『영미연구』
제48집 (2020): 191-216

Platform-based Sentence Generation: Unity of Sentence Elements and Hierarchy of Arguments without the Use of Lines and Binary Branching*

Cho, Keeseok
Professor, Cyber Hankuk University of Foreign Studies

[Abstract]

This article seeks to investigate how the unity of syntactic elements is formed and how a hierarchy of arguments is provided in the current framework of generative grammar as well as to provide alternative solutions to these requirements. Syntactic elements need grouping to form a unity. The current theory of generative grammar uses vertical and slanted lines to group syntactic elements and form a unity among them. However, the use of lines causes computational complexity and violates the inclusiveness condition. A new mechanism for providing a unity of syntactic elements will be put forth in this study. The current theory of generative grammar uses binary merge to create hierarchy between syntactic arguments. This hierarchy between syntactic arguments is necessary to satisfy the requirements of the binding theory. However, the binary merge-based hierarchy is problematic in these two

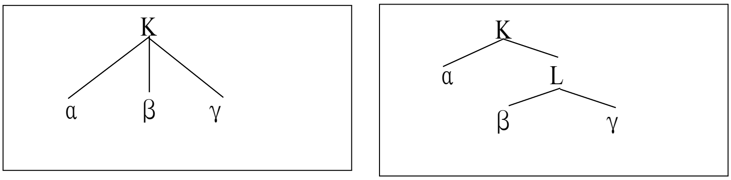
* This work was supported by a 2019 research grant from Cyber Hankuk University of Foreign Studies.

respects. First, it creates not only hierarchy between the arguments, but also that between non-arguments. Second, the binary merge-based hierarchy provides a hierarchical structure for binding theory to operate in complex sentences comprised of main clauses and embedded clauses. However, it does not provide a legitimate hierarchical structure for binding theory to operate in compound sentences comprised of coordinate clauses. For this reason, another platform-based hierarchy between such arguments will be put forth as a solution to these problems.

Key Words: unity of sentence elements, hierarchy of arguments, c-command, platform-based sentence generation, platform-based c-command

1. Introduction

The X-bar theory purported in Chomsky (1970) uses slanted and vertical lines to form a unity between syntactic elements. Suppose that we have three syntactic elements: α, β, and γ. These syntactic elements can be merged into a single unit through the use of lines.


(1) a. workspace A

b. workspace B

In (1a), three syntactic elements α, β, and γ are grouped into one unit K by multinary merge. On the other hand, in (1b), the three syntactic elements α, β, and γ are also merged into the single unit K, but instead by binary merge. In both (1a) and (1b), K is a categorial projection of any combination of α, β, and γ. In (1b), L is a categorial projection of β or γ. The merge operation in (1a) and (1b) employs the use of lines to form the single unit of the three syntactic elements. The three syntactic elements cannot be grouped into one unit without using the lines. The difference between (1a) and (1b) is that the former uses multinary merge while the latter uses binary merge. In the multinary merge system, there is no hierarchy between the three syntactic elements α, β, and γ, which symmetrically C-Command each other.

The binary merge system provides a hierarchy between the three syntactic elements, so that α can asymmetrically C-Command β and γ, and β and γ can symmetrically C-Command each other. The properties of X-bar theory can be summarized as follows.

(2) Properties of structures by X-bar theory

functions		methods
1	unity of syntactic elements	use of lines
2	hierarchy of syntactic elements	binary merge
3	C-Command	sisterhood-based C-Command
4	projection	categorial projection

The structure created by X-bar theory shows the unity of syntactic elements by the use of lines, provides the hierarchy of syntactic elements by binary merge, implements sisterhood-based C-Command; and uses categorial projection. However, the line-based binary merge system has some significant problems. First, it depends on the use of artificial lines to show a unity of the syntactic elements; but this violates the inclusiveness condition. Second, the hierarchy formed in the binary merge system cannot satisfy the binding requirements for the complex sentences comprised of main clauses and embedded clauses and compound sentences comprised of coordinate clauses; and this will be discussed in the following sections. Third, categorial projections cause a computational complexity.[1] However, a new platform-based hierarchy will be put forth as a solution to all these problems. The new platform-based hierarchy can be schematized as follows.

(3) Properties of platform-based structures

functions		methods
1	unity of syntactic elements	syntactic positioning
2	hierarchy of syntactic elements	platform-based hierarchy
3	C-Command	platform-based C-Command
4	projection	no categorial projection

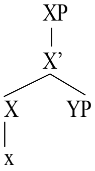
The structure created by the platform-based system shows the unity of syntactic elements by positioning them under the same syntactic position, provides the hierarchy of syntactic elements by mapping them into different platforms, implements platform-based C-Command, and uses no categorial projection. This new mechanism will be shown to be an alternative to Chomsky (1970)'s X-bar theory, in that it neither violates the inclusiveness condition nor causes any computational complexity.

