is merged with TP, constructing CP as in (81):

(79) [_{TP}T [_{VP}often [_{VP}John kisses Mary]]]

(80) [_{TP}John_i [_T,T [_{VP}often [_{VP}t_i kisses Mary]]]]

(81) [_{CP}C[_{TP}T [_{VP}often [_{VP}John kisses Mary]]]]

The strong D feature of FF(T) is uninterpretable at PF and LF, and Attract-F also affects PF and LF. Hence (79) is the earliest point for Attract-F, and the computational system takes (80); otherwise it would violate Earliness as in (81).

At the point of (80) the computational system can take two choices again: it merges C and TP to CP as in (82), or the V feature of FF(T) attracts the verb kisses as in (83) because it is an uninterpretable feature.

(82) [_{CP}C [_{TP}John_i [_T,T [_{VP}often [_{VP}t_i kisses Mary]]]]]

(83) [_{TP}John_i [_T,kisses_j+T [_{VP}often [_{VP}t_i t_j Mary]]]]

But at this point the computational system takes (82) rather than (83): this Attract-F affects PF and LF, but the V feature of FF(T) is uninterpretable only at LF; hence (80) is not the earliest point to attract the verb. This is the Procrastinate effect of English verb movement.

The above result leads us to covert object shift in English: in English object shift cannot take place overtly, since main verbs raise covertly, and hence no feature attracts the object overtly. We derive Holmberg's (1986) generalization from Attract-F and Earliness.

For example, after the computational system takes (82) for further computation, it takes two choices: Spell-out or raise the object Mary. At this time the computational system takes Spell-out, since there is no feature of T which can attract the object.

After that, the computational system raises the verb kisses, targeting T, as in (84), since the V feature of FF(T) is uninterpretable only at LF, and Attract-F affects only LF now.

(84) [_{CP}C [_{TP}John_i kisses_j+T [_{VP}often [_{VP}t_i t_j Mary]]]]

After that, the Case feature of FF(kisses) attracts FF(Mary), as in (85), since the Case feature of FF(kisses) is uninterpretable at LF, and Attract-F affects only LF.

- (85) [_{CP}C [_{TP}John_i FF(Mary)_k+kisses_j+T [_{VP}often [_{VP}t_i t_j t_k]]]]]

Under the Earliness analysis, we can derive the Procrastinate effects of verb movement and object shift in English without reference to the Procrastinate Principle at all.

Now consider covert object shift in French, repeated in (86):

- (86) a. [_{TP}Jean_i embrasse_j+T [_{VP}souvent [_{VP}t_i t_j Marie]]]
 b. *[_{TP}Jean_i embrasse_j+T Marie_k [_{VP}souvent [_{VP}t_i t_j t_k]]]

Suppose that the computational system has constructed (87). At this point the strong D feature of FF(T) attracts the subject Jean, as in (88), in the same manner as overt subject raising in English as in (80).

- (87) [_{TP}T [_{VP}souvent [_{VP}Jean Embrasse Marie]]]
 (88) [_{TP}Jean_i T [_{VP}souvent [_{VP}t_i embrasse Marie]]]

In addition, French T has a strong V feature (Chomsky 1993), and attracts the verb embrasse and must attract it at this point, since (88) is the earliest point to affect PF and LF with a feature which is uninterpretable at PF and LF.

It thus generates (89):

(89) [_{TP}Jean_i embrasse_j+T [_{VP}souvent [_{VP}t_i t_j Marie]]]

At this point the Case feature of FF(embrasse) cannot attract FF(Marie), however: (89) is not the earliest point to attract FF(Marie), since the Case feature of FF(embrasse) is uninterpretable only at LF, and this Attract-F affects both PF and LF. After Spell-out, the computational system raises the object, targeting [_τembrasse+T].

As we have seen so far, Earliness can derive Procrastinate effects locally during the course of derivation without reference to the Procrastinate Principle.

4. A Unified Analysis of Wh-Asymmetries and Wh-Adjunct Symmetries

4.1 Introduction

This chapter attempts to derive some Wh-asymmetries and Wh-adjunct symmetries from a general economy principle on derivations, the Shortest Move or Minimal Link Condition.

In section 4.2 we first review some Wh-asymmetries (e.g. argument-adjunct, argument-extraction, argument-quasi-argument, superiority effects), their descriptive generalization, and their pre-minimalist and minimalist analyses.

In section 4.3 we then develop some theoretical hypotheses for our analysis. In section 4.3.1 we investigate some properties of Wh-words and classify their characteristic in terms of feature specifications. More specifically, we classify Wh-words into three types: Wh-DP operators, Wh-adverbial operators, and Wh-NP variables, and specify their features as $\{D, Op_0\}$, $\{Adv, Op_0\}$, and $\{D_{Pro}^{26}\}$, respectively. In addition, operator-types must undergo movement for LF legitimacy, and variable type must be bound

²⁶I will use Pro as a feature in order to indicate a property of a variable which requires an operator to bind it.

for LF legitimacy.

In Section 4.3.2 we propose multiple feature attraction in which multiple features parametrically attract F, and discuss how it works in the minimalist model. For Wh-questions, a Comp attracts only one Wh-word with an operator feature (Op) or a pair of features $\langle D, Op_0 \rangle$.

Section 4.3.3 we investigate Wh-asymmetries under the Minimal Link Condition and Attract-F which are independently necessary in the minimalist model. We extend our analysis to some Wh-adjunct symmetries (argument-adjunct, pseudo-opacity, inner island condition) in section 4.4.

4.2 Some Types of Wh-asymmetries

4.2.1 Wh-asymmetries and Pre-minimalist Analyses

Linguists have long found several phenomena of Wh-asymmetries in natural language. The first type of Wh-asymmetries are superiority effects, as exemplified in (1) and (2).

- (1) a. John wonders [_{CP}who_i [_{TP}t_i bought what]]
- b. *John wonders [_{CP}what_i [_{TP}who bought t_i]]
- (2) a. [_{CP}who_i did [_{TP}you tell t_i [_{TP}PRO to read what]]]
- b. *[_{CP}what_i did [_{TP}you tell who [_{TP}PRO to read t_i]]]

Chomsky (1973) explains the contrast in (1)-(2) in terms of a condition on transformational rules that disallows a rule to apply to an element Y if there is another element Z which is superior to Y and to which it can apply. The superiority condition is formulated in (3).

(3) Superiority Condition (= (73) Chomsky (1973) p.246)

No rule can involve X, Y in the structure

...X... [_a...Z...-WYZ...]....

where the rule applies ambiguously to Z and Y and Z is superior²⁷ to Y

The superiority Condition prevents Wh-movement from applying to what in (1) and (2), since movement can equally apply to who and what at the current cycle, and who is superior to what.

The formulation of the Superiority Condition has some empirical problems in explaining the contrasts in (4) and other Wh-asymmetries below, although its properties seem to be potentially correct.

(4) a. *I wonder what who bought t

b. Who wonders what who bought t

²⁷For simplicity let superiority be a c-command relation as follows:

(i) A category X is superior to a category Y if X c-commands Y.

Sentences (4) are typical examples which violate the Superiority Condition. Sentence (4.b) is grammatical, however, if who in the matrix clause and who in the embedded clause receive a pair-list reading (Lasnik and Saito 1992).

The second type of Wh-asymmetries is argument-adjunct asymmetries. Huang (1982) observes that extraction of a Wh-adjunct from a Wh-island yields a worse deviance than extraction of an argument. They are exemplified in (5):

- (5) a. ?[_{CP}what_i did John wonder [_{CP}whether to fix t_i]]
 b. *[_{CP}How_j did John wonder [_{CP}whether to fix the car t_j]]

The argument-adjunct Wh-asymmetries can be also observed in multiple Wh-question constructions:

- (6) a. [_{CP}how_j did Fred fix what t_j]
 b. *[_{CP}what_i did Fred fix t_i how]

Huang (1982) attributes these contrasts in (5)-(6) to the ECP. The ECP can be formulated as follows²⁸: (Chomsky 1981)

²⁸For the ECP to work out, we need other auxiliary hypotheses such as the Comp-indexing algorithm, no application of Subjacency/CED to LF movement, and so on. We ignore technical details to focus our discussion.

