Ordering of Operations and Directions of Derivations

SAWADA Tsuyoshi
University of Tokyo

Abstract
This paper is concerned with a formalization of the notion of phrase structure and derivation and the problem of directionality of the operation. It aims at providing a basis for spelling out the ideas I have in mind, according to which a syntactic derivation proceeds from top-down. The formalization presented in this paper also provides a basis for syntactic theories in general which adopt the derivational approach. Most of the ideas can be seen in Sawada (2000).

Keywords: operation, derivation, partial order, top-down, dominance, asymmetrical c-command

1. Introduction

Since the derivational account of structure building is presented by Chomsky (1993), two issues of *structure building* have arisen in syntax:

(1)
a. What is the character of structure building operation?
b. When does each occurrence of structure building operation take place?

Two issues of *movement* which correspond to those of structure building have become major topics in syntactic research:

(2)
a. What is the character of movement operation?
b. When does each occurrence of movement operation take place?

This paper is an attempt to give an answer to the (a) issues in (1) and (2) assuming that what is to be called *the Certainty Principle* is at work.

The *derivation* of a sentence, were it real, should be a representation of some *calculation* process required for the computation of a sentence. Though some of the individual's neurons are working behind the computation of a sentence, whose interaction is no doubt heuristic, there must be some abstract (linguistic) level whose process is algorithmic rather than heuristic. This is what the derivation is to represent if the notion of derivation has substance. Assuming that an *elementary process* in a calculation corresponds to an

Linguistic Research 17(2000) 125-151

operation in a derivation, we reach the view that every operation in a derivation is deterministic. I call this the Certainty Principle.

The Certainty Principle, when combined with some empirical data, entails that a derivation proceeds from top-down. Given the Certainty Principle, the (near-)bottom-up operation *Merge*, which has became recently become dominant, turns out to be wrong and it must be replaced with a top-down operation. Thus, (2a) is answered. Issue (2b) can also be answered under the Certainty Principle but it is not dealt with in this paper.

The organization of this paper is as follows. Section 2 reviews the significance of derivational account of structure building and proposes the Certainty Principle, which constitutes the leading idea in this paper. In section 3, I make clear the distinction between bottom-up and top-down operations by formalizing some concepts. In Section 4, some empirical matters are provided, which entail, when combined with the Certainty Principle, that a derivation proceeds from top-down. Then, section 5 spells out the specifications of the top-down operation, which is an answer to the (a) issues of (1) and (2). Section 6 is a summary.

2. Derivational Syntax and the Certainty Principle

In an earlier stage of generative grammar, the transformational operations assumed thereabout were reduced into sequences of *Move* α, which can be considered *elementary* in two respects:

(3)
a. Since each instance of Move α obeys some local constraint such as the Proper Binding Constraint, the format of Move α is kept reasonably simple.
b. Accordingly, a complex and/ or idiosyncratic transformation effect is not due to a single instance of Move α but to a series of Move α. Thus, the timing of instances of Move α has a non-trivial consequence.

As a theory makes use of an elementary operation, it becomes possible to observe the *elementary process* within syntax and questions about them come to the fore. Let us call such a theory a *derivational* theory. At the counter end is a *representational* theory, which can be compared to a *black-box*, in which a sentence is composed in a single process and is privileged from being questioned as to what is going on within. The reduction of the transformations into Move α strengthened the derivational flavor of the theory of syntax in the area of movement, moving it towards explanatory adequacy.

However, there was an area left behind by this innovation by Move α. Namely, *structure building*. Theories in Principles and Parameters framework have assumed with few exceptions that phrase structures are built pre-syntactically, providing the *D*-Structure.

This conception has the following properties:

(4)

a. Since the sole restrictions on structure building are filters at *D*-Structure, there is no substance to the consideration of a building process as a sequence of simple rewriting rules.

b. Accordingly, the timing of sub-processes within a building process has also no substance. Interaction between any sub-process and Move α is even impossible, since structure building is separated from Move α by the representational level *D*-Structure.

These properties are surely in the counter direction to (3).

In the execution of the Minimalist Program, Chomsky (1993) eliminated *D*-structure from syntactic theory and introduced the idea that phrase structures are built derivationally. He installed a syntactic operation as the origin of a phrase structure, which came to be called *Merge* since Bare Phrase Structure theory (Chomsky (1995a)). One empirical support to this system is that some problems occur with *tough*-construction if the structure building process is not allowed to overlap with syntactic operations (See Chomsky (1995b:188)). Under the new system with Merge, structure building is derivational in two respects:

(5)

a. The format of Merge is kept extremely simple.

b. Accordingly, a complex and/ or idiosyncratic structure is not due to a single instance of Merge but to a series of Merge. A series of Merger can be interrupted by movement operations and the timing of Merger has a non-trivial consequence.

Thence the significance of Bare Phrase Structure theory is that it leads to a derivational account of structure building, which remained representational after the innovation of movement. Since this theory, movement and structure building are both regarded as composed of elementary operations. The derivational theory thus achieved is an important step towards seeking the precise process of syntax.

In this paper, I would like to formalize what follows from the convictions in (6):

(6)

a. The *derivation* of a sentence is a representation of the *calculation process* in brain required for the computation of the sentence.

b. Calculation is deterministic at its elementary level.

Formally taken, (6a) means that a derivation is isomorphic to the calculation that is actually carried out during the computation, where, an *operation* within the derivation corresponds to an *elementary process* within the calculation. Without the assumption of (6a), a *derivation* and its *operations* are abstract artifacts. Only under (6a), they will achieve reality, satisfying the virtual conceptual necessity of Chomsky (1995b). If we admit a process of calculation along this reasoning, (6b) must also be true. A calculation must be manipulated by some algorithm rather than by heuristics. (6a, b) entail that each operation in a derivation is deterministic; it is sensitive to what is going on, and is conducted only when it has something to do with the whole derivation. This concept, which appeared in Sawada (2000), is to be formalized later.

(7) the Certainty Principle, preliminary version
An operation is conducted only when it is verified on its occasion.

The principle crucially narrows down the range of answers to (2). To see how it works, we have to formalize the character of syntactic operations and derivations.

3. Formalization

3.1. The Property of Elementary Operations

Following the suggestions from various motivations (Montague (1974), Larson (1988), Kayne (1984), Chomsky (1995a)), I assume that every node of a phrase structure is binary branching. I also learn from the success of X-bar theory and its extension to functional categories (Chomsky (1986)) that a projected node is categorically related to one of its daughters: the fact that is often mistakenly called endocentricity. I strengthen this notion, following Chomsky (1995a), to the second assumption that a node and one of its daughters have identical content and there is no feature inherent in nodes that represent their projection level. The third assumption, which is not so crucial as the other two, is that word order is not a property of syntax (Chomsky (1995a)).