2. Properties of X-bar Theory

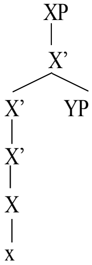
The mechanism of Chomsky (1970)'s X-bar theory uses artificial lines to show a unity of syntactic elements, and it employs categorial projections to specify the projection of syntactic structures. Consider the following structures.

(4) Essential mechanism of X-bar theory and their problems

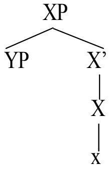
a. bar projection for complement



b. bar projection for adjunct



c. maximal projection for specifier

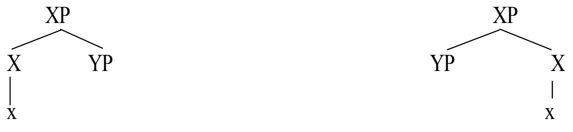


In structure (4a), the lexical item x undergoes head projection to X, which merges with the complement YP and undergoes branching projection to the bar level X’. This then undergoes non-branching projection to the max XP. The head projection introduces category X, and the branching intermediate projection introduces two slanted lines and bar level category X’. The non-branching max projection introduces vertical lines and max level category XP.

In structure (4b), the head X undergoes branching projection to bar level category X’ before it merges with the adjunct YP to project to another bar level category X’. The topmost X’ projects to max level category XP by non-branching projection. The process in (4b) introduces category X, three occurrences of bar level category X’, max level category XP, three vertical lines, and two slanted lines.

In (4c), the head X undergoes branching projection to bar level category X’ before it merges with the specifier YP to project to max level category XP. The process in (4c) introduces category X, bar level category X’, max level category XP, one vertical line and two slanted lines.[2] If we eliminate non-branching projection, (4a) and (4b) will have identical structures, and (4c) will differ only in the order of precedence between X and YP.

(5) a. head-complement and head adjunct structures b. spec-head structures



If the order were irrelevant in grammar, the structures (4a), (4b), and (4c) would all have identical structures.

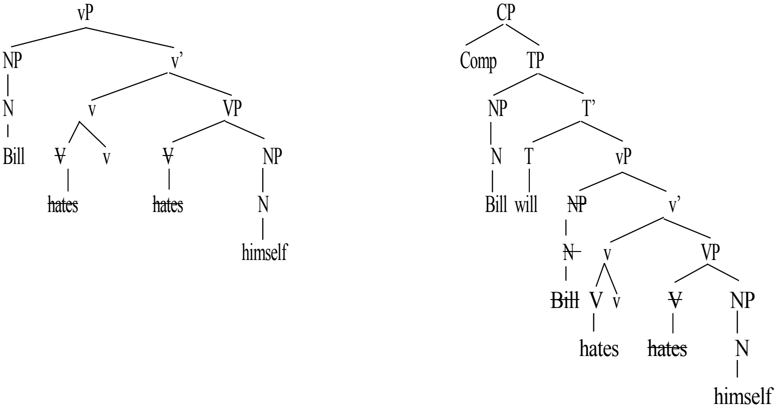
3. The Mechanism of X-bar Theory

In this section, we will demonstrate the application of X-bar theory to some binding theory-related sentences and point out some theoretical and empirical problems. Consider the following sentences.

(6) a. Bill will hate himself. (Bill = himself)
  b. Maria will show the man to himself. (the man = himself)
  c. Sue did not cry when she was bullied. (Sue = she)

Sentences (6a), (6b), and (6c) relate to binding principles A and B. Aoun (1985: 100)'s definition of anaphor requires that *Bill* C-Command *himself* in (6a).[3] The application of X-bar theory satisfies this requirement, as seen below.

(7) a. stage 1 b. stage 2.

