[REVIEW]

Spell-Out and the Minimalist Program

By Juan Uriagereka, Oxford University Press, Oxford, 2012, ix+349pp.

SATORU KANNO

*Hokkaido University of Education, Asahikawa**

Keywords: *multiple Spell-out, cycle, frustration*

1. Introduction

Juan Uriagereka's newly published book, *Spell-Out and the Minimalist Program*, examines the concept of derivational cycles, one of the central notions in linguistics, and its related notions based on the multiple Spell-out model that appeared in Uriagereka (1999). The mechanism of multiple Spell-out assumes the multiple application of the Spell-out operation in the course of the derivation of a sentence. Therefore, a linguistic expression is divided into some chunked pieces of information to be sent off to the Sensory-Motor (SM) and Conceptual-Intensional (CI) interfaces. The discussions in this book further develop the idea of multiple Spell-out and also cover the recent progress in biology, physics, neurology, anthropology, computational algorithms, and animal behavior. The combination of these adjacent fields of linguistics brings a new perspective to linguistic theorizing.

The guiding idea in this book is that language is in a state of frustration. Frustration, which is a notion often found in complexity and is used in various fields, occurs in “*the unresolvable co-existence of conflicting tendencies* that cannot settle into a steady state ...” (p. 224). Frustration in language emerges because language must meet two conflicting require-

* I would like to thank Akihiko Arano, Hiroaki Emoto, Yoshihisa Goto, Shin-ichi Kitada, Cornelia Daniela Lupsa, Nobuhiro Miyoshi, Tadao Nomura, Takahiro Tozawa, Juan Uriagereka, and Kazuhiko Yamaguchi for their valuable comments and suggestions. I would also like to thank two anonymous *EL* reviewers for their helpful comments. This study is supported by a Grant-in-Aid for Scientific Research (C) (No. 23520574). All remaining errors are my own.

ments at the same time: one is from DS, which constructs structures in a bottom-up fashion, and the other is from PF, which expresses structured information through a one-dimensional motor control (either gesture or utterance). Therefore, Uriagereka proposes that the structure of language is (1b), which contrasts with the conventional inverted Y-model in (1a).

(1) a. 'Inverted Y' model:

b. CLASH model:

In (1a), DS unidirectionally sends information to both PF and LF. However, as (1b) shows, grammar contains a tension between DS and PF, which causes the state of frustration. Uriagereka calls this proposed mechanism the CLASH model. CLASH is an acronym for Conditions Liberating A Simple Hiatus.

Uriagereka argues that due to the state of frustration, cyclicity emerges. As the reason, he says “the way to reconcile this tension is in terms of limiting the size of possible ‘talk between syntax and phonology’ to relevant cycles” (p. 271). In simpler terms, Syntax and PF can “communicate” with each other through the factorized information, and this is called cycles.

He also says “... what seems unique to the human species is the connection, precisely at the CLASH point(s), between different brain systems that are almost certainly present in other animals” (p. 279). This means that Uriagereka attributes unique aspects of human language (such as cyclicity) to the CLASH point(s) where opposing tendencies conflict and frustration occurs. Even though some forms of frustration are also found in the brain of birds due to the conflicts between different brain systems, these brain systems are different from the human ones. Hence, the uniqueness is attributable to the CLASH point(s).¹

As Uriagereka himself admits, many concepts discussed in this book are relatively familiar to linguists. However, the important contribution of this book is that these concepts are illustrated from the CLASH approach in (1b). Therefore, this book can be considered as a theoretical defense of concepts that are often mentioned and well-known but used without fundamental support from the biolinguistic base. This book also has far-reaching

¹ I thank an anonymous reviewer for clarifying this point.

theoretical consequences emerging from this line of argumentation.

This review consists of four sections. The next section summarizes the discussions of Chapters 1 to 5. These chapters deal with various ideas related to cyclicity. Section 3 focuses on the discussions of Chapters 6 and 7, both of which are central parts of this book. Section 4 is the conclusion of this review.

2. Main Ideas in Chapters 1 to 5

Chapter 1 focuses on linearization. A fairly widely held view is that language has a hierarchical structure. However, multi-dimensional structures must be translated or changed into one-dimensional speech or gestures. Thus, if language is to be usable, we face the linearization problem; that is, “we squeeze a lot of information into simple motor units,” and Uriagereka calls this problem of linearization “the Squeezing problem” (p. 49).