(8) Assumptions

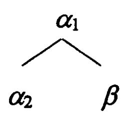
a. Binary branching

b. Identical content of a node and one of its daughters

c. No significance of word order

The assumptions entail that a phrase structure is a repetition of an elementary structure schematized in (9):

(9) Elementary structure



where α_1 and α_2 have the same content and the representational order between α_2 and β has no significance. Since sisterhood can be composed of *direct dominance* (i.e. α is a sister of β ($\neq \alpha$) iff there is γ such that γ directly dominates α and β) but vice versa, the sole elementary relation that can be found in the elementary structure is direct dominance. The elementary structure is thus composed of three nodes α_1, α_2 and β and two direct dominance relations (α_1, α_2) and (α_1, β). An elementary operation creates any of the three nodes that is previously missing and the two direct dominance relations.

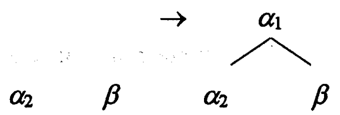
Though three nodes α_1, α_2, and β are involved in the operation in forming (9), α_1 and α_2 share identical information, so β and only either of α_1 or α_2 are necessary as *informational inputs* to execute the operation. Given β and one of the αs, the information needed to give the other α as the *informational output* follows from the operation. Thus, there are two possible combinations of informational input, namely {α_2, β} and {α_1, β}, and according to which one is given, there are two types of operations. I call (10a) and (10b) *bottom-up* and *top-down operations* respectively:

(10) Two types of elementary operations and informational status of the nodes

a. Bottom-up operation.

Informational input: α_2, β

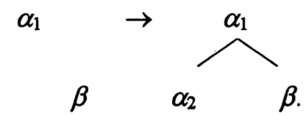
Informational output: α_1



b. Top-down operation.

Informational input: α_1, β

Informational output: α_2



However, from the identity of the content of α_1 and α_2 following from (8b), the distinction between bottom-up and top-down operations is void if we were not to assume one important thing: *cyclicity*. If we assume cyclicity, taking it to be as in (11), a sense that somewhat comes close to Watanabe's (1995: 275) Avoid Redefinition, the distinction makes sense.

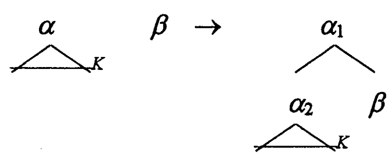
(11) Cyclicity

An operation cannot remove a(n) (elementary) relation established in a previous stage.

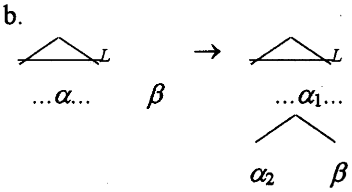
(12a) is possible only by bottom-up operation, α in its left side necessarily being α_2. If α were α_1, the direct dominance of K by α_1 would be removed by the operation. (12b) is possible only by top-down operation, α necessarily being α_1. No operation realizes (12c).

(12)

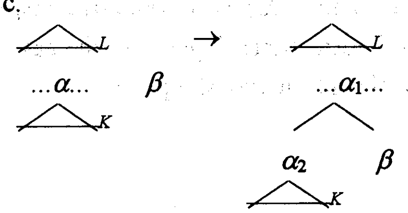
a.



b.



c.



Contrarily, suppose cyclicity is not at work. Then, all of the three processes in (12a, b, c) may be realized as either types of operations. Taking α of (12a, b, c) as α_2, all three processes are possible as an instance of bottom-up operation. Taking α as α_1, all three processes are possible as an instance of top-down operation. Actually, Chomsky's (1995b) Merge is usually alleged to be bottom-up but it is not so in a strict sense, since cyclicity is violable in some cases in this framework. Particularly adjunction, including head incorporation, (or in covert substitution,) can violate cyclicity, making (12b, c) possible despite bottom-up operation. I intend to make a theory that does not nullify the distinction between bottom-up and top-down operations. That becomes possible only under the two assumptions, which I adopt:

(13) Assumptions
a. Cyclicity (11)
b. There are only two types of syntactic theories, one that assumes only bottom-up operation, and the other only top-down operation.

Let us call the theories in (13b) *bottom-up* and *top-down syntax* respectively. The derivations in these theories are to be called *bottom-up* and *top-down derivations* respectively. Though Chomsky's (1995b) Merge is only *near*-bottom-up, the operation becomes (*strictly*) bottom-up in Chomsky (1998, 1999). In section 5, a (strictly) top-down syntax is adopted.

(13) makes clear that an informational input to *o* is not necessarily its *structural input*, a node that is involved in *o* and is introduced into the structure in the derivation *before o*. The *structural status* of a node must be distinguished from its informational status with respect to an operation. The distinction is meaningful particularly in the case of top-down operation. For top-down operation *o*, which does not allow (12a, c), its α_2 does not dominate any node immediately after *o*. If α_2 is to dominate some node, as is the general case, there must be another top-down operation afterwards that takes this α_2 as its structural input. Thus in all top-down operation schematized in (10b), their α_1 is a structural input, existing before the operation. Thence, neither α_2 nor β can be a structural input, since if they also existed in the structure before the operation, it would be established that α_1 did not directly dominate them. From (11), this would mean that α_2 and β could never be

dominated by α_1, which is not the case. Therefore, β in (10b) is a *structural output*, introduced into the structure *by* the operation, despite informational input. In the case of bottom-up operation, the structural input is the same as the informational input.

(14) Structural status
a. In bottom-up operation (10a),
Structural input: α_2, β
Structural output: α_1.
b. In top-down operation (10b),
Structural input: α_1
Structural output: α_2, β.

Though informational status and structural status are different notions, they have the following entailment:

(15)
a. α is a structural input to $o \rightarrow \alpha$ is an informational input to o. Thus,
b. α is a structural output of $o \leftarrow \alpha$ is an informational output of o.

A structural input to o is introduced into the structure before o, and is thus used by o as part of the information to execute o. An informational output of o contains information given by o (and its informational input), and hence does not exist in the structure before o. A node involved in an operation thus has one of the three statuses:

(16) Status of the nodes

	In (10a)	In (10b)
a. Informational input-structural input:	α_2, β	α_1
b. Informational input-structural output:	none	β
c. Informational output-structural output:	α_1	α_2.

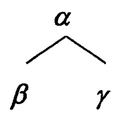
To give a formal characterization to an operation within a derivation, the crucial notion is its structural property. Thus, I define the following notation:

(17) Notation
a. o^{α} = the operation that gives α as a structural output
b. ${}^{\alpha}o$ = the operation that takes α as a structural input

${}^{\alpha}o$ is not necessarily defined for an arbitrary α, though ${}^{\alpha}o$ and o^{α} are unique when defined. Particularly, ${}^{\alpha}o$ does not exist when α is a root in a bottom-up derivation or a terminal in a top-down derivation. Using the notation in (17), the two types of elementary operations are formally characterized as follows.