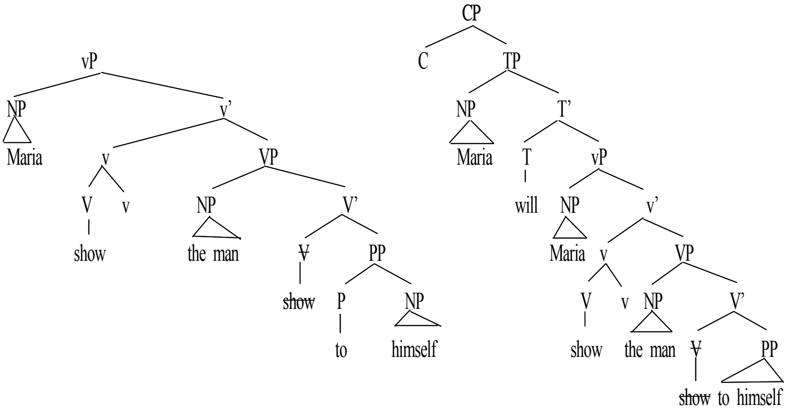


Structure (7a) shows the transitive argument structure of sentence (6a) in terms of the Chomskian double verb phrase structure. Structure (7b) shows the complete sentential structure. Regardless of whether we apply binding principle A in stage 1 or in stage

2, the antecedent *Bill* asymmetrically C-Commands the anaphor *himself*, and this satisfies the binding principle A. The structural hierarchy that implements C-Command, however, is established at the expense of computational complexity and the violation of the inclusiveness condition. The introduction of categorial elements N, NP, V, VP, v', vP, T, T', TP, C, and CP and slanted and vertical lines, which were not at the numeration, in the course of the derivation increases computational complexity and violates the inclusiveness condition.

Sentence (6b) is a dative construction with triple arguments. The binding principle A requires that the antecedent *man* C-Command the anaphor *to himself*. The Chomskian double verb phrase structure enables the antecedent *man* to C-Command the anaphor *to himself*, as shown below.

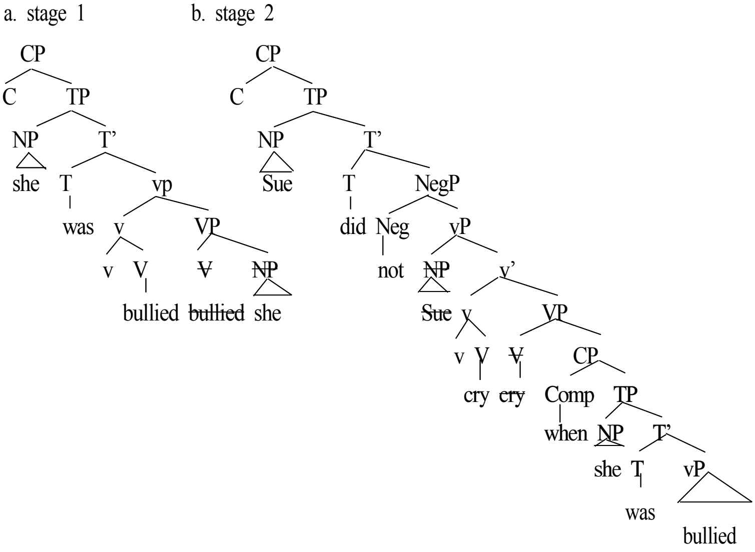
(8) a. stage 1 b. stage 2



Structure (8a) outlines the argument structure of the ditransitive verb *show* in terms of the Chomskian double verb phrase structure. Structure (6b) reaches Structure (8b) by the time Comp merges with TP. Both in stage 1 and in stage 2, the antecedent *man* asymmetrically C-Commands the anaphor *to himself*, and this satisfies the binding principle A. However, the structural hierarchy that implements the asymmetric C-Command between *the man* and *to himself* is established at the expense of introducing numerous categorial elements N, NP, P, PP, V, VP, v’, vP, T, T’, TP, C, and CP and artificial slanted and vertical lines. Adding these elements, which were not at the numeration, increases computational complexity and violates the inclusiveness condition.

Binding principle B states that the main clause subject *Sue* may or may not C-Command the embedded clause subject *she* in (6c). Structure (9a) shows the syntactic structure of the embedded clause of (6c), as shown below.

(9)



a. stage 1

b. stage 2

Structure (9a) shows the syntactic structure of the embedded clause (6c). Structure (9a) reaches structure (9b) by the time the entire syntactic structure of sentence (6c) is completed. In (9b), the main clause subject *Sue* C-Commands the embedded clause subject *she*. This does not violate the binding principle B since *Sue* is outside the governing category of *she*, which is in the embedded clause.

The mechanism of Chomsky (1970)'s X-bar theory provides a legitimate structure for the binding theory to judge the grammaticality of sentence (6c). This is also at the expense of computational complexity and the violation of the inclusiveness condition, as clearly shown in structures (9a) and (9b).

