1. Introduction

Hornstein’s newly published book, *A Theory of Syntax*, is full of provocative proposals and insightful suggestions which every scholar should be interested in if they are serious about syntax and concerned about which directions the current minimalism might go. In this review, I would like to discuss and summarize some of his major proposals. Hopefully, this discussion will demonstrate Hornstein’s contribution to the field as well as my own enthusiasm for this interesting work.

It has been more than a decade since *The Minimalist Program* (Chomsky 1995) was published. There is no doubt that the ideas and the proposals in the book had a huge impact on the theory of syntax. The fact that numerous books and articles have been published from the perspective of the Minimalist Program (MP) indicates that there is a growing interest in the field, and the subsequent framework shift has attracted much attention. Since the significance of the shift (or transition) from the GB (Government and Binding Theory) framework to MP has been discussed elsewhere (e.g. Hornstein, Nunes and Grohmann (2005), Lasnik, Uriagereka and Boeckx (2005)), the full details of the discussion will not be repeated here. However, I should emphasize that the arrival of Hornstein’s latest work may make another mark on the shift in perspective; hence it may be well worth while to discuss a few points that link the GB framework to MP and to the ideas proposed by Hornstein in his book.

* I would like to thank George Truscott and Rosalynn Chiu for proofreading the manuscript and suggesting stylistic improvements. I am also grateful to two anonymous *EL* reviewers, for their invaluable comments and suggestions. All remaining errors are my own.
Under the GB framework, researchers working on syntax have uncovered a large number of generalizations and interesting facts about human language. Various rules that allow us to describe the patterns were proposed, and each of them has been revised as a new fact arrives. As Hornstein mentions in the book (p. 6), those discoveries have been a great contribution in the GB framework, and of course the generalizations are still very important even after the shift of the framework. Notions such as ‘simplicity’ and ‘economy’ characterize two of the many concepts introduced (or re-emphasized, to be more appropriate) upon the arrival of MP.

With respect to ‘simplicity,’ many examples showing its importance can be found in the history of generative grammar. For instance, as shown in Chomsky (1977), Topicalization and Relativization are not separate rules, but these two are unified under Wh/A’-movement. As for ‘economy,’ it is fair to say that MP introduced the notion of economy as a new guideline to determine which analysis would be the best among the competitors. Also, the introduction of the notion of economy showed us clearly that in so far as the Faculty of Language is a system in human cognition, this is a property by which any cognitive system must abide. In other words, the theory of grammar in the GB framework no doubt contains a number of mechanisms that are highly specific to language where the choice between the competing hypotheses can be made only based on linguistic data. However, MP tells us that something that is not specific to language, such as economy, will also carry a vote on the choice.

Now, what Hornstein argues in this book is that there is yet another new guideline (or a piece of potential new evidence) that syntacticians must be concerned with. This is why I suggest that this book might make another mark of the shift in perspective. As will be discussed below, Hornstein argues that it is not satisfactory to propose a system that is too specific to language, since such a system (the Faculty of Language) is evolutionally unreasonable. Even though what we know about the evolution of language is still rather small, a grammar must meet evolutionary guidelines. Because of this perspective, it is quite certain that this book is a great cornerstone from which a lot of books and articles will be produced to examine the idea and eventually put it forward.1

1 There are numerous recent books and articles linking to biolinguistics that share Hornstein’s view, although the details of the proposals are of course different. The increasing interest in this perspective is also clear from the newly published journal Biolinguistics.
2. Organization of the Book

Chapter 1 lays out the background of the book. There is one big research question clearly set forth at the beginning of the book: *How should a grammar be structured?* Hornstein says that the grammatical operations of UG should be simple, and uses several pages to demonstrate what counts as simple in a system like a grammar. First, simple grammars are non-redundant. Second, a grammar with fewer basic operations is better than one with many. Third, the basic operations in a grammar should be “natural.” Yet why should the mechanism involved in a grammar be “natural”? Hornstein starts the discussion from an evolutionary assumption. The Faculty of Language, which is one distinct component in our cognitive system, is commonly estimated to have evolved in humans in the last 50,000–100,000 years (p. 4). Such a short period of time in evolution leaves an extremely small window for something to have been invented specifically for the Faculty of Language. Consequently, a minimalist investigation is pursued in an attempt to figure out which part of the Faculty of Language is linguistically specific and which part is shared with other cognitive computations. Hornstein names this investigation “Darwin’s Problem”:

(1) Darwin’s Problem: What must be added to the inventory of pre-linguistic cognitive operations and principles to deduce the principles of UG?