(18) Definition

Whenever there is an elementary structure in which α directly dominates β and γ,



a. $o^{\alpha} = {}^{\beta}o = {}^{\gamma}o$ iff they are bottom-up operation

b. ${}^{\alpha}o = o^{\beta} = o^{\gamma}$ iff they are top-down operation

Note that for (18) to be valid in general, o^{α} must be defined for arbitrary α, even when α is a terminal in a bottom-up derivation or a root in a top-down derivation. The informational and structural status of the nodes involved in the operation in these cases may not have made clear so far. In these cases, there is no node that exists before the operation and is used as informational input. The information necessary for the operation comes from outside the derivation, namely the lexicon.

(19)

For o^{α}

in a bottom-up derivation, where α is a terminal, or

in a top-down derivation, where α is a root,

a. Informational input-structural input: none

b. Informational input-structural output: none

c. Informational output-structural output: α

3.2. The Property of Derivations: Relations between Elementary Operations

Let us go on to give formal characterization of derivations. Formally, the property of a derivation can be represented as a set of relations between the operations consisting the derivation, specifically by *order* relations between the operations:

(20) Notation

For two operations o_i, o_j in a derivation,

$o_i > o_j$ iff o_i, o_j are ordered and o_i precedes o_j.

The intuition is that an operation that takes α as structural input is always preceded by an operation that gives α as structural output and that this sort of connection is the sole source of ordering between operations. Let us represent this intuition by the following recursive definition of the relation >, which is to be applicable to both bottom-up and top-down derivations.

(21) Definition
a. Base step: $o^{\alpha} > {}^{\alpha}o$
b. Recursive step: If $o_i > o_j$ and $o_j > o_k$, then $o_i > o_k$.
c. Restriction: Only those derived from above consist the relation >.

Note that > is not total, i.e. it is not the case that two operations o_i, o_j in a derivation are necessarily ordered. Two operations for which the condition in (21) is not satisfied are unordered. The relation between arbitrary operations o_i, o_j in a derivation is either:

(22)
a. $o_i = o_j$,
b. $o_i > o_j$,
c. $o_i < o_j$, or
d. unordered

Using the notation $\geq$, meaning > or =, I give the formal definition of derivation:

(23) Definition
A derivation is a pair $(O, \geq)$, where $O = \{o_1, o_2, \ldots, o_n\}$ is a set of operations.

From the definitions (18), (21) and (23), the following theorem is obvious:

(24) Theorem
A derivation is a poset (partially ordered set). That is,
a. Reflexive: $\forall(o_i, o_j). \neg (o_i \not\geq o_j \wedge o_i = o_j)$
b. Antisymmetric: $\forall(o_i, o_j). \neg (o_i \geq o_j \wedge o_i \neq o_j \wedge o_j \geq o_i)$
c. Transitive: $\forall(o_i, o_j). \neg (o_i \geq o_k \wedge o_i \not\geq o_j \wedge o_k \geq o_j)$.

As it becomes clear later, it is useful to represent some non-elementary relations between the nodes of a phrase structure by means of relations between operations. The most important relations that had been elaborated in the literature of generative grammar are *dominance* and *c-command.* Let us take dominance to be a relation recursively defined from the elementary relation direct dominance.

(25) Definition

a. Base step: If α directly dominates β, then α dominates β.

b. Recursive step: If α dominates β and β dominates γ, then α dominates γ.

c. Restriction: Only those derived from above consist the dominance relation.

For the other notion *c*-command, let us basically adopt the definition of Chomsky (1986: 8), who assumes, as I do here, that dominance is irreflexive (cf. Chomsky (1986: 92n11)).

(26) Definition

For two nodes α, β ($\alpha \neq \beta$), α *c*-commands β iff α does not dominate β and every γ that dominates α dominates β.

Notice that the condition $\alpha \neq \beta$ is added to his definition in order to make it work as intended.

A surprising fact is that the two relations between nodes are subsumed into a single relation between operations. Before going into this surprising theorem, let us formalize domination alone:

(27) Theorem

α dominates β

$\leftrightarrow$

$$\begin{cases} {}^{\beta}o \geq o^{\alpha} & \text{(In bottom-up syntax)} \\ {}^{\alpha}o \geq o^{\beta} & \text{(In top-down syntax).} \end{cases}$$

Proof

α dominates β

$\leftrightarrow \exists\, (\gamma_1, \gamma_2, \ldots, \gamma_n)\ (n \geq 0).$

α directly dominates γ_1,

γ_1 directly dominates γ_2,

$\vdots$

γ_n directly dominates β

[∵Definition (25)]

$\leftrightarrow$

$$\begin{cases} \exists(\gamma_1, \gamma_2, \ldots, \gamma_n)\ (n \geq 0).\ {}^{\gamma_1}o = o^{\alpha},\ {}^{\gamma_2}o = o^{\gamma_1},\ \ldots,\ {}^{\beta}o = o^{\gamma_n} & \text{(In bottom-up syntax)} \\ \exists(\gamma_1, \gamma_2, \ldots, \gamma_n)\ (n \geq 0).\ {}^{\alpha}o = o^{\gamma_1},\ {}^{\gamma_1}o = o^{\gamma_2},\ \ldots,\ {}^{\gamma_n}o = o^{\beta} & \text{(In top-down syntax)} \end{cases}$$

[∵Definition (18)]

$\leftrightarrow$

$$\begin{cases} \exists(\gamma_1, \gamma_2, \ldots, \gamma_n)\,(n \geq 0).{}^{\beta}o = o^{m} > {}^{m}o = o^{m\text{-}1} > \ldots > {}^{\gamma 1}o = o^{\alpha} & \text{(In bottom-up syntax)} \\ \exists(\gamma_1, \gamma_2, \ldots, \gamma_n)\,(n \geq 0).{}^{\alpha}o = o^{\gamma 1} > {}^{\gamma 1}o = o^{\gamma 2} > \ldots > {}^{m}o = o^{\beta} & \text{(In top-down syntax)} \end{cases}$$

[∵Definition (21)]

$\leftrightarrow$

$$\begin{cases} {}^{\beta}o \geq o^{\alpha} & \text{(In bottom-up syntax)} \\ {}^{\alpha}o \geq o^{\beta} & \text{(In top-down syntax)} \end{cases}$$

[∵Theorem (24)]

Q.E.D.

This theorem may not appear to be so elegant, since each inequality makes use of in a mixture the two types of notation in (17a) and (17b).