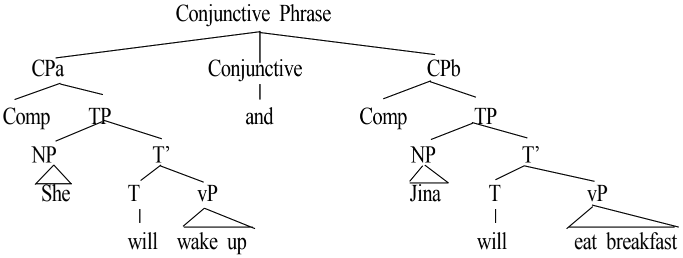
The problems of X-bar theory are not limited to computational complexity and violation of inclusiveness condition. Consider the following sentences.

(10) a. Brian will dawdle, and he will miss the bus. (Brian = he)
b.*She will wake up, and Jina will eat breakfast. (She = Jina)

Sentence (10a) is a compound sentence comprised of two independent clauses. The first conjunct subject *Brian* and the second conjunct subject *he* are coindexed. We can postulate the conjunctive phrase by either binary merge or ternary merge. The latter sticks to the mechanism of X-bar theory. Whichever structure is assumed, the first conjunct subject *Brian* cannot C-Command the second conjunct subject *he*. Hence regardless of whether the first conjunct is included in the governing category of the second conjunct subject *he*, sentence (10a) does not violate the binding principle B.

Sentence (10b) is another compound sentence that has the same structure as (10a). The first conjunct subject *She* cannot C-Command the second conjunct subject *Jina*, as shown below.

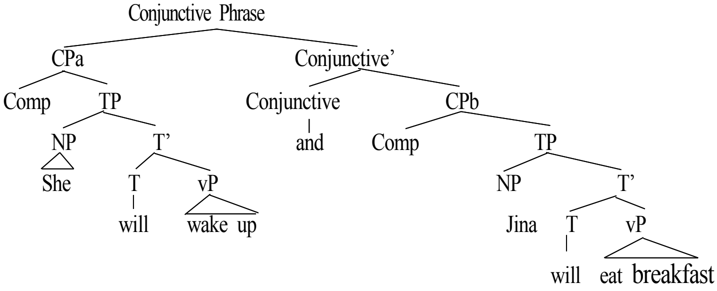
(11) Conjunctive phrase by ternary merge



Structure (11) is a conjunctive phrase headed by the conjunctive *and*, which merges with the first conjunct CPa and the second conjunct CPb at the same time. The first conjunct subject *She*, which C-Commands T' and all the nodes under T' in CPa, cannot C-Command the second conjunct subject *Jina*. Hence regardless of whether the first conjunct is included in the governing category of the second conjunct subject *Jina*, sentence (10a) does not violate the binding principle C.

The C-Command domain of the first conjunct subject *She* does not change even when we postulate the conjunctive phrase by binary merge. Consider the following structure.

(12) Conjunctive phrase by binary merge



In structure (12), where the conjunctive *and* merges with the second conjunct CPb and then merges with the first conjunct CPa, the first conjunct subject *She*, whose C-Command domain is limited to T' and all the nodes under T' in CPa, cannot C-Command the second conjunct subject *Jina*. Therefore, whether we postulate the conjunctive phrase by ternary merge as in (11), or postulate the conjunctive phrase by binary merge as in (12), we cannot account for why sentence (10b) is an ungrammatical sentence under the current framework of the binding theory.[4]

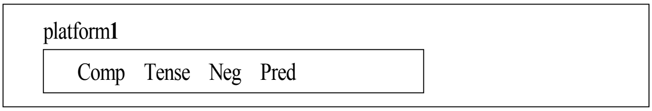
All of the above provide us with empirical reasons to revise the framework of X-bar theory and offer an alternative system of generating sentential structures, which can account for grammaticality of the compound sentences, such as in (10a) and (10b), without causing computational complexity and violating the inclusiveness condition.

4. Platform-based Sentence Generation

It has been demonstrated that the framework of X-bar theory is problematic in three significant respects. First, it depends on categorial projection, resulting in computational complexity. Second, it uses slanted and vertical lines to show the unity of syntactic elements, which causes computational complexity and violates the inclusiveness condition. Third, it cannot account for grammaticality of the compound sentences with regard to the binding theory.

This section will offer a platform-based system of sentence generation. The platform-based system can be schematized as follows.

(13) The structure of the primary platform



The platform is a template comprised of the basic sentence elements supported by Universal Grammar. Structure (13) shows a primary platform comprised of the basic sentence elements: Comp, Tense, Neg, and Pred. Comp specifies the type of sentence. Tense specifies the tense of the sentence. Neg shows the negation of the sentence. Pred is a predicate that determines the argument structure.