He says that there are two ways to solve this problem. One is linearization based on the LCA (Linear Correspondence Axiom), which adopts (2) and (3) as theoretical assumptions.

- (2) When x asymmetrically c -commands y , x is linearly ordered with regard to y . (p. 55)
- (3) If x is linearly ordered with regard to y , then x precedes y . (p. 55)

The other approach to linearization is the Mirror Linear Correspondence Axiom (MLCA), which assumes (4) instead of (3).

- (4) If x is linearly ordered with regard to y , then x follows y . (p. 55)

Both the LCA and the MLCA create a natural homomorphism between structures and orders. However, Uriagereka offers arguments for the LCA and against the MLCA from the perspective of parsers.

If we are given a parser with the LCA, this parser first meets elements such as an operator and a probe when it is parsing an utterance. It then searches for an appropriate variable and goal until it finds them. Importantly, a parser of this type first meets an item that needs to be satisfied in some way or other in the derivation. On the other hand, a parser with the MLCA first meets an element that shows an equivocal behavior such as a pronoun and a goal. These elements may or may not enter into a binding or an Agree relation. In this case, the parser cannot make an immediate decision on the status of these elements. This indetermination causes a

problem for the parser because “holding the variable in memory waiting for the operator-that-never-arrives is a waste of resources” (p. 65). Therefore, Uriagereka concludes that the LCA is preferable.

The argument here has two important implications. First, since a parser with the LCA must meet an operator before its corresponding variable, a specifier comes first or occurs on the left of its head. Second, Spell-out must be applied multiply when XP in a specifier has a complex structure. If a specifier is complex, it is treated as one chunk, separately to the other parts. As a result of the multiple application of Spell-out, a c-command relation holds in every element within each chunk. To the extent that the c-command relation holds within every element, they are linearizable at PF.

Chapter 2 discusses sub-extraction. When a subject consists of more than one element, we have two theoretical possibilities: (i) that “it can behave only as a part-less whole” (p. 89), and (ii) that the subject exists with no relation to the other main P-marker (called a main current). Uriagereka calls the former the conservative Spell-out, and the latter the radical Spell-out. With either possibility, it is expected that sub-extraction is possible only for complements, which appear only in main currents. Subjects and adjuncts resist the application of sub-extraction since they are not in the main currents.

The idea of sub-extraction is supported by the experiment by Jurka (2010). Jurka has presented the data carefully, controlling irrelevant structural factors such as prolepsis and pro-drop, and asked the grammaticality of sentences using a seven Likert scale. The results of the experiment correspond to what the multiple Spell-out model expects; extraction out of subjects (moved or non-moved) degrades grammaticality. Furthermore, sub-extraction is possible only when an object stays in a main current. These results show that sub-extraction applies to an item in the main current.

The main point of Chapter 3 is that c-command relations are a consequence of the multiple Spell-out architecture that this book assumes. As we have already seen, Spell-out must be applied multiply (when a specifier is complex) for the purpose of linearization. Therefore, the requirement for externalization derives the c-command relation of each item. Pursuing this line of argumentation, a large part of this chapter is devoted to demonstrating that the c-command relation is hard to derive from semantic, phonological, and certain other syntactic conditions.

Chapter 4 discusses various topics related to cyclicity. First, Uriagereka argues that cyclicity can be found within two-word compounds. Specifically, the two words *board* and *white* appear symmetrically at the first stage of

the derivation, but *white* moves to the left of *board* in order “to break the symmetry and yield a valid linearization at PF” (p. 164). Here, Uriagereka adopts a process called roll-up in Barrie (2006) and applies this mechanism to the analysis of compounds. As a result of the roll-up movement, we get the asymmetrical compound: [*white*- [*board* t]].²