However, the surprising theorem that we are now ready to demonstrate is different:

(28) Theorem

For two nodes α, β ($\alpha \neq \beta$),

α dominates β or α *c*-commands β

$\leftrightarrow$

$$\begin{cases} {}^{\beta}o \geq {}^{\alpha}o & \text{(In bottom-up syntax)} \\ o^{\alpha} \geq o^{\beta} & \text{(In top-down syntax).} \end{cases}$$

Proof

(α dominates β) $\vee$

(α *c*-commands β)

$\leftrightarrow$

(α dominates β) $\vee$

($\neg$ (α dominates β) $\wedge$ (every γ that dominates α dominates β)) (Under $\alpha \neq \beta$)

[∵Definition (26)]

$\leftrightarrow$

((α dominates β) $\vee$ $\neg$ (α dominates β)) $\wedge$

((α dominates β) $\vee$ (every γ that dominates α dominates β))

[∵Distribution law]

$\leftrightarrow$

(α dominates β) $\vee$ (every γ that dominates α dominates β)

[∵Complement law, Identity law]

$\leftrightarrow$

$$\begin{cases} ({}^{\beta}o \geq o^{\alpha}) \vee (\forall\gamma({}^{\alpha}o \geq o^{\gamma}).\ {}^{\beta}o \geq o^{\gamma}) & \text{(In bottom-up syntax)} \\ ({}^{\alpha}o \geq o^{\beta}) \vee (\forall\gamma({}^{\gamma}o \geq o^{\alpha}).\ {}^{\gamma}o \geq o^{\beta}) & \text{(In top-down syntax)} \end{cases}$$

[∵Theorem (27)]

$\leftrightarrow$

$$\begin{cases} ({}^{\beta}o \geq o^{\alpha}) \vee ({}^{\beta}o \geq {}^{\alpha}o) & \text{(In bottom-up syntax)} \\ ({}^{\alpha}o \geq o^{\beta}) \vee (o^{\alpha} \geq o^{\beta}) & \text{(In top-down syntax)} \end{cases}$$

[∵Theorem (24)]

$\leftrightarrow$

$$\begin{cases} ({}^{\beta}o \geq o^{\alpha} > {}^{\alpha}o) \vee ({}^{\beta}o \geq {}^{\alpha}o) & \text{(In bottom-up syntax)} \\ (o^{\alpha} > {}^{\alpha}o \geq o^{\beta}) \vee (o^{\alpha} \geq o^{\beta}) & \text{(In top-down syntax)} \end{cases}$$

[∵Definition (21)]

$\leftrightarrow$

$$\begin{cases} {}^{\beta}o \geq {}^{\alpha}o & \text{(In bottom-up syntax)} \\ o^{\alpha} \geq o^{\beta} & \text{(In top-down syntax)} \end{cases}$$

[∵Theorem (24), Consistency Principle]

Q.E.D.

For some readers, the relation ≥ might not seem to appear as a simple relation, or for some, *asymmetric c-command* may be more important than *c*-command. For them, I briefly demonstrate a similar theorem. This theorem turns out to be useful later:

(29) Theorem

For two nodes α, β ($\alpha \neq \beta$),

α dominates β, $\vee$ α asymmetrically *c*-commands β

$\leftrightarrow$

$$\begin{cases} {}^{\beta}o > {}^{\alpha}o & \text{(In bottom-up syntax)} \\ o^{\alpha} > o^{\beta} & \text{(In top-down syntax).} \end{cases}$$

Proof

(α dominates β) $\vee$ (α asymmetrically *c*-commands β)

$\leftrightarrow$

(α dominates β) $\vee$ ((α *c*-commands β) $\wedge$ $\neg$ (β *c*-commands α))

$\leftrightarrow$

((α dominates β) $\vee$ (α *c*-commands β)) $\wedge$

((α dominates β) $\vee$ $\neg$ (β *c*-commands α))

[∵Distribution law]

$\leftrightarrow$

((α dominates β) $\vee$ (α *c*-commands β)) $\wedge$

$\neg$ (β *c*-commands α)
[$\because$ (α dominates β) $\rightarrow \neg$ (β *c*-commands α)]

$\leftrightarrow$

(((α dominates β) $\vee$ (α *c*-commands β)) $\wedge \neg$ (β dominates α)) $\wedge$
$\neg$ (β *c*-commands α)
[$\because$ (α dominates β) $\vee$ (α *c*-commands β) $\rightarrow \neg$ (β dominates α)]

$\leftrightarrow$

((α dominates β) $\vee$ (α *c*-commands β)) $\wedge \neg$ ((β dominates α) $\vee$ (β *c*-commands α))
[$\because$ Association law, De Morgan's law]

$\leftrightarrow$

$$\begin{cases} {}^{\beta}o \geq {}^{\alpha}o \wedge {}^{\alpha}o \not> {}^{\beta}o & \text{(Under } \alpha \neq \beta \text{ and in bottom-up syntax)} \\ o^{\alpha} \geq o^{\beta} \wedge o^{\beta} \not> o^{\alpha} & \text{(Under } \alpha \neq \beta \text{ and in top-down syntax)} \end{cases}$$

[$\because$ Theorem (28)]

$\leftrightarrow$

$$\begin{cases} {}^{\beta}o > {}^{\alpha}o & \text{(In bottom-up syntax)} \\ o^{\alpha} > o^{\beta} & \text{(In top-down syntax)} \end{cases}$$

Q.E.D.

Finally, let us represent another complex relation between nodes of a phrase structure, which comes out to be rather simple when expressed as a relation between operations. It involves the following identity relation, which should be obvious from (14) and (17):

(30) Theorem

$\alpha = \beta$

$\leftrightarrow$

$$\begin{cases} o^{\beta} = o^{\alpha} & \text{(In bottom-up syntax)} \\ {}^{\alpha}o = {}^{\beta}o & \text{(In top-down syntax)} \end{cases}$$

The complex relation I want is:

(31) Theorem

α dominates $\beta \vee \alpha = \beta$

$\leftrightarrow$

$$\begin{cases} o^{\beta} \geq o^{\alpha} & \text{(In bottom-up syntax)} \\ {}^{\alpha}o \geq {}^{\beta}o & \text{(In top-down syntax)} \end{cases}$$

Proof

α dominates $\beta \vee \alpha = \beta$

$\leftrightarrow$

$$\begin{cases} {}^{\beta}o \geq o^{\alpha} \vee o^{\alpha} = o^{\beta} & \text{(In bottom-up syntax)} \\ {}^{\alpha}o \geq o^{\beta} \vee {}^{\alpha}o = {}^{\beta}o & \text{(In top-down syntax)} \end{cases}$$

[∵Theorem (27), (30)]

$\leftrightarrow$

$$\begin{cases} o^{\beta} > {}^{\beta}o \wedge (o^{\beta} = o^{\alpha} \vee {}^{\beta}o \geq o^{\alpha}) & \text{(In bottom-up syntax)} \\ o^{\beta} > {}^{\beta}o \wedge ({}^{\alpha}o \geq o^{\beta} \vee {}^{\alpha}o = {}^{\beta}o) & \text{(In top-down syntax)} \end{cases}$$

[∵Definition (21)]

$\leftrightarrow$

$$\begin{cases} o^{\beta} \geq o^{\alpha} & \text{(In bottom-up syntax)} \\ {}^{\alpha}o \geq {}^{\beta}o & \text{(In top-down syntax)} \end{cases}$$

[∵Theorem (24), Consistency Principle]

Q.E.D.