The subsequent chapters in Hornstein’s book are organized around Darwin’s Problem. In Chapter 2, he takes up c-command, a relation that is found everywhere in the grammar. He argues that c-command demands an explanation regarding why it is so involved in the grammar. Hornstein claims that c-command is more of a by-product of a grammar that has particular features than anything else. Chapter 3 deals with Merge. Hornstein argues that Merge is a combination of Concatenate and Label. Chapter 4 further explores consequences of the proposal on Label in Chapter 3. Specifically, he shows how the proposed system handles adjunction. Chapter 5 summarizes findings from Chapters 2 to 4. Chapter 6 illustrates how the present approach does not fit well with a modern minimalist account, namely an AGREE-based system. Chapter 7 concludes the book.

3. C-command

Chapter 2 deals with c-command, which we know is exploited everywhere in the grammar. Hornstein discusses the following three domains that in-
volve c-command crucially: Binding, Linearization, and Minimality. The aim of this chapter is to account for the reason why c-command plays a crucial role in the grammar, and to show that “c-command is a necessary by-product of some kinds of Merge-based grammars” (p. 19). The space allotted for this Review only allows me to discuss two cases: Binding and Linearization.

3.1. Binding

There is a common observation that the distributions of anaphors and traces left by A-movement are quite similar (see (2) and (3) below). Both anaphors and A-traces must be c-commanded and co-indexed by their antecedents. In the present context, c-command is crucially invoked to license anaphors and A-traces. One way to capture this commonality, which Hornstein proposed earlier, is to assume that the anaphor is a residue of overt A-movement (Hornstein (2001)).

(2) a. John believes himself to be tall  
   b. *John believes himself is tall  
   c. *John would prefer for Mary to like himself

(3) a. John was believed \( t \) to be tall  
   b. *John was believed \( t \) is tall  
   c. *John would preferred for it to be seen \( t \) (= John would be preferred to be seen)

An approach like Hornstein (2001) provides an account of why we find c-command everywhere in the grammar. The c-command requirement in various phenomena indicates that the same operation (i.e. movement) is employed in each of these cases. Now the new task that arises from such an approach is to figure out why the phrase that undergoes movement always c-commands its previous positions.\(^2\) According to Hornstein, movement has this property if it obeys the Extension Condition, illustrated below.

\[
\begin{array}{c}
\alpha \ldots X \ldots \rightarrow \[\beta \rightarrow [\beta \rightarrow [\alpha \ldots X \ldots] \ldots Y \ldots]]
\end{array}
\]  
(p. 22)

The schema in (4) illustrates a case where two constituents, \( \alpha \) and \( \beta \), merge and create a larger structure. Note that the Extension Condition was satisfied since the resulting structure contains the original \( \alpha \) and \( \beta \) as its subparts. Thus, a phrase that undergoes movement (which is an instance of

\(^2\) Although the illustration above is a case of A-movement, note that this c-command requirement on movement is not limited to A-movement, but includes A′-movement, as well.
Merge) or merges at the root of the structure necessarily c-commands the rest of the structure. Hornstein concludes, “the requirement that a moved expression c-command its launch site follows from the Extension Condition” (p. 22).

### 3.2. Linearization

Another domain that crucially involves c-command is the linearization operation, which turns hierarchically organized phrases into left-to-right sequences. The standard assumption is that a version of the LCA (Linear Correspondence Axiom) (Kayne (1994)) must be a part of the grammar to do the job (see also Nunes (2004)). The most important part of the LCA is asymmetric c-command, since the algorithm will linearize \( \alpha \) before \( \beta \) if \( \alpha \) asymmetrically c-commands \( \beta \). Hornstein notes that the “c-command” part of the system is in fact not so crucial, but the “asymmetric” part is. He further argues that there is another asymmetric relation we can encode in the grammar to linearize linguistic structure; he proposes that Merge is asymmetric and that a new version of the LCA that linearizes \( \alpha \) before \( \beta \) if \( \alpha \) has merged with \( \beta \) is preferred. Note that “\( \alpha \) has merged with \( \beta \)” is different from “\( \beta \) has merged with \( \alpha \).” Here is a concise illustration that derives the word order *John likes her*.