- (7) A nonpronominal empty category must be properly governed.
- (8) α properly governs β iff α governs β and (i) α is a lexical head (lexical government), or (ii) α is coindexed with β (antecedent government).
- (9) α governs β iff for all x , x a maximal projection, x dominates α iff x dominates β . (Aoun & Sportiche 1981)

Following Huang's ECP accounts of the contrast in (5), in (5.a) the trace t_i is a sister to the verb fix and hence lexically governs it. It thus satisfies the ECP. In (5.b) the trace t_j is an adjunct outside the governing domain of the verb (lexical government), and hence must be antecedent-governed to observe the ECP. The trace t_j in (5.b) cannot be locally bound by how, since there is another Wh-word in the embedded Comp. Hence it is not antecedent-governed nor lexically governed, and so violates the ECP.

In sentence (6.a) the trace is locally bound by how from the Comp, and satisfies the ECP. If how in sentence (6.b) undergoes LF movement, however, the trace of how cannot be locally bound by how, because the Comp has already had the index of what. So (6.b) violates the ECP.

The ECP account of Argument-adjunct asymmetries correctly derives the fact that adjuncts must move to a local Comp position prior to other Wh-phrases.

Although the ECP subsumes the Superiority Condition, it lacks explaining the contrast in (2), repeated just below:

- (10) a. [_{CP}who_i did [_{TP}you tell t_i [_{TP}PRO to read what]]]
 b. *[_{CP}what_i did [_{TP}you tell who [_{TP}PRO to read t_i]]]

In (2) each trace is lexically governed by the verb, and hence satisfies the ECP, but (2.b) is ungrammatical. It also has problems explaining argument-quasi-argument and argument-extraction asymmetries (see below).

Another problem is that we can hardly keep the ECP in the minimalist program, since it assumes (i) that some derivational principle like Subjacency and CED should apply differently to SS movement and LF movement, while principles are assumed to apply in the same way in the minimalist model, because there is no distinction between SS and LF; and (ii) that scope-bearing elements should undergo LF movement which is not motivated by morphological properties and hence disallowed in the minimalist program.

The third type of Wh-asymmetries are argument-quasi-argument asymmetries as in (11). Rizzi (1990) claims that the verb weigh in (11.a) assigns which box a referential θ role, and that the verb weigh in (11.b) assigns how much a nonreferential θ role, although both of them are understood as arguments of the verb weigh.

- (11) a. [_{CP}Which box did Bill wonder whether John weighed t_i]
 b. * [_{CP}how much did Bill wonder whether John weighed t_i]

Rizzi (1990) accounts for the contrast in (11) in terms of referentiality and relativized minimality.

- (12) (= (28) Rizzi 1990 p.86)

A referential index must be licensed by a referential θ role.

- (13) (= (29) Rizzi 1990 p.87)

X binds Y iff (i) X c-commands Y, and (ii) X and Y have the same referential index.

Rizzi (1990) claims that a referential index is legitimate only if it is associated to referential θ role, and the A'-dependencies must be expressed through binding relations which are also associated to referential θ roles. If no index is legitimate and no binding is available, for legitimacy the A'-dependency must resort to antecedent-government which is subject to relativized minimality.

- (14) (= (40) Rizzi 1990 p.92)

X antecedent-governs Y iff

- (i) X and Y are nondistinct

- (ii) X c-commands Y
- (iii) no barrier intervenes
- (iv) Relativized Minimality is respected.

(15) (= (15) Rizzi 1990 p.7)

Relativized Minimality

X α -governs Y only if there is no Z such that

- (i) Z is a typical potential α -governor for Y,
- (ii) Z c-commands Y and does not c-command X.

(16) (= (16) Rizzi 1990 p.7)

Z is a typical potential head governor for Y = Z
is a head m-commanding Y.

(17) (= (17) Rizzi 1990 p.7)

- a. Z is a typical potential antecedent governor
for Y, Y in an A-chain = Z is an A specifier c-
commanding Y.
- b. Z is a typical potential antecedent governor
for Y, Y in an A'-chain = Z is an A' specifier
c-commanding Y.
- c. Z is a typical potential antecedent governor
for Y, Y in an X_0 -chain = Z is a head c-
commanding Y.

For sentence (11), Rizzi claims that in (11.a) the verb weigh is an agentive verb which assigns a referential θ role to its object; in (11.b) the verb weigh is a stative verb which assigns a nonreferential θ role to its complement. In

(11.a) the trace t_i can be connected to which box through binding, since the index i is licensed by a referential θ role that the verb weigh assigns to its object. So the A'-dependency is legitimate in (11.a). On the other hand, in (11.b) no index is legitimate under (12), since a nonreferential θ role is assigned by the stative verb weigh. For the A'-dependency of how much and its trace, a chain of antecedent-government relations is the only option. But the operator how much fails to antecedent-govern its trace, since there is a closer intervening potential A'-governor, i.e. an operator in the spec of the embedded Comp, for it. Thus the A'-dependency is illegitimate in (11.b).

Following Rizzi (1990), we can generalize that only elements (or arguments) assigned a referential θ -role can be extracted from a Wh-island; all other elements (adjuncts and quasi-arguments) assigned a nonreferential θ role or assigned no θ role cannot be extracted from a Wh-island.

The relativized minimality analysis covers argument-adjunct asymmetries, argument-quasi-argument asymmetries and some adjunct symmetries in a unified way. Although the basic spirit of Relativized Minimality seems to be correct, the formulation cannot explain superiority effects like (2), and the contrast in (4) and in (18), since a referential θ role assigned argument is not subject to Relativized Minimality, and should have been extractable across a potential antecedent governor. It is also hard to maintain

the formulation in the minimalist model, since it refers to the distinction between referential and nonreferential θ roles which are not assumed to be formal features in the minimalist program.

Finally, there are other Wh-asymmetries which we may call argument-extraction asymmetries, exemplified in (18).

- (18) a. ?[_{CP}What_i did John wonder [_{CP}whether to fix t_i?]]
 b. ?[_{CP}What_i did John wonder [_{CP}how to fix t_i?]]
 c. *[_{CP}What_i did John wonder [_{CP}who fixed t_i?]]

As we have discussed before, arguments can usually be extracted from a Wh-island. In sentences (18) an argument what is extracted from a Wh-island, but it yields a severe deviance only in (18.c). Descriptively, arguments cannot be extracted through a Comp filled with an argument. In (18.c) the embedded Comp is filled with an argument who, through which what is extracted. On the contrary, the embedded Comp is filled with nonargument in (18.a) and (18.b).

In next section we will investigate Wh-asymmetries under economy considerations in the minimalist model.

4.2.2 Some Minimalist Analyses

Kitahara (1994) attempts to reduce four types of Wh-asymmetries (i.e. superiority effects, argument-adjunct,

argument-quasi-argument, argument-extraction asymmetries) to Chomsky's (1993, 1994, 1995) general economy principle, the Shortest Move (or the MLC). The Shortest Movement Requirement (SMR) is defined as in (19)-(20).

(19) (= (13) Kitahara 1994 p.61)

Shortest Movement Requirement (SMR)

Minimize the length of each feature-checking movement.

(20) (= (14) Kitahara 1994 p.61)

Shortest Feature-Checking Movement

Let X and Y be two nodes in a tree.

Let Z be the closest c-commander of X, bearing α -feature.

The movement of X to Y is the shortest α -checking movement of X iff Y and Z are in the same minimal domain.

(21) (= (15) Kitahara 1994 p.61)

Closest C-commander Bearing α -Feature

X is the closest c-commander of Y bearing α -feature iff

(i) X bears α -feature, and

(ii) X c-commands Y, and

(iii) no category bearing α -feature intervenes between X and Y.

(22) (= (16) Kitahara 1994 p.61)

Z intervenes between X and Y iff X c-commands Z
and Z c-commands Y.

(23) (= (17) Kitahara 1994 p.62)

C-command

X c-commands Y iff

- (i) neither X nor Y dominates the other, and
- (ii) a category immediately dominating X dominates Y.

(24) (= (18) Kitahara 1994 p.62)

Domain

the domain of CH $(\alpha_1, \dots, \alpha_n)$

= the set of categories contained in Max (α_1) ,
each member of which does not contain any α_i .

(25) (= (19) Kitahara 1994 p.62)

Minimal Domain

the minimal domain of CH $(\alpha_1, \dots, \alpha_n)$ =

the smallest subset K of the domain of CH

$(\alpha_1, \dots, \alpha_n)$ such that for any τ a member of the
domain of CH $(\alpha_1, \dots, \alpha_n)$, some β member of K
dominates τ .