Using some of the formalization given in this section, we can represent another property of a derivation, which follows from the property of a phrase structure as being a tree:

(32) Axiom

For α, β taken from a single phrase structure,

a. $\forall\{\alpha, \beta\}. \exists sup\{\alpha, \beta\}$ [*sup*: least upper bound]

b. $\forall\{\alpha, \beta\}\ (\alpha \neq \beta). \nexists l.b.\{\alpha, \beta\}$ [*l.b.*: lower bound].

with respect to partial order $\succcurlyeq$, taking $\alpha \succcurlyeq \beta$ to mean that α dominates β or $\alpha = \beta$. Converting this axiom into the terms of order between operations, the following theorem follows:

(33) Theorem

a. $\forall\{o_i, o_j\}$ (o_i, o_j: from a derivation).

$$\begin{cases} \exists inf\{o_i, o_j\} & \text{(In bottom-up syntax)} \quad [\textit{inf}\text{: greatest lower bound}] \\ \exists sup\{o_i, o_j\} & \text{(In top-down syntax)} \end{cases}$$

b. $\forall\{o_i, o_j\}$ (o_i, o_j: from a derivation, $o_i \neq o_j$).

$$\begin{cases} \nexists u.b.\{o_i, o_j\} & \text{(In bottom-up syntax)} \quad [\textit{u.b.}\text{: upper bound}] \\ \nexists l.b.\{o_i, o_j\} & \text{(In top-down syntax)} \end{cases}$$

Proof

a. $\forall\{\alpha, \beta\}.\exists\gamma.\forall\delta.(\gamma \succcurlyeq \alpha \wedge \gamma \succcurlyeq \beta \wedge (\delta \succcurlyeq \alpha \wedge \delta \succcurlyeq \beta \rightarrow \delta \succcurlyeq \gamma))$

b. $\forall\{\alpha, \beta\}\ (\alpha \neq \beta).\nexists\gamma.(\alpha \succcurlyeq \gamma \wedge \beta \succcurlyeq \gamma)$

[∵Axiom (32)]

$\Leftrightarrow$

a. $\forall\{\alpha, \beta\}.\exists\gamma.\forall\delta.$

$$\begin{cases}(o^\alpha \geq o^\gamma \wedge o^\beta \geq o^\gamma \wedge (o^\alpha \geq o^\delta \wedge o^\beta \geq o^\delta \rightarrow o^\gamma \geq o^\delta)) & \text{(In bottom-up syntax)} \\ ({}^\gamma o \geq {}^\alpha o \wedge {}^\gamma o \geq {}^\beta o \wedge ({}^\delta o \geq {}^\alpha o \wedge {}^\delta o \geq {}^\beta o \rightarrow {}^\delta o \geq {}^\gamma o)) & \text{(In top-down syntax)}\end{cases}$$

b. $\forall\{\alpha, \beta\}\ (\alpha \neq \beta).\nexists\gamma.$

$$\begin{cases}(o^\gamma \geq o^\alpha \wedge o^\gamma \geq o^\beta) & \text{(In bottom-up syntax)} \\ ({}^\alpha o \geq {}^\gamma o \wedge {}^\beta o \geq {}^\gamma o) & \text{(In top-down syntax)}\end{cases}$$

[$\because$ Theorem (31)]

$\Leftrightarrow$

a.

$$\begin{cases}\forall\{o^\alpha, o^\beta\}.\exists o^\gamma.\forall o^\delta.(o^\alpha \geq o^\gamma \wedge o^\beta \geq o^\gamma \wedge (o^\alpha \geq o^\delta \wedge o^\beta \geq o^\delta \rightarrow o^\gamma \geq o^\delta)) & \text{(In bottom-up syntax)} \\ \forall\{{}^\alpha o, {}^\beta o\}.\exists{}^\gamma o.\forall{}^\delta o.({}^\gamma o \geq {}^\alpha o \wedge {}^\gamma o \geq {}^\beta o \wedge ({}^\delta o \geq {}^\alpha o \wedge {}^\delta o \geq {}^\beta o \rightarrow {}^\delta o \geq {}^\gamma o)) & \text{(In top-down syntax)}\end{cases}$$

b.

$$\begin{cases}\forall\{o^\alpha, o^\beta\}\ (o^\alpha \neq o^\beta).\nexists o^\gamma.(o^\gamma \geq o^\alpha \wedge o^\gamma \geq o^\beta) & \text{(In bottom-up syntax)} \\ \forall\{{}^\alpha o, {}^\beta o\}\ ({}^\alpha o \neq {}^\beta o).\nexists{}^\gamma o.({}^\alpha o \geq {}^\gamma o \wedge {}^\beta o \geq {}^\gamma o) & \text{(In top-down syntax)}\end{cases}$$

[$\because$ Theorem (30)]

$\Leftrightarrow$

a. $\forall\{o_i, o_j\}.\exists o_k.\forall o_l.$

$$\begin{cases}(o_i \geq o_k \wedge o_j \geq o_k \wedge (o_i \geq o_l \wedge o_j \geq o_l \rightarrow o_k \geq o_l)) & \text{(In bottom-up syntax)} \\ (o_k \geq o_i \wedge o_k \geq o_j \wedge (o_l \geq o_i \wedge o_l \geq o_j \rightarrow o_l \geq o_k)) & \text{(In top-down syntax)}\end{cases}$$

b. $\forall\{o_i, o_j\}\ (o_i \neq o_j).\nexists o_k.$

$$\begin{cases}(o_k \geq o_i \wedge o_k \geq o_j) & \text{(In bottom-up syntax)} \\ (o_i \geq o_k \wedge o_j \geq o_k) & \text{(In top-down syntax)}\end{cases}$$

Q.E.D.