The primary platform can accommodate up to one argument or one sentence at the maximum. In the case that the predicate is a two-place predicate requiring two arguments, the primary platform can be extended to the secondary platform, which accommodates the second argument. In the case that the predicate is a three-place predicate requiring three arguments, the primary platform can be extended to the secondary platform and tertiary platform, which can accommodate the second and third arguments, respectively.

In the case that the sentence is a complex sentence or compound sentence comprised of more than one clause, the primary platform, regardless of the valency of the predicate, can be extended to the secondary platform to accommodate the second clause.

Let us see how the platform-based system operates and meets the structural requirements of the binding principles with regard to sentences (6a), (6b), and (6c), repeated here as (14a), (14b), and (14c).

(14) .a. Bill will hate himself. (Bill = himself)

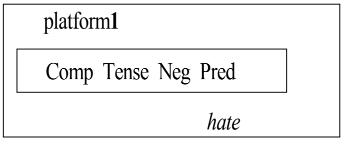
b. Maria will show the man to himself. (the man = himself)

c. Sue did not cry when she was bullied. (Sue = she)

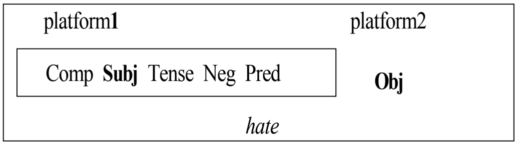
Let us begin with sentence (14a). Suppose that we are at stage 2, where we have the primary platform that accommodates the predicate *hate*.

(15) The primary platform and secondary platform

a. stage 2



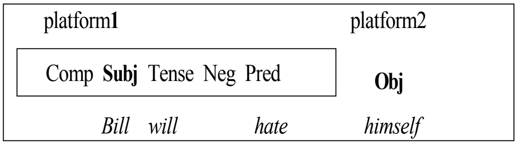
b. stage 3



Since the predicate *hate* is a two-place predicate, two argument positions are specified. One is the subject position in the primary platform, and the other is the object position in the secondary platform. The subsequent lexical insertion develops stage 3 to stage 4, where the primary platform and the secondary platform are both saturated with lexical items.

(16) The primary platform and secondary platform

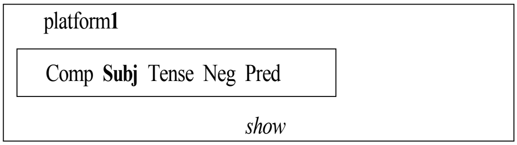
a. stage 4



In (16), *Bill* belongs to the primary platform, and *himself* belongs to the secondary platform. Since the primary platform is structurally superior to the secondary platform, it follows that *Bill* asymmetrically C-Commands *himself*. This satisfies the requirements of the binding principle A.[5)]

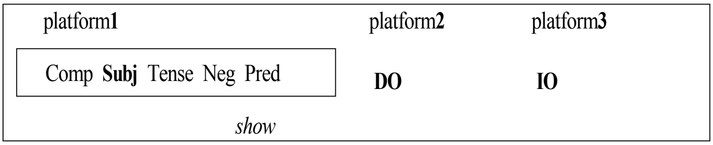
Sentence (14b) is a dative construction that has one external argument and two internal arguments. The sentence relates to both binding principles A and B. We will see how the proposed platform-based system satisfies the asymmetric C-Command requirement between the argument *the man* and the argument *to himself*. The numeration of sentence (14b) is NU = {Maria, show, will, man, the, himself, to}. The initial lexical support sets up stage 1, where the subject position is installed.

(17) Stage 1



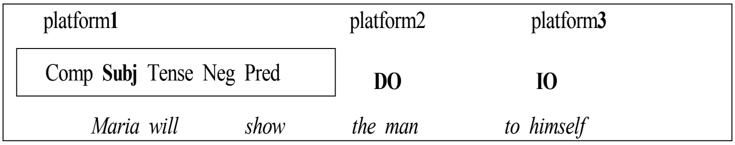
The dative predicate *show* is a three-place predicate that requires three arguments. Hence two extra platforms are installed that can accommodate the second argument and third argument, as shown below.

(18) Stage 2



The subsequent operation feeds each platform with the relevant lexical items and exhausts the numeration to zero. This process derives the final stage, as shown below.

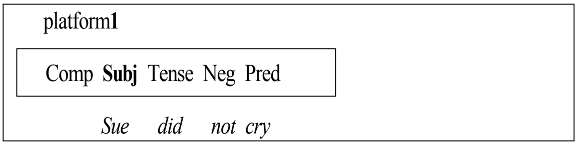
(19) Stage 3



In stage 3, there are three arguments. The external argument *Maria* belongs to the primary platform and the two internal arguments *the man* and *to himself* belong to the secondary and tertiary platforms, respectively. The internal argument *the man*, which belongs to the secondary platform, asymmetrically C-Commands the internal argument *to himself*, which belongs to the tertiary platform. This satisfies binding principle A.