Second, he also discusses cross-linguistic variations on the Spell-out strategy. As we have seen, the radical Spell-out mechanism assumes a spelled-out item to have no relation to the main current, while the conservative mechanism, on the other hand, requires it to stay on the main current as a large word-like chunk. If a language adopts the radical strategy, this language should have the null element *pro* that serves as “a glue” to connect a specifier with a main P-marker or main current. On the other hand, if a language adopts the conservative strategy, the specifier stays on the main current in a flattened fashion. From the discussion, it is expected that if *pro* is used, the reconstruction effect would not be observed since *pro* simply works as a connector for a specifier that lives in a different derivation from the main current. On the other hand, if *pro* cannot occur, the reconstruction effect should be observed. Uriagereka argues for the correctness of this prediction, using Spanish and Basque. Spanish has *pro* in the subject position, but not in the object position. In this language, the reconstruction effect is only observed in the movement of objects. Moreover, Basque, which has both a subject *pro* and an object *pro*, shows no reconstruction effect.

Third and finally, Uriagereka discusses the derivation of adjuncts. His main idea is that they are “free floating” with no relation to the main current (p. 179). For example, no element within an adjunct enters into a binding relation with the one in the main current. This is because “... the adjunct lives its own structural life, so much as that it doesn’t get affected by standard binding considerations” (p. 183). Uriagereka argues that the idea here is similar to Chametzky’s (1996) analysis, which claims that adjuncts have no label in the derivation.

Chapter 5 shows that the multiple Spell-out mechanism is applied to PF and LF. Uriagereka discusses the phenomena of salvation-by-deletion as a piece of evidence for multiple Spell-out on the PF side. Lasnik (2001) and Merchant (2001) argue that the violation of movement out of an island can be nullified by deleting the relevant part of the sentence. Uriagereka

² I thank an anonymous reviewer for clarifying the derivation of compounds.

assumes that an island violation arises when a sentence contains some parts that are not linearizable. Thus, when unlinearized parts are deleted, the resulting sentence becomes qualifiable at PF. Uriagereka further cites Hornstein et al. (2007), which says "... the MSO [= Multiple Spell-Out] approach provides a reasonable understanding of the ‘*’ diacritic and its elimination" (p. 199).

The multiple Spell-out approach relevant to LF is the mechanism called reprojection, which re-labels the phrase-marker (typically having a quantifier), as shown in (5).

The change of a label is banned in the representational model. However, the MSO architecture opens the possibility of a change after the syntax, that is, LF.

3. CLASH Model and Its Consequences in Chapter 6 and Chapter 7

Chapter 6 introduces dynamical frustration and shows how three manifestations of dynamical frustration are related to the neurophysical substrate of language. Uriagereka bases most of the discussion in this chapter on Binder's (2008) idea that dynamical frustration manifests itself in three forms: scale, geometric, and computational. Moreover, what Uriagereka pursues and develops in this chapter is Binder's intuition that "emphasizes the abstract connections between three forms of frustration ..." (p. 39). Specifically, Uriagereka focuses on two connections or translations: "one directed towards externalization and one in the direction of the workings of a computational mind" (p. 249). The former translation appears in the changing of scale frustration to geometrical frustration, and the latter connection emerges through the translation of scale frustration to computational frustration.³

First, let me summarize the discussion on the relation between scale frus-

³ As Binder himself admits, his work is very preliminary, and many precise details remain to be fully investigated. However, needless to say, this does not undermine the value of Uriagereka's attempt to extend Binder's ideas to linguistics.

tration and geometrical frustration. By examining birdsongs, Uriagereka shows that, in birds, the brain physiology underlying birdsong can be stated in terms of scale frustration, which arises due to two opposing tendencies: neurons controlling expiration by the elevation of air pressure, on the one hand, and ones controlling the inspiration between song elements, on the other. He says “[t]hat aspect of bird thinking ... is what ought to be seen as an instance of scale frustration” (p. 248). Therefore, scale frustration is an internal state underlying the brain physiology in birds. Importantly, this internal structure is externalized in the form of geometrical frustration that is computable in a linear fashion. Therefore, the fact that translation is possible in a mutual direction between scale frustration and geometrical frustration “establishes an equivalence between a set of unidimensional chirping series and some multidimensional dynamically frustrated system” (p. 249). Thus, we can find a correlation between scale frustration at the neurophysical level and geometrical frustration at the externalized level.