4. the Certainty Principle and Dependency

In this section, I first formalize the Certainty Principle mainly by means of order between operations, using the results of section 3. Then we see that bottom-up syntax is incompatible, under the Certainty Principle, with some alternation paradigms found in languages and that rather top-down syntax is in concordance with the facts.

Under (7) lie two types of considerations. Negative Condition and Positive Condition. In order for o^α to be verified on its occasion, no element that judges some morphological/ lexical forms placed in α as illicit may be introduced into the derivation by an operation previous or simultaneous to o^α. This is the Negative Condition. In formalizing Negative Condition, two types of dependencies must be distinguished: asymmetrical dependency (typically government) and symmetrical dependency (agreement). When two elements symmetrically depend on each other, in agreement, it is a subtle matter whether the introduction of one of them into the structure is prior to the other and determines its form. Negative Condition of the Certainty Principle should be relevant only for asymmetrical relation, in which cases it is obvious that one element is prior and asymmetrically affects the other.

In order for an operation to be verified, it must also be motivated by some element previously introduced into the derivation. This is the Positive Condition. Following these considerations, I give the formalization of the Certainty Principle as in (34):

(34) the Certainty Principle, formalized

a. Negative Condition

If the existence and/ or the content of α may affect β but not vice versa, then $o^{\alpha} > o^{\beta}$.

b. Positive Condition

$\forall o^{\alpha} (\exists \beta . o^{\alpha} = {}^{\beta}o). \exists o^{\gamma}. (o^{\gamma} > o^{\alpha} \wedge \exists \text{P (P: a property of } \gamma). \text{ P requires } \alpha)$

Note that (34a) does not only disallow $o^{\beta} \geq o^{\alpha}$. $o^{\alpha}$and o^{β} also must not be unordered.

Note that o^{α} in (34b) ranges only over operations that have structural input. Operations in a bottom-up derivation that give a terminal and the operation in a top-down derivation that gives a root are exempt from the condition (Cf. (19)).

4.1. Alternation in A-Bar-Bound Positions

In many languages, an element in *A*-bar position can establish coreference, at least substandardly, with a resumptive pronoun placed in *A*-position, besides establishing coreference by the ordinary *A*-bar movement strategy. In English, resumptive pronoun and *A*-bar movement strategies show (near-)complementary distribution. In the argument here, it is not so important whether this complementary distribution is a strict one or not. From the fact that they are both *A*-bar bound by *wh*-phrases, complemented with the (near-)complementary distribution, we can reasonably say that resumptive pronouns and *A*-bar traces are two forms of the same category. That is, resumptive pronoun vs. *A*-bar trace is a lexical paradigm.

In some cases, an *A*-bar-bindee must take the form of *A*-bar trace and cannot be dressed in resumptive pronoun.

(35)

a. who_1 does Mary like e_1

b. * who_1 does Mary like him_1

c. to whom_1 are you writing a letter e_1

d. * to whom_1 are you writing a letter (to) him_1

e. whose theory$_1$ do you like e_1
f. * whose theory$_1$ do you like it$_1$
g. who$_1$ e_1 likes John
h. * who$_1$ she$_1$ likes John
i. whose friend$_1$ e_1 came to the party
j. * whose friend$_1$ he$_1$ came to the party
k. when$_1$ will Mary come e_1
l. * when$_1$ will Mary come then$_1$
m. where$_1$ does live e_1
n. * where$_1$ does she live there$_1$
o. which theory$_1$ was it difficult to understand e_1
p. * which theory$_1$ was it difficult to understand it$_1$
q. who$_1$ did she claim that John likes e_1
r. * who$_1$ did she claim that John likes her$_1$
s. which book$_1$ did you sell after reading e_1
t. * which book$_1$ did you sell after reading it$_1$

In other cases, the form is preferred to be resumptive pronoun and not *A*-bar trace.

(36) Islands
a. * which theory$_1$ was [understanding e_1] difficult
b. which theory$_1$ was [understanding it$_1$] difficult
c. * who$_1$ did she make [the claim that John likes e_1]
d. who$_1$ did she make [the claim that John likes her$_1$]
e. * which book$_1$ did you go to school after reading e_1
f. which book$_1$ did you go to school after reading it$_1$

(37) Intervening elements

a. * who_1 do they know [whether e_1 likes my theory]

b. who_1 do they know [whether he_1 likes my theory]

c. * the boy [who_1 Bill wondered [what Jeff said to e_1]]

d. the boy [who_1 Bill wondered [what Jeff said to him_1]]

e. ?? this problem [op_1 that my chemistry professor wondered [whether or not anyone could do e_1]]

f. this problem [op_1 that my chemistry professor wondered [whether or not anyone could do it_1]]

g. ?? those documents [$which_1$ I can never remember [where I put e_1]]

h. those documents [$which_1$ I can never remember [where I put $them_1$]]

(c, f: Napoli (1993). h: Haegeman (1994))

(38) Islands with intervening elements

a. * the man [who_1 they think [that [when Mary marries e_1] then everyone will be happy]]]

b. the man [who_1 they think [that [when Mary marries him_1] then everyone will be happy]]]

c. ?* who_1 can't you make any sense out of the papers [that e_1 writes]

d. ? who_1 can't you make any sense out of the papers [that he_1 writes]

(a: Haegeman (1994))

(39) coordinate structure

a. * I met the boy [who_1 Bill invited [me and e_1] to the party]

b. * I met the boy [who_1 Bill invited [t_1 and me] to the party]

c. ? I met the boy [who_1 Bill invited [me and him_1] to the party]

(a, b, c: Napoli (1993))

Some factors affecting the form of the *A*-bar bindee, though not necessarily uniquely determining it, is whether there is an island or a coordination that dominates the *A*-bar bindee and whether there is an intervening element that asymmetrically *c*-commands the *A*-bar bindee. On the other hand, it is doubtful that an island, etc., is created or not in order to meet the requirement of a resumptive pronoun or an *A*-bar trace. Thus, there is an asymmetrical dependency from the structure in question to the *A*-bar bindee. Let α be an island, an intervening element or a coordination that may affect the *A*-bar bindee β. According to Negative Condition (34a) of the Certainty Principle, α and β must satisfy

(40)

$o^{\alpha} > o^{\beta}$

From the observation above, there are cases where α dominates β or α asymmetrically *c*-commands β. Let us first consider the former. From theorem (29) and (31), we get:

(41)

α dominates β

$\leftrightarrow$

$$\begin{cases} \text{a. } {}^{\beta}o > {}^{\alpha}o \wedge o^{\beta} \geq o^{\alpha} & \text{(In bottom-up syntax)} \\ \text{b. } o^{\alpha} > o^{\beta} \wedge {}^{\alpha}o \geq {}^{\beta}o & \text{(In top-down syntax)} \end{cases}$$

depending on which syntax is assumed. We can easily see that (41a) contradicts with (40) but that (41b) in concordance with (40).