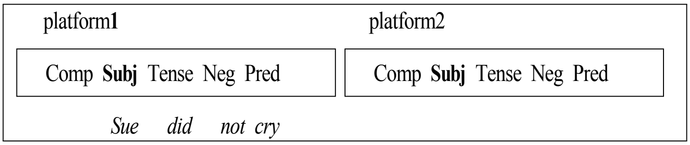
Sentence (14c) is a complex sentence comprised of a main clause and an embedded clause. This sentence relates to binding principle B. We will see how the platform-based system satisfies the structural requirements between the main clause subject and the embedded clause subject. The numeration of sentence (14c) is NU = {Sue, did, not, cry, when, she, was, bullied}. The derivation reaches the following stage by the time the primary platform is fully lexically saturated.

(20) Stage n



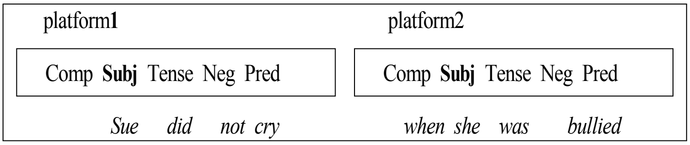
In stage n, the main clause that belongs to the primary platform is completed. In order to generate the embedded clause, we copy the structure of the primary platform and sets up the secondary platform, as shown below.

(21) Stage n+1



The stage in (21) reaches final stage n+2 when the secondary platform is fully lexically saturated, as shown below.

(22) Stage n+2



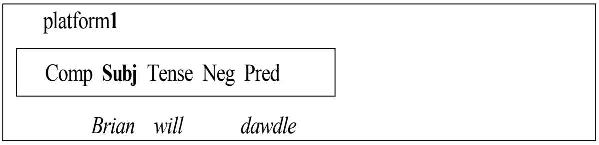
In (22), the main clause subject *Sue*, which belongs to the primary platform, asymmetrically C-Commands the coindexed embedded clause subject *she*, which belongs to the secondary platform. This does not violate binding principle B because *Sue* is outside the governing category of *she*.

So far we have shown how the proposed platform-based system provides legitimate syntactic structures for sentences (14a), (14b) and (14c). The syntactic structures generated by the platform-based system satisfy the binding principles without using categorial projection and slanted and vertical lines. This enables the computational system to avoid computational complexity and violation of the inclusiveness condition. Now, we will turn to sentences (10a) and (10b), repeated here as (23a) and (23b).

(23) a. Brian will dawdle, and he will miss the bus. (Brian = he)
b.*She will wake up, and Jina will eat breakfast. (She = Jina)

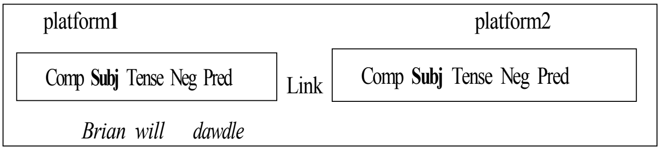
Sentences (23a) and (23b) are compound sentences comprised of two coordinate clauses. We will see how the platform-based system will account for the grammaticality of sentence (23a) and the ungrammaticality of sentence (23b). Let us begin with sentence (23a). Suppose that we have reached stage m, where the primary platform of the first conjunct is fully lexically saturated.

(24) Stage m



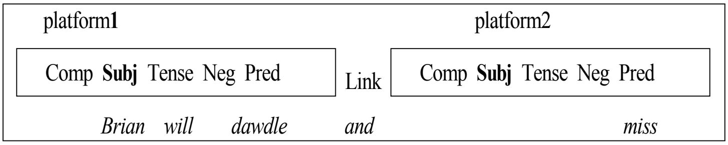
In stage m, the primary platform of the first conjunct contains three lexical items. Since the sentence is a compound sentence comprised of two coordinate clauses, we copy the structure of the primary platform and set up the secondary platform, shown below as stage m+1.

(25) Stage m+1



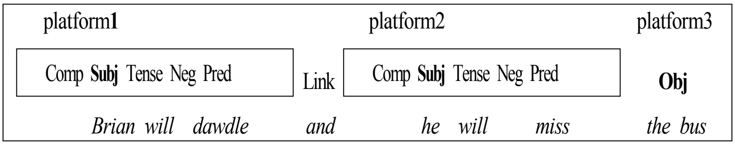
In stage m+1, we have the primary platform and the secondary platform, which are conjoined with Link. The subsequent initial lexical support for the secondary platform develops stage m+1 to stage m+2, as shown below.