To illustrate how the SMR works, suppose the following
configuration:

(26) (= (22) Kitahara 1994 p.63)

Y ... Z_α ... X

The movement of X to Y satisfies the SMR only if (i) X β -checks with Y where α not equal β , or (ii) X α -checks with Y where (a.) Z is the closest c-commander of X, bearing α -feature, and (b.) Y and Z are in the same minimal domain.

Now consider the superiority effect (1), repeated in (27), and its derivation in (28).

- (27) a. John wonders [_{CP}who_i [_{TP}t_i bought what]]
 b. *John wonders [_{CP}what_i [_{TP}who bought t_i]]
- (28) a. [_{CP}C_{OP} [_{TP}who bought what]]
 b. [_{CP}who_i [_{C'}C [_{TP}t_i bought what]]]
 c. John wonders [_{CP}who_i [_{C'}C [_{TP}t_i bought what]]]

Suppose the computations have constructed (28.a). The strong +Wh of the Comp attracts who; if it attracts what it would violate the SMR, which is the case of (27.b). After that the computations map (28.b) to (28.c).

The SMR also correctly explains the superiority effect in (2) exactly in the same manner as in (1), which was the problem with the ECP account.

But the SMR makes the contrast in (4) remain unsolved, which is a critical counterexample of the superiority account. We repeat it in (29).

- (29) a. *I wonder what who bought t
 b. Who wonders what who bought t

At the point of constructing the embedded CP in (29), the +Wh feature of the Comp must attract who, not what, observing the SMR, since who is the closer to the Comp in the minimal domain for +Wh checking than what.

Kitahara (1994) accounts for argument-adjunct asymmetries in terms of Chain Formation Requirement (CFR) in addition to the SMR. The CFR is defined as below:

(30) (= (18) Kitahara 1994 p.120)

Chain-Formation Requirement (CFR)

An application of Target α T_i yields more than one chain only if T_i is violation-free.

Consider (5), repeated in (31). Suppose the computations construct (32) and (33) for (31.a) and (31.b), respectively.

(31) a. ?_{[CP what_i did John wonder [CP whether to fix t_i]]}

b. *_{[CP How_j did John wonder [CP whether to fix the car t_j]]}

(32) a. [_{CP}did [_{TP}John wonder [_{CP}whether PRO to fix what]]]

b. [_{CP}what_i [_C.did [_{TP}John wonder [_{CP}whether PRO to fix t_i]]]]]

c. [_{CP}what_i [_C.did [_{TP}John wonder [_{CP}whether PRO to [_{AgRP-O}t'_i fix t_i]]]]]]]

- (33) a. [_{CP}Did [_{TP}John wonder [_{CP}whether PRO to fix a car
how]]]
- b. [_{CP}how_i [_C.did [_{TP}John wonder [_{CP}whether PRO to
fix the car t_i]]]]]

Consider (31.b) first. Suppose that the computations have constructed (33.a). At this point the strong +Wh feature of the matrix Comp attracts how, which violates the SMR, since whether in the embedded Comp is closer to the matrix Comp. He claims that the violation of the SMR yields no chain formation of how and its trace in (31.b), and that its LF representation in (34) is not legitimate because violation of the SMR disallows how and its trace to be in a chain under the CFR, and the operator how undergoes vacuous binding.

- (34) [_{CP}how_i [_C.did [_{TP}John wonder [_{CP}whether PRO to fix
the car t_i]]]]]

Thus the sentence (31.b) violates the SMR and LF legitimacy, which yields severe deviance.

Consider (31.a) now. Suppose that (32.a) is derived. At this point the strong +Wh feature of the Comp attracts what, violating the SMR and thereby yielding no chain of what and its trace. In this case, however, what needs to check its Case at LF, and so its trace (or copy) undergoes

LF movement as shown in (32.c). This single violation-free application of movement can yield two chains: an operator-variable chain (what_i , t'_i) and an argument chain (t'_i , t_i). So the LF representation in (35) satisfies the LF legitimacy.

- (35) [_{CP} what_i [_Cdid [_{TP}John wonder [_{CP}whether PRO to [_{AgRP-O}
 t'_i fix t_i]]]]]

Thus (31.b) violates the SMR and LF legitimacy, but (31.a) violates only the SMR, which results in the contrast in (31).

However, this analysis cannot explain the contrast in (18), repeated in (36):

- (36) a. ?[_{CP} What_i did John wonder [_{CP}whether to fix t_i ?]]
 b. ?[_{CP} What_i did John wonder [_{CP}how to fix t_i ?]]
 c. *[_{CP} What_i did John wonder [_{CP}who fixed t_i ?]]

Consider (36.c). Suppose the computations have constructed (37.a). At this point the embedded Comp attracts who, observing the SMR, in mapping (37.a) to (37.b). The computations further construct (37.c). At this point the matrix Comp attracts what, violating the SMR. At LF, however, there is a single violation-free movement which raises the trace of what, yielding two chains, the operator-

variable and argument chain. The LF representation is shown in (37.e).

- (37) a. [_{CP}C [_{TP}who fixed what]]
 b. [_{CP}who_i [_C·C [_{TP}t_i fixed what]]]
 c. [_{CP}did [_{TP}John wonder [_{CP}who_i [_C·C [_{TP}t_i fixed what]]]]]
 d. [_{CP}what_j [_C·did [_{TP}John wonder [_{CP}who_i [_C·C [_{TP}t_i fixed t_j]]]]]]]
 e. [_{CP}what_j [_C·did [_{TP}John wonder [_{CP}who_i [_C·C [_{TP}t_i [_{AgRP-o} t'_j fixed t_j]]]]]]]]]

Then (36.a-b) and (36.c) violate only the SMR, but only (36.c) is severely deviant, but (36.a-b) are marginally deviant.

In next section we offer an alternative unified analysis of Wh-asymmetries under multiple feature attraction and the Minimal Link Condition.

4.3 A Unified Analysis of Wh-asymmetries

4.3.1 Feature Specifications of Wh-words and [+Wh] Comps

For a Wh-interrogative the minimalist program assumes that a strong feature Op₀ of a Comp attracts a feature Op₀

of a Wh-phrase, as shown in (38).

- (38) a. [_{CP}C(did)_{Op-Q} [_{TP}John buy what_{Op-Q}]]
 b. [_{CP}what(Op_Q)_i [_{C'}C(did)_{Op-Q} [_{TP}John buy t_i]]]
 c. Spell-out

If Spell-out applies before (38.b), it would crash at PF; if (38.b) never happens, it would crash at PF and LF: a strong feature is not interpretable at PF and LF; and the operator what_{Op-Q} also undergoes vacuous binding, which is illegitimate at LF. Some condition on vacuous binding is necessary for LF legitimacy for an independent reason.²⁹

Consider (39). In (39.a) the Comp does not have any Op_Q feature, and a Wh-phrase what with Op_Q stays in situ. Sentence (39.b) has a Comp with Op_Q, and what merges to the Comp rather than it moves to the Comp. To explain the ungrammaticality of (39), we presumably need (40) for LF legitimacy.

- (39) a. *John bought what.
 b. *What did John fix a car.

(40) LF Legitimacy

An operator must nonvacuously binds a variable,
 and a variable must be bound by an operator.

²⁹See also: Lasnik and Saito (1992).

In (39) what_{Op-Q} violates LF Legitimacy: in (39.a) it is not in a Comp and it binds nothing; in (39.b) what binds nothing, although it occurs in a Comp.

One important thing to note is that LF Legitimacy does not motivate movement at all; rather, morphologically-driven movement results in the LF Legitimacy.

On the other hand, English does not allow a Comp to be doubly filled. In English multiple questions only one Wh-phrase moves to the Comp with Op_Q, as shown in (41). It indicates that in English a Comp parametrically has only one strong Op_Q feature.

- (41) a. Who did Bill persuade to buy what?
 b. *What Who did Bill persuade to buy

In (41.a), on the other hand, who moves to the Comp and binds its trace, which satisfies LF Legitimacy; but Wh-in-situ what does not observe LF Legitimacy, since it occurs in situ, not in a Comp, and binds nothing.

To explain the grammaticality of (41.a), we may not say that like the ECP account, the Comp has one more weak Op_Q feature and it attracts what at LF. This approach cannot explain the contrast in (42).

- (42) a. *Who did John leave after he met t?
 b. Who left after he met who?