Next, Uriagereka further discusses the other connection among the three forms of dynamical frustration: the connection between scale and computational frustration. He shows that the physiological substrate of the birdsongs expressed in scale frustration is captured in terms of computational frustration. This correlation between the two forms of frustration is important because computational complexity signals forms of thought. When birdsongs are considered from the computational perspective, their computational complexity can be captured in terms of the Chomsky Hierarchy. Uriagereka argues, based on Trevisan et al. (2006), that the computational complexity of the birdsongs should be in between finite state automata and pushdown automata. Therefore, it is expressed in-between regular languages and context-free languages.

Chapter 7 articulates his CLASH model in (1b). His main idea is that grammar is in a state of frustration. Frustration arises when two independent or orthogonal tendencies co-exist in an unsolvable way. Thus, in grammar, the phonological requirement for the purpose of linearization conflicts with the structural requirement to build sentences in a multidimensional way. One of the consequences of the dynamic frustration model for language is cyclicity, as we have seen in section 1 of this review.

This chapter also focuses on phonological and syntactic domains. In these domains some essential properties emerge as a consequence of frustration. Although the author offers five instances, space considerations force me to offer only one of them, which concerns the syllabic patterns in lan-

guage.⁴

He first sets out the following three rules that bring out Fibonacci patterns:

(6) *F Game*

- (i) Starting with either a + or −,
- (ii) Go on to concatenate it to another + or a −, with one condition:
- (iii) Avoid combining identical symbols, unless they are adjacent to a different symbol. (p. 281)

A Fibonacci pattern is an integer sequence following the formula $F_n = F_{n-1} + F_{n-2}$, with the proviso that the first two numbers are 0 and 1. The pattern produces the following integer sequence: 0, 1, 1, 2, 3, 5, 8, 13.... Uriagereka considers the rules in (6) to be a repulsive force, because we get elements that increase indefinitely in Fibonacci patterns by following the rules in (6). Additionally, he also offers the following rules to make groups within each element:

(7) *Linguistic Conditions*

- (i) Nucleus Constraint: Look for a maximal space. Then
- (ii) Onset Constraint: Try to assign an onset boundary to that space.
- (iii) Coda Constraint: Try to assign a coda boundary to that space. (p. 282)

The rules in (7) are linguistic conditions, and they are considered as an attractive/gluing force, a force that makes some groupings within each element that is produced by the rules in (6). In other words, each element produced by (6) is made from a combination of the following six chunks:

- (8) a. + − b. + − − c. + − + d. + − − + e. − f. − + (p. 282)

If we assign + as a consonant and − as a vowel, the syllabic patterns emerge in (9).

- (9) CV (and its stressed variant CVV), CVC (and its stressed variant CVVC), V, and VC (p. 280)

⁴ The discussion of this chapter simplifies actual linguistic phenomena, abstracting away from many details. The reader should read this chapter as a first attempt to derive linguistic properties from the other fields such as physics and physiology. However, I also hasten to add that the research effort here actually marks a profound step in the current understanding of language and its adjacent fields. Therefore, the question of whether the claim here is correct or not is an empirical issue to be addressed in further research.

(9) shows possible syllabic patterns in language. Remember here that the items in (8) and the phonological variants in (9) are a result of frustration, because these items are produced to solve the two conflicting tendencies: (6) as a repulsive force and (7) as a gluing force. Thus, syllabic patterns found in language occur as the products of frustration.⁵

After laying out four other Fibonacci patterns as instances of the existence of frustration in language, Uriagereka further shows that the CLASH approach can predict possible structures in syntax. For example, when we construct a tree structure, strictly obeying the X'-format, we get (10).

We again find Fibonacci patterns in the number of the maximal projections, the intermediate projections, and the heads. These numbers grow indefinitely in this pattern. The reason for the presence of the patterns in structures like (10) is that the Merge operation acts as a repulsive force. Therefore, the language faculty contains the self-organizing algorithm that produces structures like (10), and the driving engine is the Merge operation, which has the following properties: (i) binary, (ii) labeling, (iii) structural preservation, (iv) unboundedness, and (v) flexibility (p. 291). As a result of these properties, Fibonacci patterns emerge in structure-building. Thus, he says