\[
\begin{align*}
\text{(5) a. Merge } & \text{her with likes: [her likes]} \\
\text{b. Merge } & \text{v with [her likes]: [v [her likes]]} \\
\text{c. Copy } & \text{likes and merge with v: [likes+v [her likes]]} \\
\text{d. Merge } & \text{John with [likes+v [her likes]]: [John [likes+v [her likes]]]}
\end{align*}
\]

If this approach is in the right direction, there is no need to keep the notion of c-command as a key component in the grammar. Linearization is possible without relying on asymmetric c-command, but rather relying on asymmetric Merge.

### 4. Label

The aim of Chapter 3 is to show that Merge has two independent op-

---

\(^3\) Of course, there has to be a way to determine what is a thing to merge, and what is a thing to be merged with. Hornstein suggests that the way features are checked off upon merger would provide a principled account and it is also possible to encode the head parameter in the system. This can surely be explored further in future research.
operations, Label and Concatenate. Hornstein argues that operations such as Concatenate are found in other cognitive systems, but Label is an operation that is specific to the Faculty of Language.  

Hornstein started his discussion from an often-noted property of linguistic objects, namely recursive embedding. In modern minimalist accounts, Merge is responsible for building syntactic structure, and the repetitive application of Merge renders recursive embedding. One important feature in the recursive embedding is that Merge can handle not only a lexical item but also a set as a part to be merged. In other words, a lexical item such as the and dog can be an input in the Merge operation, and a set \{the, dog\} also counts as an input in the operation. Second, as we have already seen above, it is stipulated that Merge applies only at the “root.”

To illustrate Hornstein’s proposal, let us consider concatenation first. The operation Concatenate is defined as below, and is said to be short for recursive embedding since it does not render a hierarchical structure.

\[
\text{Concatenate: } A, B \rightarrow A^B: C, A^B \rightarrow C^A^B
\]

However, it is certain that the operation is recursive; there is no upper bound on the number of its application. Another important feature of Concatenate is that, in (6), \(A^B\) is not taken as atomic, although each of its parts, \(A, B\), is. Once \(A^B\) is taken as atomic, and is liable for further concatenation, hierarchy emerges. Hornstein then argues that Merge is a species of concatenation. In addition, “labeling, understood as it is within Bare Phrase Structure, supplies the necessary ingredient to get one from a flat beads-on-a-string system to a hierarchical nesting system” (p. 55).

\[
\text{Label } A^B \rightarrow [A, A^B]
\]

The resulting structure should be read as the complex object \(A^B\) that has a label \(A\), which is again a concatenatable atom. Given the label \(A\), the complex has all the properties that the lexical item \(A\) has. In other words, labeling is understood as having the “is-a” relation (see also Boeckx (2006)).

The proposal advocated here has several important consequences. First, the combination of Concatenate and Label provides an account for the endocentricity constraint under which selection and subcategorization are restricted to head-to-head relations. Put differently, Hornstein suggests, if “all

---

4 It is possible to consider Hornstein’s proposal as a response to the frequently noted claims advocated by scholars such as E. Bates (Bates (2004), for instance), where she argues that there is nothing specific to language, and properties of language can be derivable from domain-general cognitive architecture. See Marcus (2009) for a further discussion on this point and related pieces.
inter-lexical relations were parasitic on concatenation, then a head X could select/subcategorize Y only if X concatenated with Y” (p. 61). When C concatenates with $[A \ A^B]$ (a complex with label A), the complex is an atom A, and only the label A is visible, which is the same as the head of the complex. Second, the proposal that only the label is visible upon Merge can provide an account for the Extension Condition. Merge applies at the root not because it is stipulated as such, but because the label at the root is only visible for the operation Concatenate. Hence, the Extension Condition is derived if Merge is considered as the combination of Concatenate and Label. Interestingly, this line of analysis indicates that Move is motivated only when the element needs to check more than one feature. The more features it has to be checked, the more it has to be concatenated with different heads.