The adjunct clause is known to be an extraction island. The ungrammaticality of (42.a) is due to extraction of who from the extraction island. If a weak Op_Q feature of the Comp can attract who from the adjunct clause at LF in (42.b), how can we justify the LF extractability without any distinction between S-structure and LF?

For Wh-in-situ Tsai (1994) argues that Chinese Wh-phrases have a variable, but do not have an operator Op_Q , as shown in (43).

(43) N_0
 / \
 Wh ind.x

The variable ind.x undergoes some type of binding for Wh-dependencies rather than movement. Under this assumption we can explain the contrast in (42). In (42.b) the Wh-in-situ who has only a variable without Op_Q . Then who does not undergo movement for Wh-dependency. Wh-in-situ will take binding (or linking) for Wh-dependency.

If this approach is correct, then English has two types of Wh-words. One is that a Wh-word has only an operator Op_Q ; the other that a Wh-word has only a pronominal variable. (cf. Chierchia (1991), Hornstein (1995)) Their differences can be represented in terms of feature specification:

- (44) a. Wh-operator: {Op, Q}.
 b. a Pronominal Wh-word: {D_{Pro}}.

The operator Wh-word forms a Wh-dependency by movement, and a variable Wh-word forms a Wh-dependency by linking (in Higginbotham's (1983) and Hornstein's (1995) terms).

Furthermore, Reinhart (1993), Tsai (1994) and others distinguish Wh-NPs from Wh-adverbials. They claim that Wh-adverbials do not have an indefinite variable, and Wh-NPs do. In other words, Wh-adverbials have only an operator form. This assumption indicates that Wh-NPs can form Wh-dependencies either by movement or by binding, since they can have an operator or a variable feature, while Wh-adverbials can form Wh-dependencies only by movement leaving a trace as a variable behind³⁰.

It is also important here to note that Wh-NPs are categorically NPs, forming a DP with a D feature, while Wh-adverbials are categorically adverbials, lacking a D feature. We represent these as the following structures:

- (45) a. Wh-NP: [_{DP}D [_{NP}Wh]]
 b. Wh-Adverbial: [_{AdvP}Wh]

To sum up, we minimally represent Wh-words in terms of feature specification as in the following:

³⁰See Tsai (1994) for more consequences.

- (46) a. Wh-NP: {D, Op, Q}
 b. Wh-in-situ: {D_{Pro}}
 c. Wh-adverbials: {Adv, Op, Q}

Now consider some features of a Comp. It is known that a Comp has a Op_Q-bearing feature for Wh-questions. In addition, the Comp has a D feature, as shown below:

- (47) a. That John left was pleasing.
 b. Whether John went was important.

In English a tense has a strong D feature. It attracts a D feature overtly for PF convergence. This is the case of VP-internal subject raising to the Spec of TP. In (47) the CP clause occupies the Spec of TP, and should have a D feature to check the strong D feature of the tense. In many other cases, too, we can see that CPs have distribution very similar to DPs.

4.3.2 Multiple Feature Attraction

Under the minimalist assumptions, the operation Move must be driven by the requirement that some morphological feature F must be checked. If a target attracts a feature F, Attract-F/Move-F automatically carries FF(F). If Move -F is triggered by PF, it pied-pipes its full category (for PF

convergence). (Chomsky (1995))

If we carefully look into Attract-F, however, we need to make sure which F attracts which F. Consider several cases here.

Suppose that we have two features F1 and F2, one functional category X, and two lexical categories Y and Z in the lexicon. Then we have several possible choices of feature selections. Consider Case 1 in (48) where the symbol * means that the feature is uninterpretable:

(48) Case 1: Let $FF(X) = \{ *F1 \}$ and $FF(Y) = \{ F1 \}$.

For a derivation to converge, $FF(X) = \{ *F1 \}$ must attract $FF(Y) = \{ F1 \}$. Then $FF(X)$ and $FF(Y)$ can be in a checking relation, since both have the same feature F1.

Consider Case 2 in (49).

(49) Case 2: Let $FF(X) = \{ *F1, *F2 \}$ and $FF(Y) = \{ F1, F2 \}$.

In this case, too, $FF(X) = \{ *F1, *F2 \}$ must attract $FF(Y)$ for convergence, and Attract-F is successful and the derivation converges, since $FF(x) = \{ *F1, *F2 \}$ can enter into a checking relation with $FF(y) = \{ F1, F2 \}$.

Consider Case 3 in (50):

(50) Case 3: Let $FF(X) = \{ *F1, *F2 \}$ and $FF(Y) = \{ F1 \}$ and

$$FF(Z) = \{F2\}.$$

For convergence, *F1 and *F2 of FF(x) each attracts its corresponding feature. In this case FF(X) must attract a feature F twice: first, the *F1 of FF(X) attracts FF(Y)={F1}, and second attracts FF(Z)={F2}, where the order of Attract-F does not matter here.

For Case 3 here we can think about another possibility of Attract-F. Different from the previous assumption that *F1 and *F2 of FF(x) each independently attracts FF(Y) and FF(Z), a pair <*F1,*F2> of FF(X) triggers Attract-F.

If we take this option for Case 3, the derivation would crash, since neither FF(Y) nor FF(Z) has a pair <F1,F2> and Attract-F fails.

If FF(Z) were to have {F1,F2}, then it would converge, since FF(X)=<*F1,*F2> could attract it.

In the minimalist program, the universal principles are assumed to be invariant and common to all human language faculties, and parameters (or options) are assumed to be "restricted to functional elements and general properties of the lexicon" (Chomsky 1994 p.4) If we parametrize features to attract in this way, we would explain some language variations of movement. We propose the following parametrization for Attract-F:

(51) Attract F where the number of F and the type of F

can be parametrized language to language.

This approach completely fits the minimalist assumption about parametrization. More specifically, English has the following parameter for a Comp:

(52) F of a Comp attracts F where F is Op_0 or $\langle D, Op_0 \rangle$.

In the subsequent sections we will discuss the consequences, investigating Wh-asymmetries.

Finally, consider Case 4 in (53) for Attract-F and MLC:

(53) Case 4: Let $FF(X) = \{ *F1, *F2 \}$ and $FF(Y) = \{ F1, F2 \}$ and $FF(Z) = \{ F1, F2 \}$.

$FF(X)$ attracts either $FF(Y)$ or $FF(Z)$ exclusively at this time, since both $FF(Y)$ and $FF(Z)$ have $\{ F1, F2 \}$ and can be in a checking relation with $\{ *F1, *F2 \}$ of $FF(X)$. In this case, however, the order is relevant. If $FF(Y)$ is closer to $FF(X)$ than $FF(Z)$, Attract-F attracts $FF(Y)$ but cannot attract $FF(Z)$ because of the Minimal Link Condition; if $FF(Y)$ is in equidistance with $FF(Z)$, it attracts either of them; otherwise it attracts $FF(Z)$.

Following Chomsky (1995), we can define the Minimal Link Condition as follows:

(54) (= (110) Chomsky 1995 p.311)

Minimal Link Condition

K attracts α only if there is no β , β closer to K than α , such that K attracts β .

(55) (Chomsky 1995 p.358)

β is closer to the target K than α if β c-commands α .

In next section let us consider the Minimal Link Condition and multiple feature attraction can explain Wh-asymmetries.

4.3.3 Analysis

4.3.3.1 Some Basic Assumptions

In previous sections we have elaborated some distinction among Wh-words in terms of feature specifications. Wh-Nps minimally have $\{D, OP_Q\}$, Wh-adverbials $\{Adv, OP_Q\}$, and Wh-in-situ $\{D_{pro}\}$.

We have also seen that a Comp has $\{D, OP_Q\}$ for Wh-questions. We also assume that a comp has only one OP_Q in English to disallow a doubly-filled Comp. Furthermore, the Comp parametrically attracts Wh-phrases with $\{OP_Q\}$ or $\langle D, OP_Q \rangle$.

Consider the following three configurations.

- (56) [_{CP}C_{D, Op-Q} [_{TP}... WhP_{D, Op-Q}]]
- a. the Op_Q of FF(C) attracts FF(WHP).
 - b. A pair of features <D, Op_Q> of FF(C) attracts FF(WhP).
- (57) [_{CP}C_{D, Op-Q} [_{TP}... WhP_{ADV, Op-Q}]]
- a. the Op_Q of FF(C) attracts FF(WHP).
 - b. A pair of features <D, Op_Q> of FF(C) cannot attract FF(WhP).
- (58) [_{CP}C_{D, Op-Q} [_{TP}... WhP_{DPro}]]
- a. the Op_Q of FF(C) cannot attract FF(WHP).
 - b. A pair of features <D, Op_Q> of FF(C) cannot attract FF(WhP).