Let us consider the case where α asymmetrically *c*-commands β. From theorem (29) and (31), we get:

(42)

α asymmetrically *c*-commands β

$\leftrightarrow$

$$\begin{cases} \text{a. } {}^{\beta}o > {}^{\alpha}o \wedge o^{\beta} \not\geq o^{\alpha} & \text{(In bottom-up syntax)} \\ \text{b. } o^{\alpha} > o^{\beta} \wedge {}^{\alpha}o \not\geq {}^{\beta}o & \text{(In top-down syntax).} \end{cases}$$

(42b) has no problem with (40). Note that (42a) contradicts with (40). From (42a), an equivalence of α asymmetrically *c*-commanding β in bottom-up syntax, the relation between o^{α} and o^{β} is most specifically $o^{\beta} \not\geq o^{\alpha}$, which does not entail $o^{\alpha} > o^{\beta}$ since $\geq$ is partial. From definition (21), namely (21c), the relation between o^{α} and o^{β} is unordered (case (22d)). It is not unclear. This contradicts with (40).

4.2. Alternation in A Position

There are three types of overt nominal expressions that can appear in *A*-position: anaphor, pronominal and *r*-expression. A significant fact is that they show near-complementary distribution (Chomsky (1981), Huang (1983)). From the fact that the distribution of three types of expressions is restricted to *A*-position, complemented with the near-complementary distribution, we can reasonably say that anaphor, pronominal and *r*-expression are different forms of the same category (c.f. Reinhart (1983)). That is, the distinction between anaphor, pronominal and *r*-expression is a lexical/ morphological paradigm.

The form of overt expression α in A-position (whether anaphor, pronominal or r-expression) is affected by whether there is an element A-binding α and by whether there is one A-binding α in some local domain. This dependency, widely accepted as the Binding Condition, can be represented as follows, ignoring the break of complementary distribution, the phenomena of long distance anaphor and split antecedent and perhaps some other.

(43) Equivalence to the Binding Conditions

Let

$B_\beta(\alpha) \leftrightarrow \alpha\, A$-binds β and

$G_\beta(\alpha) \leftrightarrow \alpha$ is inside the governing category of β.

a. If $\exists\alpha.\ B_\beta(\alpha) \wedge G_\beta(\alpha)$, then β is an anaphor.

b. Otherwise if $\exists\alpha.\ B_\beta(\alpha)$, then β is a pronominal.

c. Otherwise, β is a pronominal or an r-expression.

(44)

a. John_1 hit {himself_1/ *him_1/ *John_1}

b. John_1 thinks that Mary_2 loves {*himself_1/ him_1/ *John_1}

c. Mary_2 loves {*himself_1/ him_1/ John_1}

Thus, the form of α is affected by the existence of β that A-binds α (in some local domain). On the other hand, it cannot be the case that the form of a binder is determined by the form of its bindee.

(45)

a. {John_1 believes $\text{himself}_{1/*2}$ to be/ $\text{he}_{1/*2}$ is/ $\text{John}_{1/*2}$ is} killing himself_1

b. {John believes himself_1 to be/ he_1 is/ John_1 is} proving that he_1 is a genius

c. * {John believes himself_1 to be/ he_1 is/ John_1 is} proving that John_1 is a genius

In (45), the form of the shadowed element in the higher clause does not seem to be dependent on the shadowed element in the lower clause. In (45a), all three forms are allowed as long as they are binders. In (45b), all forms are allowed. In (45c), all forms are disallowed. If a binder is dependent on a bindee, it must be the case that its existence in the first place is dependent, rather than its form. This means that a binder is created (as in (45a)) or not (as in (45c)) depending on the form of α. This is not plausible. What is to be concluded is that there is an asymmetrical dependency from the existence of a (local) binder to the form of a bindee, not vice versa. Let α be an element in a position that can possibly bind β. According to Negative Condition (34a) of the Certainty Principle, α and β must satisfy (40).

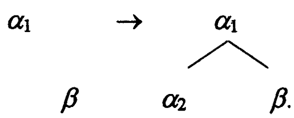
The set of such αs is a subset of elements c-commanding β. We can further restrict this set to the set of elements asymmetrically *c*-commanding β. From theorem (29) and (31), we get (42). As we saw, (42b) has no problem with (40), while (42a) contradicts with (40).

5. Top-Down Operation

In the preceding sections, we saw that bottom-up syntax induced contradiction with some facts of natural language under the assumption of the Certainty Principle unlike top-down syntax. In this section, I adopt top-down syntax.

Let us call the top-down operation *expansion*. An instance of expansion schematized in (46) involves three nodes with different combinations of informational and structural status.

(46)



α_1: Informational input-structural input

β: Informational input-structural output

α_2: Informational output-structural output

A significant fact is that there is no need to introduce some notational feature in order to express this difference in top-down syntax. This is not so in bottom-up syntax (See (16)). In the following sections, let us see the properties of the nodes of each type.

5.1. α_2: Informational Output (-Structural Output)

The informational output α_2 of expansion is the node that is usually referred to as *head* in the broad sense, i.e. in the sense not limited to zero level projections. Thus, a node can be determined as a head or not by being referred to its informational status with respect to expansion.

(47)

Head (in the broad sense) = informational output with respect to an expansion.

This identification of head is useful in identifying different positions within a phrase structure. Following the inclusiveness condition, Chomsky (1995b: 251) proposes a way to dispense with referential indices by inspecting the context in which the phrase appears. He argues that occurrences of an identical element can be identified from each other by being referred to by their sisters. However, this does not seem to work in general. For deletion to apply complementarily to the two occurrences of [*that John was asleep*] in (48) at LF, following preference principle, they must be at least distinguished.

(48) [which claim that John was asleep] did you hear [~~which claim that John was asleep~~]

According to Chomsky's method, each occurrence would be represented by its respective sister, *claim* in both cases. To distinguish the two occurrences of *claim*, it must be done by referring to the two occurences of [*that John was asleep*], which causes a regress.

A different way of appealing to the context to identify an element is to refer to the path connecting the element to the root node. In top-down syntax, any α is connected to the unique root by one and only one path from the moment it is introduced into the structure, and this connection does not change throughout the derivation (See (11)). By the head-non-head distinction of the two sisters by their informational status, every path is uniquely identified without stipulating any notational feature. Therefore, every node in a phrase structure can be identified throughout the derivation.

In bottom-up syntax, however, a node can be identified in this way only after the root is created by the final operation. That is, a node may be identified after the derivation but cannot be identified throughout the derivation. If indices were to be really dispensed with, then not only LF interpretation but also the computation must be able to identify the nodes without index. For computation, identification of a node is needed during the derivation but this is impossible in bottom-up syntax.