5. Adjunction

In Chapter 4, Hornstein further explores the consequences of the theory of phrase structure proposed in Chapter 3, focusing on adjunction. Here, he examines in detail how adjunctions should be handled in syntax. First he reviews some general properties of adjunction. It is well known that adding adjuncts does not alter selection properties or subcategorization relations of the head. Thus, the relation between the auxiliary verb *have* and the participle form of the verb *eaten* is not disrupted by *quickly* or *in the yard* in the example below.

(8) a. has/*is $[VP \ \text{eaten a bagel}]$

   b. has/*is $[VP \ [VP \ \text{eaten a bagel}] \ \text{quickly}] \ \text{in the yard}$

Let us look at how Hornstein handles the case above in his theory of phrase structure. Given that the label represents the “is-a” relation, X concatenated with YP is an X in (9). Also, the complex $[X \ X YP] \ \text{concatenated with WP}$ is also an X.

(9) $[X \ [X \ X YP] \ WP] \ ZP$

Note that this can capture one of the important features of adjunction in that it does not change the selection relations among heads. Structures with adjunct elements behave exactly the same as those without adjunct elements, as clearly predicted from the label.

As illustrated in the example below, it is of interest to see how the sys-

---

5 Hornstein notes the material in this chapter is based on joint work with Jairo Nunes.
tem handles VP movement. It has been known that only XP categories can undergo movement, and the observation below indicates that \[\text{kick Fred}\] as well as \[\text{kick Fred in the yard}\] is a VP, but \[\text{kick}\] is not.

\[(10)\]

a. It was kick Fred that John did  
b. It was kick Fred that John did in the yard  
c. It was kick Fred in the yard that John did  
d. *It was kick that John did Fred

This observation seems to be unexpected under the new theory of phrase structure. If we follow the schema in (9), the label for \[\text{kick Fred}\] would be the same for \[\text{kick}\], but then the pattern in (10) is quite perplexing.

Hornstein claims that the movement of \[\text{kick}\] is unacceptable due to a violation of the A-over-A condition. The A-over-A condition prohibits the movement of \[v \text{kick}\] since it is part of the larger structure of the same type, namely \[v \text{kick Fred}\]. This solution, then, invites a further question: Why is it possible for \[v \text{kick Fred}\] to move though it seems to be part of \[v \text{kick Fred in the yard}\]? Hornstein answers this question by arguing that the system allows cases where no label is provided after concatenation, and that adjuncts in fact allow such an option. According to Hornstein, when objects are concatenated but no label is provided to the object, the resulting structure “is not an atomic object and so cannot be input for further concatenation” (p. 90). For instance, when \[\text{eat the cake}\] undergoes movement as in (12a), the structure must be like (11a) where the adjunct is not part of the labeled phrase. Also, when \[\text{eat the cake in the yard}\] undergoes movement as shown in (12b), the adjunct \text{in the yard} is part of the larger structure as in (11b).

\[(11)\]

a. \[v \text{eat}^\text{the-cake}\] in-the-yard  
b. \[v \text{eat}^\text{the-cake}\] in-the-yard

\[(12)\]

a. \text{eat the cake he did in the yard}  
b. \text{eat the cake in the yard he did}  
c. *\text{eat he did the cake in the yard}

Note that (12a) cannot be derived from (11b) since it would violate the A-over-A condition. Finally, under this system, the A-over-A condition is also responsible for the unacceptability of (12c). Assuming that “complements must be inside labeled concatenates and adjuncts need not be” (p. 90), movement of \text{eat} leaving the complement \text{the cake} behind would violate the A-over-A condition.
6. AGREE and the Existential Construction

In Chapter 6, Hornstein argues against an AGREE-based system (Chomsky (2000, 2001), and subsequent work), where AGREE is construed as an operation that relates two things (probe and goal) at a distance, mediated by c-command. Hornstein notes a few large perspective shifts from early minimalist accounts (e.g. Chomsky (1995)) to more modern accounts (Chomsky (2000, 2001)), and claims that the older system is a better choice.

Let me illustrate one specific “complaint” he has toward a system with AGREE. Hornstein notes that there is a redundancy between AGREE and the modern conception of Move. This is because not only in the case of AGREE, but also with Move, one of the core functions is to relate two things at a distance. In particular, if Move is virtually a conceptually necessary operation (as it is with Internal Merge, which is an instance of Merge), it is conceptually odd, as Hornstein claims, to have an extra operation AGREE, which does the same job. In early minimalist accounts, Move (either overt or covert) is the only operation that has the privilege to establish long-distance dependencies. Furthermore, Move is necessary and motivated in order to establish local agreement relations. Recall that the proposed system restricts feature checking only when it is concatenated with a relevant head. In modern minimalist accounts, things get reversed; AGREE is necessary in order for a phrase to undergo movement.