In (56) the Comp can attract the Wh-NP with Op_Q or <D, Op_Q>, since the Wh-NP also has those features which can be in a checking relation with the features of the Comp. This configuration represents a sentence like (59.a).

On the other hand, in (57) the Op_Q can attract FF(WhP), while a pair <D, Op_Q> of the Comp cannot attract the Wh-Adv, since the Wh-adverbial has no D feature in its formal features {Adv, Op_Q}. Hence the movement of Wh-adverbials is always triggered by the Op_Q of the Comp for Wh-questions. This configuration represents (59.b).

Wh-in-situ constructions can be represented as in (58). In (58) FF(Comp) cannot attract FF(WHP) with either Op_Q or <D, Op_Q>, since FF(WHP) does not have any Op feature.

- (59) a. what did Mary eat?
 b. How did Mary eat pizza?

We also accept Chomsky's (1993) assumption that there is no QR-like LF movement. Hence Wh-in-situ does not undergo QR-like LF movement to the Spec of CP to simply take scope³¹.

Under these hypotheses we will attempt to give a unified analysis of Wh-asymmetries in next section.

4.3.3.2 Argument-Adjunct Asymmetries

Let us start with argument-adjunct asymmetries, as shown in (60).

- (60) a. What did John wonder how to fix?
 b. *How did John wonder what to fix?

Consider the derivation of (60.a) first, which can be described as in (61):

- (61) a. [_{CP}C_{Op-Q} [_{TP}PRO to [_{VP}[_{VP}fix what_{D,Op-Q}] how_{Op-Q}]]]
 b. [_{CP}how_{Op-Q-i} [_CC_{Op-Q} [_{TP}PRO to [_{VP}[_{VP}fix what_{D,Op-Q}]]]

³¹This does not mean that there is no covert Wh-movement. Presumably, Wh-in-situ can undergo LF movement if Wh-in-situ has an Op₀ feature, and a Comp also has a non-strong uninterpretable Op₀ feature.

$t_i]]]]$

- c. $[_{CP}C(\text{did})_{\langle D, Op-Q \rangle} \text{John wonder } [_{CP}\text{how}_{Op-Q-i} [_{C'}C_{Op-Q}$
 $[_{TP}PRO \text{ to } [_{VP}[_{VP}\text{fix what}_{\langle D, Op-Q \rangle}] t_i]]]]]$
- d. $[_{CP}\text{what}_{\langle D, Op-Q \rangle-j} [_{C'}C(\text{did})_{\langle D, Op-Q \rangle} \text{John wonder}$
 $[_{CP}\text{how}_{Op-Q-i} [_{C'}C_{Op-Q} [_{TP}PRO \text{ to } [_{VP}[_{VP}\text{fix } t'_j] t_i]]]]]]]$

Suppose that the computational system has constructed (61.a) for (60.a). At this point the computational system can take two choices for further derivation: one is that FF(C) attracts a Wh-phrase with Op_Q ; the other choice is that it attracts a Wh-phrase with $\langle D, Op_Q \rangle$. Putting aside the second choice for a moment, let us focus on the first choice here. The Op_Q of the Comp can potentially attract FF(how) or FF(what), which maps (61.a) to (61.b) or (62), respectively:

- (62) $*[_{CP}\text{what}_{\langle D, Op-Q \rangle-i} [_{C'}C_{Op-Q} [_{TP}PRO \text{ to } [_{VP}[_{VP}\text{fix what}_i]$
 $\text{how}_{Op-Q}]]]]] \rightarrow \text{*the MLC}$

If we compare (61.b) and (62) at this point under economy considerations, the former is more economical than the latter, since FF(how) is structurally closer to the attractor than FF(what). The derivation (62) violates the MLC. So the MLC picks (61.b) for an optimal derivation at this point.

After that the computational system is supposed to

construct (61.c). At this point the matrix Comp can potentially attract a Wh-phrase in three ways: (i) the Op_0 attracts FF(How) in the embedded Comp, which maps (61.c) to (63); (ii) the Op_0 attracts FF(what), which maps (61.c) to (64); and (iii) features $\langle D, Op_0 \rangle$ attracts FF(what), which maps (61.c) to (61.d).

(63) $*[_{CP} how_{Op-Q-i} [_{C'} C(did)_{Op-Q} John\ wonder\ [_{CP} t'_i\ [_{C'} C_{Op-Q} [_{TP} PRO\ to\ [_{VP} [_{VP} fix\ what_{\langle D, Op-Q \rangle}] t_i]]]]] \rightarrow *the\ Last\ Resort\ Condition$

(64) $*[_{CP} what_{\langle D, Op-Q \rangle-j} [_{C'} C(did)_{Op-Q} John\ wonder\ [_{CP} how_{Op-Q-i} [_{C'} C_{Op-Q} [_{TP} PRO\ to\ [_{VP} [_{VP} fix\ t'_j] t_i]]]]] \rightarrow *the\ MLC$

If we compare (61.d), (63) and (64) under economy considerations, the derivation (61.d) is the most economical derivation: (61.d) observes all three derivational economy conditions, while (63) violates the Last Resort Condition, and (64) violates the MLC. That is, at the point of (61.c), the Op_0 cannot attract FF(how) in the embedded Comp, since the chain (how_i, t_i) has already satisfied bare output conditions; if so, it would violate the Last Resort Condition, as in (63). If the Op_0 attracts FF(what), it would violate the MLC, as in (64), since there is an intervening category with Op_0 which is closer to the matrix Comp than FF(what). If the Comp attracts FF(what), Attract-F then observes the MLC, since the intervening How does not

have features $\langle D, Op_Q \rangle$. FF(how) has features $\{\text{Adv}, Op_Q\}$. Furthermore, by this attraction FF(what) can also be legitimate at LF, escaping from vacuous binding. Hence the computational system maps (61.c) to (61.d), being an optimal derivation.

Now consider (60.b). Suppose the computations have constructed (65.a). At this point, the Comp attracts a Wh-phrase with either Op_Q or $\langle D, Op_Q \rangle$. So far we have considered the first choice above for (60.a). If the Comp attracts what with a pair $\langle D, Op_Q \rangle$, it observes the MLC, since the intervening how does not have this pair. Then Attract-F successfully maps (65.a) to (65.b). After that the computations further construct (65.c). At this point the Comp cannot take $\langle D, Op_Q \rangle$ to attract how, since how does not have that pair. So it should attract how with Op_Q . But this violates the MLC, since the intervening what in the spec of the embedded CP has Op_Q feature. If the matrix Comp attracts FF(what) in the embedded CP, then it would violate the Last Resort Condition and LF Legitimacy, since what has already formed a Wh-dependency, and how vacuously binds itself in situ.

- (65) a. $[_{CP}C_{\langle D, Op_Q \rangle} [_{TP}PRO \text{ to fix } what_{\langle D, Op_Q \rangle} how_{Op_Q}]]$
 b. $[_{CP}what_{\langle D, Op_Q \rangle-1} [_C C_{\langle D, Op_Q \rangle} [_{TP}PRO \text{ to fix } t_1 how_{Op_Q}]]]$
 c. $[_{CP}C(\text{did})_{Op_Q} \text{ John wonder } [_{CP}what_{\langle D, Op_Q \rangle-1} [_C C_{\langle D, Op_Q \rangle}]]]$

- [_{TP}PRO to fix t_i how_{Op-Q}]]]]
- d. * [_{CP}how_{Op-Q-j} [_{C'}C(did)_{Op-Q} John wonder [_{CP}what_{{D, Op-Q}-}
i [{C'}C_{D, Op-Q} [_{TP}PRO to fix t_i t_j]]]]]]

Thus we successfully explain argument-adjunct asymmetries under Attract-F and the MLC. Extraction of an argument out of a Wh-island does not violate derivational economy, but extraction of an adjunct out of a Wh-island violates the MLC.

4.3.3.3 Argument Extraction Asymmetries

Now consider (66) exhibiting argument extraction asymmetries.

- (66) a. What did John wonder how to fix?
 b. *What did John wonder who bought?