(26) Stage m+2


platform**1**
platform2
Comp **Subj** Tense Neg Pred
Link
Comp **Subj** Tense Neg Pred
Brian will dawdle and miss

In stage m+2, the lexical item *miss* is a two-place predicate that requires two argument positions, and the object position is postulated in the tertiary platform and lexically saturate the secondary and tertiary platform to the maximum, as shown below.

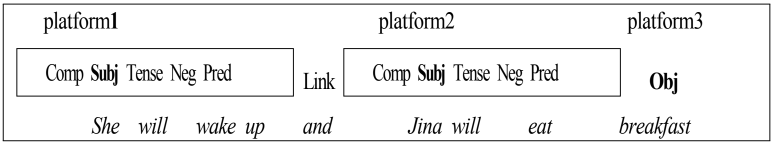
(27) Stage m+3


platform**1**
platform2
platform3
Comp **Subj** Tense Neg Pred
Link
Comp **Subj** Tense Neg Pred
Obj
Brian will dawdle and he will miss the bus

In stage m+3, the first conjunct subject *Brian*, which belongs to the primary platform, asymmetrically C-Commands the second conjunct subject *he*, which belongs to the secondary platform. Since the first conjunct subject *Brian* is not C-Commanded by the second conjunct subject *he*, binding principle C is not violated. The sentence is rightfully grammatical.

Next, we will see how the platform-based system accounts for the ungrammaticality of sentence (23b). Suppose that we have reached stage x, where the primary, secondary, and tertiary platforms are all set up with full lexical support, as shown below.

(28) Stage x



platform**1** platform2 platform3

Comp **Subj** Tense Neg Pred Link Comp **Subj** Tense Neg Pred **Obj**

She will wake up and Jina will eat breakfast

In stage x, the first conjunct subject *She*, which belongs to the primary platform, asymmetrically C-Commands the second conjunct subject *Jina*, which belongs to the secondary platform. This violates binding principle C, and the sentence is rightfully ungrammatical. Therefore, the platform-based system accounts for not only the sentences that X-bar theory can account for, but also the sentences that X-bar theory cannot account for. All of the aforementioned provide us with empirical reasons to believe that the platform-based system is a workable alternative to the framework of X-bar theory.

4. Conclusions

This study has discussed the framework of X-bar theory from a critical point of view. The framework of X-bar theory is problematic in three significant respects. First, it depends on categorial projection that causes computational complexity. Second, it uses vertical and slanted lines to show the unity of syntactic elements. This violates the inclusiveness conditions. Third, the syntactic structures formed by the framework of X-bar theory cannot account for the grammaticality and ungrammaticality of compound sentences with regard to the binding principles.

The platform-based system is a workable alternative to the framework of X-bar theory in three significant respects. First, it does not depend on categorial projection that causes computational complexity. Second, it does not use vertical and slanted lines to show the unity of syntactic elements. It shows the unity of syntactic elements by positioning them under the same syntactic position. Third, the syntactic structures generated by the platform-based system can account for the grammaticality and ungrammaticality of compound sentences with regard to the binding principles. All of these provide us with empirical reasons to believe that the platform-based system is a workable alternative to the current framework of X-bar theory.

Notes

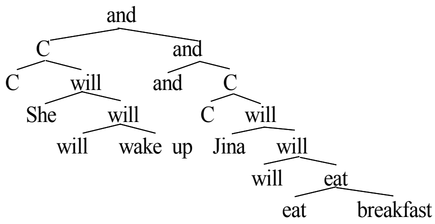
1) Chomsky's (1995) bare phrase structure differs from X-bar theory in that it dispenses with all categorial projections of lexical items. This avoids the violation of inclusiveness condition and reduces the burden of computational complexity. However, the elimination of categorial projection creates an issue in that it necessitates labeling of merged structures and notation to distinguish between substitution and adjunction.

2) Jackendoff (1977: 41) introduces three level projections, such as single bar projection, double bar projection, and triple bar projection. However, we will be confined to two level projections here.

3) Chomsky (1981: 166) defines the notion c-command in terms of containment. However, we will use the sisterhood-based notion of c-command as in (i).
(i) A node A c-commands its sisters and all the nodes that its sister dominates.