5.2. β: Informational Input-Structural Output

Since node β in (46) is an informational input, its content is given from outside the operation. One possibility is lexical insertion, in which case the content of β is given from the lexicon. Another possibility is movement/ copy, in which case β contains an occurrence/ copy of a node previously introduced into the structure. Conversely, whenever a new lexical item or a moved item/ copy is to fill β, the information for β is given from outside the operation and β appears as a new node in the structure; β is the informational input-structural output with respect to that operation.

(49)
Informational input-structural output
↔
a. location of lexical insertion
b. destination of movement/ copy

The informational input-structural output β with respect to some operation is called maximal projection in conventional terms.

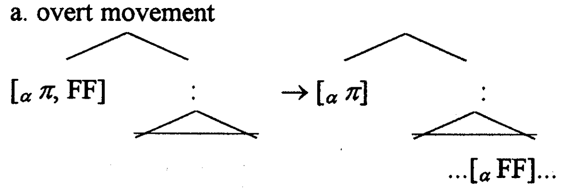
(50)
Maximal projection = informational input-structural output with respect to an expansion

Thus in top-down syntax, the location of lexical insertion and the destination of movement/ copy is a maximal projection.

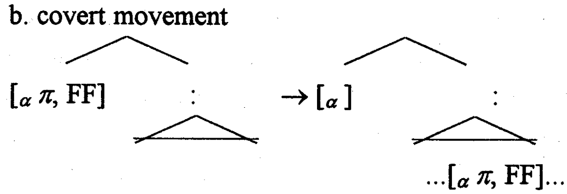
When there is a movement/ copy, the content of trace/ copy is identical, except for the phonological feature. Therefore, the antecedent of movement/ copy operation must also have the status of informational input-structural output with respect to an expansion. In conventional terms, the antecedent of a movement/ copying operation is always a maximal projection. Since head incorporation is tending to be excluded from syntax, being replaced by different mechanism, my consideration predicts what is usually accepted

In movement/ copy, the PF feature π of the informational input-structural output either stays in the starting point or lowers to the destination. When π stays in the starting point, which is the usual case especially in *A*-movement, it becomes overt movement. When π moves along with other features to the destination of the movement, it becomes covert movement.

(51)
a. overt movement



b. covert movement



5.3. Not α_1: Non-Structural Input

A minimal projection is a node that does not dominate any node. This property is expressed in top-down syntax as not being the structural input with respect to any operation:

(52)
Minimal projection = structural input with respect to no expansion

Being a minimal projection has correlation with phonological realization. A node that is phonologically realized is a minimal projection. Conversely, we can take it that a minimal projection is a node that is phonologically realized, admitting null phonological realization for empty heads, such as C of matrix declarative in English. This means that not being a structural input with respect to any expansion is equivalent to being phonologically realized. Then, it is natural to assume the following:

(53) Fixing a node as the location of pronouncing a PF feature π causes the node to be inaccessible as a structural input.

When a node is phonologically realized, it does not project (downwards) any more.

A node dominated by the antecedent or by the trace/ copy of a movement/ copying operation may not be a structural input to further operation because the identity of the antecedent and the trace/ copy would not be guaranteed. This fact is deduced only if phrase structures that have fixed all its π can be an antecedent of a movement/ copying operation. Take this as a condition included within movement/ copying process.

(54)
Movement/ copying process seeks for an antecedent that has fixed all the π within it.

It is widely accepted that two elements involved in a movement/ copy process are in a(n) (asymmetric) c-command relation, known as the Proper Binding Condition, with the exception of head incorporation, including absorption, which now tends to be replaced by different mechanism. My formalization predicts this to be so, without additional stipulations.

(55)
For antecedent α and its trace/ copy β, α asymmetrically c-commands β.

Proof

Since the morphological form of a trace/ copy in the usual overt movement case, i.e. phonologically empty, is due to the existence of its antecedent and not vice versa, the two elements are in an asymmetrical dependency. Therefore from (34a),

(56)
$o^{\alpha} > o^{\beta}$

where α is the antecedent and β is its trace/ copy. From (53) and (54), there must not exist ${}^{\alpha}o$ for antecedent α. Thus,

(57)
$^{\alpha}o \nsucc {}^{\beta}o.$

From (28) and (29), (56) and (57) entail (55).
Q.E.D.

6. Summary

Under plausible assumptions, a syntactic theory that adopts the derivational approach is either of the two types: bottom-up or top-down. Using the formalized notion of phrase structure and derivation, I stated the Certainty Principle to adopt the algorithmic view of derivation. From the empirical side, we observed that in the paradigms in A-bar-bound positions and A-positions, an element that is dependent on another element is either dominated or asymmetrically c-commanded by the latter. Combining the Certainty Principle with the observation, Merge turned out to be wrong and it was replaced with a top-down operation.

References

Chomsky, Noam. (1986) *Barriers*, MIT Press, Mass.

Chomsky, Noam. (1993) "A Minimalist Program for Linguistic Theory," *The View from Building 20*, ed. by Hale and Keyser, MIT Press, Mass.

Chomsky, Noam. (1995a) "Bare Phrase Structure," *Government and Binding Theory and the Minimalist Program*, ed. by Webelhuth, Oxford, Blackwell.

Chomsky, Noam. (1995b) *The Minimalist Program*, MIT Press, Mass.

Chomsky, Noam. (1998) *Minimalist Inquiries: The Framework, MIT Working Papers in Linguistics* **15**.

Chomsky, Noam. (1999) *Derivation by Phase, MIT Occasional Papers in Linguistics* **18**.

Haegeman, Liliane. (1994) *Introduction to Government and Binding Theory*, 2nd ed., Oxford, Blackwell.

Huang, C. T. James (1983) "A Note on the Binding Theory," *Linguistic Inquiry* **14**: 554-561

Kayne, Richard S. (1984) *Connectedness and Binary Branching*, Foris, Dordrecht.

Larson, Richard (1988) "On the Double Object Construction," *Linguistic Inquiry* **19**: 335-391.

Montague, Richard (1974) *Formal Philosophy: Selected Papers of Richard Montague*, ed. by R. H. Thomason, Yale University Press, New Heaven.

Napoli, Donna. J. (1993) *Syntax: Theory and Problems*, New York, Oxford University Press.
Reinhart, Tanya (1983) *Anaphora and Semantic Interpretation*, Croom Heim, London.
Sawada Tsuyoshi (2000) *The the Certainty Principle and Top-Down Operation*, M.A. thesis, Tokyo Metropolitan University.
Watanabe, Akira (1995) "Conceptual Basis of Cyclicity," *MIT Working Papers in Linguistics* **27**.