Suppose that the computations have constructed (67.a) for (66.b). At this point, the Comp attracts who with Op_Q. If it instead attracts what, it would violate the MLC, because who is intervening between the attractor Comp and what, and the feature of who is the same as the feature to attract. It maps (67.a) to (67.b). After that the computations further map (67.b) to (67.c). At this point the matrix Comp attempts to attract what with either Op_Q or

<D, Op_Q>, but it would violate the MLC, since the intervening who in the embedded Comp has those features.

- (67) a. [_{CP}C_{D, Op-Q} [_{TP}who_{D, Op-Q} bought what_{D, Op-Q}]]
 b. [_{CP}who_{{D, Op-Q}-i} [_{C'}C_{D, Op-Q} [_{TP}t_i bought what_{D, Op-Q}]]]
 c. [_{CP}C(did)_{D, Op-Q} John wonder [_{CP}who_{{D, Op-Q}-i} [_{C'}C_{D, Op-Q} [_{TP}t_i bought what_{D, Op-Q}]]]]]
 d. * [_{CP}what_{{D, Op-Q}-j} [_{C'}C(did)_{D, Op-Q} John wonder [_{CP}who_{{D, Op-Q}-i} [_{C'}C_{D, Op-Q} [_{TP}t_i bought t_j]]]]]]]

If an argument is extracted out of a Wh-island across another Wh-argument, it also violates the MLC.

4.3.3.4 Argument-Quasi-Argument Asymmetries

Sentence (68) is an example of argument-quasi-argument asymmetries.

- (68) a. Which box did Bill wonder whether John weighed
 t_i
 b. *How much did Bill wonder whether John weighed
 t_i

Quasi-arguments behave like adjuncts in some cases, although they receive a θ -role (and presumably Case, too) as arguments. First of all, they behave like Wh-adjuncts in

extraction out of a Wh-island, as shown in (68). Arguments can be extracted out of a Wh-island, yielding mild deviance, while adjuncts cannot be extracted; if so, it would yield severe deviance. If quasi-arguments are extracted out of a Wh-island as in (68.b), it will yield severe deviance like ordinary adjunct extraction.

Second, sentence (68.a) can be passivized, raising which box to the Spec of TP, as in (69.a), but (68.b) cannot, as shown in (69.b):

- (69) a. Which box was weighed by John?
 b. *How much was weighed by John?

In passivization DP-movement takes place for two reasons: one is that the Case feature of the complement NP and T should be checked for convergence; the other that the strong D feature of T should be checked by a DP. Ordinary DP arguments have Case and D features, and so can be raised to the Spec of TP in passivization. If this is correct, presumably quasi-arguments may lack a D feature, (although it might have a N-feature). If it were to have a D feature, it would be raised to the Spec of TP in passivization, and check the strong D feature.

The assumption that a quasi-argument how much has the feature specification {N, Op₀} also gives us some explanation of the fact that, like Wh-adjuncts, quasi-

arguments cannot be extracted out of a Wh-island.

For (68.a) the computational system constructs the derivation as in (70).

- (70) a. [_{CP}Did+C_{D, Op-Q}] [_{TP}Bill wonder [_{CP}whether_{Op-Q} John weighed which box]]]
 b. [_{CP}[Which box]_i] [_CDid+C_{D, Op-Q}] [_{TP}Bill wonder [_{CP}whether_{Op-Q} John weighed t_i]]]]

Suppose that the computational system has constructed (70.a). At this point a feature Op_Q of the matrix C' cannot attract which box, because there is an intervening category with Op_Q; if so, it would violate the MLC.

But features <D, Op_Q> can attract it without violating the MLC at all, because the intervening category whether is assumed to lack a D feature. The computational system thus generates (70.b), which converges.

For (68.b), on the other hand, the computational system constructs the derivation given in (71):

- (71) a. [_{CP}Did+C_{Op-Q}] [_{TP}Bill wonder [_{CP}whether_{Op-Q} John weighed how much]]]
 b. * [_{CP}[How much]_i] [_CDid+C_{Op-Q}] [_{TP}Bill wonder [_{CP}whether_{Op-Q} John weighed t_i]]]]

Suppose that the computational system has constructed

(71.a). At this point a feature Op_0 of the matrix C fails to attract how much, because there is an intervening category with Op_0 which is closer to the matrix C. Features $\langle D, Op_0 \rangle$ also fails to attract how much, because the quasi argument how much lacks a D feature. So quasi-argument cannot be extracted out of a Wh-island.

The correlation between Passivizability and extractability out of a Wh-island seem to be more plausible than a correlation between referentiality and extractability out of a Wh-island. As we have discussed in section 4.2, Rizzi (1990) argues that an element assigned a nonreferential θ role cannot be extracted out of a Wh-island, since an intervening Wh-phrase in the embedded Comp blocks antecedent-government under relativized minimality.

In (72) laps seems to be assigned a nonreferential θ role. In spite of that, like an ordinary argument, it can be extracted out of a Wh-island as in (73).

(72) John swam laps.

(73) ?How many laps did Bill wonder whether John swam
t?

Interestingly, (72) can also be passivized as in (74):

(74) Laps were swam by John.

In other words, although laps in (72) is assigned a nonreferential θ -role, it can be extracted out of a Wh-island if it can be passivized (i.e. it has a D feature).

Thus we can also reduce argument-quasi-argument asymmetries to the asymmetry of a D feature availability among Wh-phrases³².

4.3.3.5 Superiority Effects and Some Residues

The MLC analysis can also explain some superiority effects. Consider the following examples:

- (75) a. Who bought what?
 b. *What did who buy?
- (76) a. Who did you tell to read what?
 b. *What did you tell who to read?

The contrasts in (75) and (76) could be explained by the Superiority Condition, and can also be explained by the MLC under economy considerations. Take the following derivations for (75):

(77) [_{CP}C_{Op-Q} [_{TP}who(_{Op_Q}) bought what(_{Op_Q})]]

(78) [_{CP}who(_{Op_Q})_i [_{C'}C_{Op-Q} [_{TP}t_i bought what(_{Op_Q})]]]

³²Probably referentiality/nonreferentiality and specificity/nonspecificity are closely related to some properties of a D feature and non-D feature.

(79) *_{[CP}what(Op_Q)_i [_C·C_{Op-Q} [_{TP}who(Op_Q) bought t_i]]]

Suppose that the computational system has constructed (77). At this point the strong matrix C_{Op-Q} should attract a Wh-phrase. There are two choices here: it attracts either who or what exclusively. If it attracts who as in (78), it will observe the MLC, and if it attracts what as in (79), it will violate the MLC, since who is closer to the target than what.

We can explain the contrast in (76) in the same way. The computational system constructs (80); then the matrix C_{Op-Q} should attract who rather than what, since who is closer to the target than what.

(80) [_{CP}C(did)_{Op-Q} [_{TP}you tell who(Op_Q) [_{CP}Pro to read what(Op_Q)]]]

(81) [_{CP}who(Op_{Qi}) [_C·C(did)_{Op-Q} [_{TP}you tell t_i [_{CP}Pro to read what(Op_Q)]]]]]

(82) *_{[CP}what(Op_Q)_i [_C·C(did)_{Op-Q} [_{TP}you tell who(Op_Q) [_{CP}Pro to read t_i]]]]]

However, this is not sufficient to explain the contrasts in (75) and (76). If we take a look at derivations, (78) and (81), which observe the MLC, they would violate LF Legitimacy, since what(Op_Q) remains in situ and undergoes vacuous binding.

We have assumed that if a Wh-word has an operator and a Q feature, it must raise to the Spec of CP; otherwise it would violate LF Legitimacy for Wh-dependencies at LF. In the minimalist program, however, a Wh-word cannot move to the spec of CP in order to simply observe LF Legitimacy, if this movement is not driven by morphological feature checking. In other words, a Wh-word with an operator feature and a Q feature must move to the spec of CP to check some feature of a Comp, and the result of this movement will observe the principle of LF Legitimacy.

If this is true, only one Wh-word must have an operator feature in English multiple questions, since in English a [+Wh] Comp parametrically has only one operator feature. In English, on the other hand, a wh-in-situ should contain a pronominal variable in Hornstein's (1995) terms. Hence a Wh-in-situ has the feature specification $\{D_{Pro}\}$. Further, this variable must be coindexed or linked with another Wh-element.

Keeping this in mind, let us consider (75) again, repeated in (83).

(83) Who bought what?