Directly related to the system he proposed, Hornstein also notes some incompatibilities with an AGREE-based system. First, the operation AGREE crucially employs c-command, since a probe must c-command a goal to establish a relevant relationship. If the discussion of c-command in Chapter 2 is correct, c-command should not be a primitive in the grammar. Second, although Chomsky (2008) suggests c-command is a mark of computational efficiency, Hornstein notes that while it is certain that c-command restricts the search space, the benefit in terms of the computational efficiency is not robust enough to keep the notion of c-command as a primitive in the grammar. He argues that even if the range (or maybe more concretely, ‘depth’) of search is further restricted within certain bounds (e.g., the search

---

6 In the chapter, Hornstein also contrasts AGREE with another operation “Agree,” which operates on a local dependency between elements (e.g. Spec-head agreement, head-complement agreement). Note that Hornstein argues against only the non-local version of agreement operations. Space limitation does not allow us to discuss fine details regarding those two operations. See Lasnik (2002) for some relevant discussion.
of a goal continues only within a predetermined domain, such as a phase), it is still computationally unbounded. Third, an AGREE-based system is incompatible with sideward movement. In sideward movement, the landing position does not c-command the launching site; it is in unconnected sub-trees. Finally, when Move is construed as Internal Merge, it is unclear why c-command is required only for Internal Merge, but not for External Merge. If both are an instance of Merge, the core requirement must be respected without any partition.

Hornstein reviews Existential Constructions since they exhibit agreement without apparent movement and the analyses of these constructions constitute a strong argument for an AGREE-based account. A standard example is shown in (14).

(14) There are certain to be mice/*a mouse in the tub.
In an AGREE-based account, the finite $T^0$ probes the goal / the associate (mice), checking relevant features. Now, there are several arguments for (and also some problems with) an account with Move (i.e. without AGREE), under which some kind of movement is assumed to establish a relation between an expletive and the associate. Although Hornstein goes over each one of these in some detail, let me just mention one of his arguments for and two against a movement analysis. One argument for a movement analysis is as follows. The dependency between an expletive and the associate is A-chain like; the dependency is illicit when an overt A-movement is also prohibited.

(15) a. *There seems that someone is in the room (p. 135)
   b. *Someone seems that $t$ is in the room
One problem for a movement analysis relates to scope; Den Dikken (1995) shows that the associate that is under negation in the surface position cannot take a wider scope over negation. Next, when the associate does not overtly move to the surface subject position, a kind of defective agreement pattern can show up. This is not an option when the relevant NP appears in the subject position without an expletive.

(16) a. There is a dog and a cat on the roof (p. 137)
   b. *A dog and a cat is on the roof
Hornstein argues that all of the cases above can be handled properly if Existential Constructions are treated as “an instance of doubling along the lines of Sportiche (1988)” (p. 139). The basic idea is that an expletive and the associate are generated as a pair, and the expletive there undergoes an overt movement to check its features against $T^0$. This analysis can account for the A-chain-like property of Existential Constructions since the movement of
there is A-movement. Also, the scope fact follows naturally; the associate does not take a wide scope since it does not move from the surface position. Finally, the defective agreement pattern is expected since T⁰ agrees with the expletive; the associate has a relation with T⁰ only indirectly.

7. Summary

In summary, Hornstein suggests one potentially fruitful and exciting direction that the current minimalist might pursue. Trying to minimize “overly linguistic” parts of grammar, such as c-command, we will have a better chance to explain how the Faculty of Language arises in the human mind. Some crucial mechanisms for language, such as Concatenate, are shared with other cognitive systems, yet it is fascinating that a small, but important invention, such as Label, permits the system to have very interesting properties that are unique to humans.

REFERENCES


[received December 20 2009, revised and accepted April 15 2010]

Faculty of Science and Engineering
Kinki University
3–4–1 Kowake, Higasi-Osaka-shi
Osaka 577–8502
e-mail: onohajime@kindai.ac.jp