4) Suppose that we assume Chomsky's (1995) bare phrase structure. Then sentence (10b) will have the following bare structure without any categorial projections, such as:

(i)



Even under the bare phrase structure, the c-command domain of the first conjunct subject She, whose c-command domain is limited to its sister will and all the elements that will dominates, cannot c-command the second conjunct subject Jina. Therefore, we cannot account for why sentence (10b) is an unacceptable sentence even under the bare phrase structure.

5) In the platform-based system, the hierarchy between arguments is determined by the hierarchy of the platforms that the arguments belong to. The lexical items belonging to the primary platform asymmetrically c-command the lexical items belonging to the secondary platform and tertiary platform. In turn, the lexical items belonging to the secondary platform asymmetrically c-command the lexical items belonging to the tertiary platform.

Works Cited

Aoun, Joseph. *The Grammar of Anaphora*. Cambridge, MA: MIT P, 1985.

Baker, Mark C. *Incorporation: A Theory of Grammatical Function Changing*. Chicago: U of Chicago P, 1988.

Chomsky, Noam. "Remarks on Nominalization." *Readings in English Transformational Grammar*. Ed. R. Jacobs and P. Rosenbaum, 1970. 184-221.

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

Chomsky, Noam. "A Minimalist Program for Linguistic Theory." *View From Building 20: Essays in Linguistics in Honor of Sylvain Bromberger*. Ed. Kenneth Hale and Samuel J. Keyser. Cambridge, MA: MIT P, 1993. 1-52.

Chomsky, Noam. "Bare Phrase Structure." *The Principles and Parameters Approach to Syntactic Theory: A Synopsis*. Ed. Gert Webelhuth. Oxford:Blackwell, 1995. 395-439.

Chomsky, Noam. *The Minimalist Program.* Cambridge, MA: MIT P, 1995.

Chomsky, Noam. "Minimalist Inquiries: The Framework." *MIT Occasional Papers in Linguistics* 15 (1998: 1-61.

Chomsky, Noam. "Beyond Explanatory Adequacy." Unpublished manuscript. Cambridge, MA: MIT P.

Chomsky, Noam. "Three Factors in Language Design." *Linguistic Inquiry* 36 (2005): 1-22.

Chomsky, Noam. "Problems of Projection." *Lingua* 130 (2013).

Chomsky, Noam. "Problems of Projection: Extension." *Structures, Strategies and Beyond: Studies in Honour of Adriana Belletti*. Ed. Elisa Di Domenico, Cornelia Hamann, and Simona Matteini. Amsterdam: John Benjamins, 2015.

Jackendoff, Ray *X-Bar Syntax: A Study of Phrase Structure*. Cambridge, MA: MIT P, 1977.

국문초록

플랫폼 기반 문장 생성: 선과 이분지 병합을 사용하지 않고 통사체들의 단위성과 논항 간의 위계를 생성 방안

조 기 석

단독 / 사이버한국외국어대학교 교수

본 연구는 생성문법에서 통사체들의 단위성과 위계가 어떻게 나타나는지 논의하고 통사체들의 단위성과 위계를 나타낼 수 있는 새로운 방안을 제시하고자 한다. 생성문법에서는 현재 사선과 수직선을 사용하여 통사 체들의 단위성을 나타내고 있지만 사선과 수직선을 사용하는 것은 통사적 연산과정을 복잡하게 한다. 통사체들의 단위성을 나타낼 수 있는 새로운 방안을 제시하고자 한다. 또 생성문법은 이분지 병합을 사용하여 통사체들의 위계를 생성한다. 통사체들의 위계, 특히 논항들의 위계는 결속이론이 요구하는 구조적 특성이다. 하지만 이분지 병합에 입각한 위계는 첫째, 논항 간의 위계뿐만 아니라 거의 모든 통사 체들 간의 위계를 만들어낸다. 둘째, 병합에 입각한 위계는 주절과 종속절로 되어 있는 복문에서는 결속이론이 필요로 하는 위계를 만들어 낼 수 있지만 등위절로 되어있는 중문에서는 결속이론이 필요로 하는 위계를 만들어 낼 수 없다. 따라서 본 연구에서는 새로운 플랫폼 기반 문장생성 방식을 제안하여 이러한 문제를 해결한다.

주제어: 통사 체들의 단위성, 논항들 간의 위계, 성분통어, 플랫폼 기반 문장생성, 플랫폼 기반 성분통어

논문접수일: 2020.01.22
심사완료일: 2020.02.26
게재확정일: 2020.02.27

이름: 조기석
소속: 사이버한국외국어대학교, 영어학부 교수
주소: 서울 노원구 월계동 광운로 2나길 50 삼창아파트 나동 807
이메일: d9501001@hanmail.net