For (83) we may have the following possible representations:

- (84) a. $*[_{CP}who(Pron)_i [_{TP}t_i \text{ bought } what(Pron)]]$
 b. $*[_{CP}who(Pron)_i [_{TP}t_i \text{ bought } what(OP)]]$
 c. $*[_{CP}who(OP)_i [_{TP}t_i \text{ bought } what(Op)]]$
 d. $*[_{CP}who(OP)_i [_{TP}t_i \text{ bought } what(Pron)_j]]$
 e. $[_{CP}who(OP)_i [_{TP}t_i \text{ bought } what(Pron)_i]]$

In the representations of (84), first of all, (84.a) and (84.b) both violate Attract-F, since in the representations who does not have any Op feature to be attracted by the Comp; (84.c) also violates LF Legitimacy, since the operator remains in situ; what_{Op} in (84.c) cannot move to the Spec of C at LF, since there is no Op feature in the Comp which has already checked with the Op feature of who. (84.d) is also illegitimate at LF, since the Wh-in-situ what(Pron)_j is not bound by another Wh-element. On the other hand, (84.e) can converge, since what(Pron)_i is linked to who, and no principle blocks this linking.

Now consider (85).

- (85) *what did who buy?

We can also have the following representations for (85):

- (86) a. $*[_{CP}what(Pron)_i [_{TP}who(Pron) \text{ buy } t_i]]$
 b. $*[_{CP}what(Pron)_i [_{TP}who(Op) \text{ buy } t_i]]$

- c. $*[{}_{CP}\text{what}(\text{Op})_i [{}_{TP}\text{who}(\text{Op}) \text{ buy } t_i]]$
- d. $*[{}_{CP}\text{what}(\text{Op})_i [{}_{TP}\text{who}(\text{Pron})_j \text{ buy } t_i]]$
- e. $*[{}_{CP}\text{what}(\text{Op})_i [{}_{TP}\text{who}(\text{Pron})_i \text{ buy } t_i]]$

The representations (86.a) and (86.b) violate Attract-F, as discussed before. (86.c) violates the LF Legitimacy, since who(Op_Q) remains in situ and has vacuous binding, and also violates the MLC, since what crosses who with {D, Op_Q}. The representation (86.d) is also illegitimate at LF, since who(Pron)_j is not bound by another Wh-element. What will then happen if what is linked to who as in (86.e)? This will violate weak crossover, following Chierchia (1991) and Hornstein (1995). We can define the weak crossover as follows:

(87) (= (7) Hornstein (1995) p.100)

A pronoun cannot be linked to a variable on its right
 $*\text{Op} \dots \text{pronoun}_i \dots \text{variable}_i \dots$

The LF condition (87) forbids the LF structure in (88.a), while it permits (88.b).

- (88) a. $*\text{Who}_i \text{ does his}_i \text{ mother love } t_i?$
- b. $\text{Who}_i \text{ does his}_j \text{ mother love } t_i?$

If we compare the LF representation (86.e) with (88.a),

we can see parallelism between the two.

We can see more data for such parallelism below:

- (89) a. *What_i do you expect its_i author to publish t_i
 b. *What_i do you expect who_i to publish t_i
- (90) a. *What_i did you give its_i owner t_i
 b. *What_i did you give who_i t
- (91) a. *Who_i did you send his_i book to t_i?
 b. *Who_i did you send what_i to t_i?

The LF representations of (89.b), (90.b) and (91.b) can be illustrated in (92).

- (92) a. *What_i do you expect [_{who}pron_i D] to publish t_i
 b. *What_i did you give [_{who}pron_i D] t_i
 c. *Who_i did you send [_{what}pron_i D] to t_i?

Under the WCO analysis, we can also explain the following contrast:

- (93) a. *John wonders what_i who_i bought t_i.
 b. *Who_i t_i wonders what_j who_j bought t_j.
 c. Who_i t_i wonders what_j who_i bought t_j.

If we assume that who in the embedded clause is a pronominal form, not an operator form, the representations

in (93) observe the MLC. Although who is a pronominal form in (93.a) and in (93.b), both examples violate the WCO. On the other hand, (93.c) observes the MLC and the WCO. We expect (93.c) to be grammatical.

4.4 Further Consequences: Some Adjunct Symmetries

Rizzi (1990) observes that there are some symmetries among Huang's (1982) argument-adjunct asymmetries, Obenauer's (1984) pseudo-opacity effects, and Ross's (1983) inner islands, and attempts to treat them in an unified way under relativized minimality. As we have discussed before, adjuncts cannot be extracted from a Wh-island:

- (94) a. What did John wonder how to fix?
 b. *How did John wonder what to fix?

Obenauer (1984) observes that in French a VP-initial quantifier-bearing adjunct blocks extraction of some VP-internal categories. In French a Wh-word combien (how many/much) can be extracted from the spec of the NP, or pied-pipes the whole NP, as shown in (95).

- (95) (= (27) Rizzi 1990 p.12)
 a. [Combien de livres] a-t-il consultés t
 how-many of books did he consult

- b. Combien a-t-il consulté [t de livres]
 'How many did he consult of books'

If a quantifier-bearing adverb occurs VP-initially, however, combien alone cannot be extracted from the Spec of the NP, although the pied-piping the whole NP is possible:

(96) (= (30) Rizzi 1990 p.12)

- a. Combien de livres a-t-il beaucoup consultés t
 how-many of books did he a lot consult
 b. *Combien a-t-il beaucoup consulté [t de livres]
 'How many did he a lot consult of books'

The inner islands observe that adverbials cannot be extracted from the scope of a negative operator:

(97) (= (5) Rizzi 1990 p.3)

- a. Bill is here, which they (don't) know t
 b. Bill is here, as [Op_i they (*don't) know t_i]

All of those cases are generalized by Rizzi that arguments can be extracted but adjuncts cannot.

Under our analysis we can derive these symmetries. So far we have assumed that the feature Op₀ of a Comp (or optionally a pair <D, Op₀> in English) should attract a Wh-word. Now let us assume that the Op₀ of a Comp can also

parametrically be underspecified as Op, and that the OP attracts an operator. This does not mean that a Comp does not have a Q feature. It is just underspecified. This underspecification is also completely compatible with our previous analyses of Wh-asymmetries. It can still attract a Wh-word, since a Wh-word with two features Op and Q, and an FF(Wh-word) is attracted if an Op feature is attracted.

In section 4.3.3.2 we have already considered the asymmetry in (94).

Now let us consider (96). Suppose that a pair $\langle D, Op_Q \rangle$ of the Comp attract a Wh-word. Then it can possibly attract the whole NP with $\langle D, Op_Q \rangle$, as shown in (96.a). This observes the MLC, since there is no intervening category with $\langle D, Op_Q \rangle$ between the NP and the Comp. If the Op of the Comp attracts combine in (96.b), it cannot be successful, violating the MLC, since there is an intervening category Neg with an Op feature between the NP and the Comp. The $\langle D, Op_Q \rangle$ of the comp cannot attract only combien, since it has no D feature.

We can have the same account for the contrast in (97). If the Comp $\langle D, Op_Q \rangle$ attracts a wh-word, it observes the MLC. The intervening Neg operator does not have any D feature. If the Op feature of the Comp attracts a Wh-word, it will violate the MLC because of the intervening Neg operator.

5. Conclusion and Further Research

In the minimalist program economy considerations have played a very important role in optimizing a language system: reducing the components of language only to virtually conceptually necessary modules, and deriving various principles from a very general property of economy.

This thesis has attempted to localize derivational economy uniformly and strictly, and to elucidate its significance and consequences, investigating the cyclicity in overt derivations, Procrastinate effects, Wh-asymmetries, and adjunct symmetries in the minimalist program. By localizing derivational economy, we achieve the following desirable results:

- Derivational economy becomes strictly derivational.
- Computational complexity is significantly reduced by generating only a set of optimal derivations.
- The optimality of a derivation is consistent in a course of derivation and at the interface levels.
- Derivational economy becomes homogeneous in terms of locality.

We also make a proposal that Procrastinate should be

eliminated and replaced with Earliness. With Earliness we have the following advantages:

- All the derivational economy conditions become localized uniformly.
- The Last Resort Condition becomes strengthened so that it can block "no operation".
- The cyclicity of overt computation and Procrastinate effects are derived from one principle, Earliness.

We also hypothesize that multiple features of a target can attract F, and that multiple feature attraction can presumably be parametrized. Under the local Minimal Link Condition multiple feature attraction offers a unified analysis of Wh-asymmetries such as argument-adjunct, argument-extraction, argument-quasi-argument, superiority effects, and adjunct symmetries such as argument-adjunct, pseudo-opacity, and inner island conditions.