⁵ His argumentation crucially relies on the assumption that Fibonacci patterns are ubiquitous not only in the animal kingdom but also in the overall body-plan of plants, and that it is natural to observe the patterns in language. At this point, Juan Uriagereka (personal communication) has suggested to me that there are three possible reasons why Fibonacci patterns are observed in language. First, the Fibonacci effect was created in the brains of pre-sapiens as a result of the clash between DS and PF. It then became a primitive character in our current brains. In this view, Fibonacci patterns are not primitive from an evolutionary perspective. The second and more plausible possibility that he has suggested is that frustrated conditions between DS and PF exist for the brains of babies, but within years or months after birth, as the brains mature, LF is stabilized. The final and less plausible but possible explanation for the presence of Fibonacci patterns in language is concerned with performance. Processing sentences causes a clash between DS and PF. The important point is to understand that Fibonacci patterns are very common, not only in language, but also in the natural world.

“[n]one of these properties [of Merge] need to be blamed on ‘interface conditions’” (p. 293). Rather, they are a result of “the internal coherence of the system that generates it” (p. 293). Therefore, the properties of the Merge operation do not arise from the so-called interface conditions, but from the self-organizing algorithm that offers Fibonacci patterns.

Finally, Uriagereka examines the periodicity of phase discussed by Richards (2007), who claims that a phase (P) and a non-phase (N) appear in the P-N combination: $[_P [_N [_P [_N [_P [_N \dots]]]]]]]$. In the rules of (6), the most frequently observed patterns are the $+ -$ pattern in (8). Therefore, if we assign $+$ as P and $-$ as N, we can get the P-N pattern, which is the most common periodicity of phase in language. Again, the patterns of the P-N combination are also accounted for by Fibonacci patterns.

Moreover, given that the P-N patterns appear as a central pattern of the periodicity of phase, we have an explanation of why v and C are phases, but not V and T. Since syntactic derivation proceeds in a bottom-up fashion, the lower heads such as V and T work as non-phases (N) in the P-N pattern, and the higher merged heads such as v and C serve as phases (P).

4. Conclusion

The discussions in this book should have a far-reaching impact on linguists and researchers in its adjacent fields. For linguists, this book is worth reading because of its extensive coverage of empirical data and theoretical consequences, all of which are centered on cyclicity, one of the central ideas in linguistics. Moreover, this book is recommended for non-linguists, too, because it makes clear the issues in the relation between language and biology/cognitive science. In the recent Minimalism, a great deal of attention has been paid to the third factor condition. This book is an insightful, ambitious, and fruitful achievement in pursuing this line of research.

REFERENCES

- Barrie, Michael (2006) *Dynamic Asymmetry and the Syntax of Noun Incorporation*, Doctoral dissertation, University of Toronto.
- Binder, Philippe M. (2008) “Frustration in Complexity,” *Science* 320, 322–333.
- Chametzky, Robert A. (1996) *Phrase Structure: From GB to Minimalism*, Blackwell, Oxford.

- Hornstein, Norbert, Jairo Nunes and Juan Uriagereka (2007) "The Dynamics of Islands: Speculations on the Locality of Movement," *Linguistic Analysis* 33, 149–175.
- Jurka, Johannes (2010) *The Importance of Being a Complement: CED Effects Revisited*, Doctoral dissertation, University of Maryland.
- Lasnik, Howard (2001) "When Can You Save a Structure by Saving It?" *NELS* 31, 301–320.
- Merchant, Jason (2001) *The Syntax of Silence*, Oxford University Press, Oxford.
- Richards, Mark (2007) "On Feature Inheritance: An Argument from the Phase Impenetrability Condition," *Linguistic Inquiry* 38, 563–572.
- Trevisan, Marcos A., Gabriel B. Mindlin and Franz Goller (2006) "Nonlinear Model Predicts Diverse Respiratory Patterns of Birdsong," *Physical Review Letters* 96, 058103.
- Uriagereka, Juan (1999) "Multiple Spell-Out," *Working Minimalism*, ed. by Samuel D. Epstein and Norbert Hornstein, 251–281, MIT Press, Cambridge, MA.

[received December 11 2012, revised and accepted April 27 2013]

Department of Education
 Hokkaido University of Education, Asahikawa
 9 Hokumon-cho, Asahikawa-shi
 Hokkaido, 070–8621
 e-mail: kanno.satoru@a.hokkyodai.ac.jp