We also have some more areas to which our analysis can potentially apply and extend.

One area is LF cyclicity. Recently it has been reported by Bures (1993), Jonas and Bobaljik (1993), Tsai (1994), and others that LF computation is cyclic. If LF cyclicity analysis is correct, it can also be derived from Earliness straightforwardly.

Another area is some argument-adjunct asymmetries in

parasitic gap constructions (Cinque 1990). Parasitic gaps are permissible only if they are referential NPs. If we reduce referentiality/nonreferentiality to some properties of a D feature, as presented in this thesis for the analysis of Wh-asymmetries and adjunct symmetries, we may derive the argument-adjunct asymmetries in parasitic gap constructions from multiple feature attraction and the Minimal Link Condition.

Another area is some Wh-asymmetries in scope ambiguity and extraction (Cinque 1990). If a Wh-NP is extracted out of a Wh-island, it takes only wide scope over quantifiers within the Wh-island, while it exhibits scope ambiguity if it is extracted from a that-clause. If $\langle D, Op_0 \rangle$ attraction is assumed to generate a different LF representation from Op attraction, we may be able to explain relationship between some scope asymmetries and extractability under multiple feature attraction.

We will leave all these areas for further research.

REFERENCES

- Aoun, J. and D. Sportiche (1981) "On the formal theory of government." The Linguistic Review 2, 211-236.
- Bobaljik (1995) "In terms of merger: Single output syntax and the strict cycle." Papers on minimalist syntax: MIT working papers in Linguistics 27, 41-64. Cambridge, Mass.: MIT Press.
- Brody, M. (1995) Lexico-Logical Form: A Radically Minimalist Theory. Cambridge, Mass.: MIT Press.
- Bures, A. (1993) "There is an Argument for an LF Cycle Here." CLS 28, 14-35.
- Chierchia, G. (1991) "Functional WH and weak crossover," in D. Bates (ed.) Proceedings of WCCFL 10, 75-90.
- Chomsky, N. (1970) "Remarks on Nominalization," R. Jacobs and P. Rosenbaum, eds., Readings in English Transformational Grammar, Waltham, Mass.: Ginn.
- Chomsky, N. (1973) "Conditions on Transformations," reprinted in Chomsky 1977, Essays on Form and Interpretation, North-Holland, New York.
- Chomsky, N. (1977) Essays on Form and Interpretation, North Holland, Amsterdam.
- Chomsky, N. (1986) Knowledge of language. New York; Praeger.
- Chomsky, N. (1991) "Some Notes on Economy of Derivation and Representation." R. Freidin, eds., Principles and Parameters in Comparative Grammar, Cambridge, Mass.: MIT Press.
- Chomsky, N. (1993) "A Minimalist Program for Linguistic Theory." K. Hale and S. J. Keyser, eds., The View from Building 20: Essays in Linguistics in Honor of Sylvain Bromberger, 1-52. Cambridge, Mass.: MIT Press.
- Chomsky, N. (1994) "Bare Phrase Structure." Mit Occasional Paper in Linguistics 5. Cambridge, Mass.: MIT Press.
- Chomsky, N. (1995) The Minimalist Program. Cambridge, Mass.: MIT Press.
- Chomsky, N. and H. Lasnik (1993) "The theory of principles and parameters." J. Jacobs, A. von Stechow, W. Sternefeld, and T. Vennemann, eds., Syntax: An international handbook of contemporary research. Berlin: de Gruyter.
- Cinque, G. (1990) Types of A'-dependencies. Cambridge, Mass.: MIT Press.
- Collins, C. (1994) "Economy of Derivation and the Generalized Binding Condition." Linguistic Inquiry 25, 45-61.
- Collins, C. (1995) "Toward a theory of optimal derivation" Papers on minimalist syntax: MIT working

- papers in Linguistics 27, 65-104. Cambridge, Mass.: MIT Press.
- Emonds, J. (1978) "The Verbal Complex of V'-V in French." Linguistic Inquiry 9, 151-175.
- Epstein, S. D. (1991) "Derivational Constraints on A'-Chain Formation." Linguistic Inquiry 23, 235-259.
- Fukui, N. (1993) "Parameters and optionality." Linguistic Inquiry 24, 399-420.
- Higginbotham, J. (1983) "Logical form, binding and nominals" Linguistic Inquiry 14, 395-420.
- Holmberg, A. (1986) Word order and syntactic features in the Scandinavian languages and English. Doctoral dissertation, University of Stockholm, Stockholm.
- Hornstein, N. (1995) Logical Form: From GB To Minimalism Cambridge, Mass.: Blackwell.
- Huang, J. (1982) Logical Relations in Chinese and the Theory of Grammar. Doctoral dissertation, Cambridge, Mass.
- Jackendoff, R. (1977) X' syntax: A study of phrase structure. Cambridge, Mass.: MIT Press.
- Jonas, D. and J. Bobaljik. (1993) "Specs for subjects: The role of TP in Icelandic." Papers on Case & Agreement I. MIT Working Papers in Linguistics 18, 59-98.
- Kayne, R. (1994) The antisymmetry of syntax. Cambridge, Mass.: MIT Press.
- Kitahara, H. (1994) Target α : A unified theory of Movement and structure Building. Doctoral dissertation, Harvard University, Cambridge, Mass.: MIT Press.
- Kitahara, H. (1995) "Target α : deducing strict cyclicity from derivational economy." Linguistic Inquiry 26, 47-78.
- Kitahara, H. (1996) "Minimal Syntactic Procedure: Deriving the Timing of Movement." Paper presented at Michigan State University, Dept of Linguistics, East Lansing, Michigan.
- Larson, R. (1988) "On the double object construction" Linguistic Inquiry 19, 335-391.
- Larson, R. (1990) "Double object revisited: Reply to Jackendoff." Linguistic Inquiry 21, 589-632.
- Lasnik, H. (1992) "Case and expletives." Linguistic Inquiry 23, 381-405.
- Lasnik, H. (1993) "Lectures on Minimalist Syntax," UCWPL Occasional Papers 1. Storrs.
- Lasnik, H. (1995) "Case and expletives revisited: On greed and other human failings." Linguistic Inquiry 26, 615-635.
- 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, Mass.: MIT Press.
- Lee, Daehee (1995) "The Revised Greed Principle and Phrase Structure Constructions: the Earliness Principle,"

- presented in Michagan State University Linguistic Colloquium, October 1995.
- Lee, Daehee (1996) "The Timing Principle On Syntactic Derivations," presented in Michigan Linguistic Society Annual Conference, October 1996.
- Longobardi, G. (1994) "Reference and proper names: A theory of N-movement in syntax and Logical Form." Linguistic Inquiry 25, 609-665.
- Obenauer, H. (1984) "On the Identification of Empty Categories." Linguistic Review 4, 153-202.
- Oka, T. (1993) "Shallowess" Papers on Case & agreement II: MIT working papers in linguistics 19, 255-320. Cambridge, Mass.: MIT Press.
- Oka, T. (1995) "Fewest steps and island sensitivity." Papers on minimalist syntax: MIT working papers in Linguistics 27, 189-208. Cambridge, Mass.: MIT Press.
- Pesetsky, D. (1989) "Language Particular Processes and the Earliness Principle." Ms., MIT, Cambridge.
- Reinhart, T. (1993) "Wh-in-situ in the framework of the minimalist program." Lecture given at the Utrecht linguistics Colloquium.
- Rizzi, L. (1990) Relativized Minimality. Cambridge, Mass.: MIT Press.
- Ross, J. R. (1983) Inner Islands. Manuscript, MIT.
- Tsai, W. D. (1994) On Economizing the theory of A-bar dependencies. Doctoral dissertation, Cambridge, Mass.: MIT Press.
- Ura, H. (1995) "Towards a theory of 'Strictly derivational' economy condition" Papers on minimalist syntax: MIT working papers in Linguistics 27, 243-268. Cambridge, Mass.: MIT Press.
- Sauerland, U. (1995) "Early features" In Papers on minimalist syntax: MIT working papers in Linguistics 27, 223-242. Cambridge, Mass.: MIT Press.
- Watanabe, A. (1995) "Conceptual Basis of Cyclicity" Papers on minimalist syntax: MIT working papers in Linguistics 27, 269-291. Cambridge, Mass.: MIT Press.