A Graph-Theoretic Reanalysis of Bare Phrase Structure Theory and its Implications on Parametric Variation

Miyoko Yasui

Dokkyo University, Soka

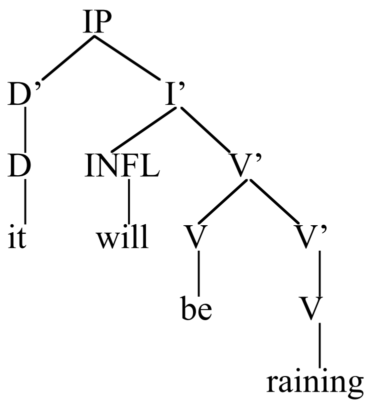
Abstract

This paper attempts to offer a minimalistic theory of syntactic structure from a graph-theoretic point of view with special emphases on elimination of category and projection labels and the Inclusiveness Condition. A syntactic structure is regarded as a graph consisting of the set of lexical items, the set of relations among them and nothing more. Operations of internal and external MERGE are redefined as applying on graphs. It is claimed that head-initial and head-final languages share basic syntactic graphs and operations and that differences in their phonetic realizations such as the positions of heads and wh-phrases are deduced from the natural extension of traversals of graphs, which are extensively studied in graph theories. A major consequence of the theory is its explanation for the correlations between the head-parameter values, on the one hand, and the types of wh-movement and word order freedom, on the other: head-initial languages such as English exhibit overt wh-movement and position their specifiers in a fixed position due to their downward traversal, while head-final languages like Japanese are subject to the upward traversal and hence do not raise wh-phrases obligatorily and their word orders are relatively free.

1. Graph-theoretic Properties of Standard Phrase Structure Diagrams

A representation of phrase structure such as (1) is a kind of directed graph:[1]

(1)



A directed graph G consists of a finite set of nodes (V) and a finite set of ordered pairs of nodes (E) that express immediate domination relations. (1) is nothing more than G= (V, E) defined in (2):

(2) V={IP, D’, I’, D, INFL, V’, V, V’, V, it, will, be, raining}

E={<IP, D’>, <IP, I’>, <D’, D>, <I’, INFL>, <I’, V’>, <V’, V>,

<V’, V’>, <V’, V>, <D, it>, <INFL, will>, <V, be>, <V, raining>}

Besides the defining properties as a directed graph, a phrase structure diagram has been assumed to have properties such as (3):

(3) (i) There is one node, called the root, that is dominated by no nodes and from which there is a path to every node.[2]

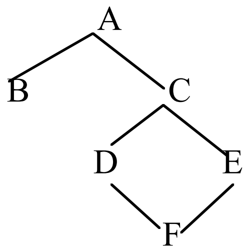
(ii) Every node other than the root has exactly one node that immediately dominates it.

(iii) The nodes each node immediately dominates are ordered from the left.[3]

(3i) says that a well-formed sentence needs to constitute a single connected graph with one special node as its root. (3iii) need not or should not be retained within the minimalist program, where linear order is assumed to play no significant syntactic role.

Like (3i), (3ii) has been adopted in virtually every theory of phrase structure, and it specifically excludes a diagram with a closed route such as (4):[4]

(4)



The offending node is F, which is dominated by two nodes, D and E; thus, (4) violates (3ii).

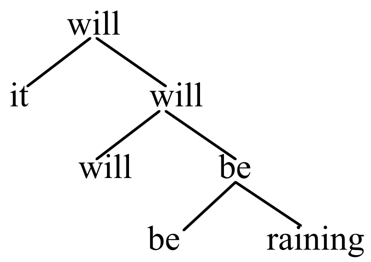
Whether (3ii) should be assumed or not depends on other assumptions on phrase structure. I will show that a graph-theoretic reanalysis of some of the fundamental assumptions in the minimalist program, specifically of bare phrase structure theory, leads to the rejection of (3ii). In the alternative theory to be proposed below, the output of external MERGE is a tree with the property (3ii) but that of internal MERGE (or movement) is a graph that has a closed route. This distinction will offer a natural explanation for the parametric difference in wh-movement; the PF requirement of linearizing lexical items forces a graph with a closed route to be changed into a tree in either of the two possible ways, which correspond to overt and covert movement.

The organization of the rest of the paper is as follows: In section 2, major assumptions in bare phrase structure theory will be examined from a viewpoint of graph theory, with special attention to the Inclusiveness Condition. In section 3, external and internal Merge will be defined according to the alternative phrase structure theory. Specifically, internal MERGE will be regarded as an operation forming a closed route; there is no necessary reason to exclude such a graph within syntax. It will be argued in section 4, however, that the PF requirement as to word order leads to the elimination of a closed route without altering the overall structure; two options are available, corresponding to overt wh-movement in head-initial languages and covert wh-movement in head-final languages. It will also be discussed how the nodes of a syntactic graph are traversed systematically and linearly ordered, from which differences in word order between these languages will be deduced. Section 6 is a brief discussion on some of the remaining issues.

2. A Graph-Theoretic Reanalysis of Bare Phrase Structure Theory

One important simplification of phrase structure pursued since Chomsky (1995: Chapter 4) is the elimination of category and projection labels by the extensive use of lexical items themselves, which is motivated by the Inclusiveness Condition. To meet the Inclusiveness Condition, (5) is to be assumed instead of (1) (=(2)):[5]

(5)

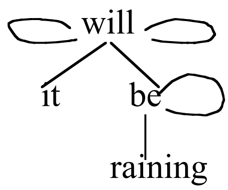


V={it, will, will, will, be, be, raining}

E={<will, it>, <will, will>, <will, will>, <will, be>, <be, be>, <be, raining>}

(5) contains nodes with the same labels *will* and *be*. If nodes with the same

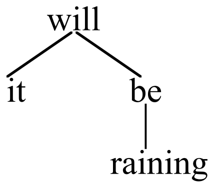
label are to be identified as one, the set V in (5) is non-distinct from {it, will, be, raining}.[6] Then, (5) is forced to be replaced by (6):

(6) 

V={it, will, be, raining}

E={<will, it>, <will, will>, <will, will>, <will, be>, <be, be>, <be, raining>}

(6) contains three loops, <will, will>, <will, will> and <be, be>, which express nothing other than intermediate projections. Therefore, (7) is to be finally assumed here:

(7) 

V={it, will, be, raining}

E={<will, it>, <will, be>, <be, raining>}

(7) might not look like a syntactic tree, but the set of nodes V is essentially the numeration in the sense of Chomsky (1995: 225-227), and E seems to best meet the Inclusiveness Condition in that no projection and category labels are added. Moreover, the order pair <α, β> is generally defined as {{α}, {α, β}}, and it looks quite close to Chomsky's (1995:244-245) definition of the object formed from α and β of the type α: {α, {α, β}}. For expository convenience, Chomsky continues to employ graphical representation such as (5), acknowledging that they are more complex than are absolutely necessary. If {{α}, {α, β}} is adopted instead of {α,{α, β}} as the definition of the object formed from α and β , the discrepancy between the formal definition and its graphical representation will disappear, which seems to be a desired result.

3. External and Internal Merge

How can the operations of internal and external MERGE be defined here? First, let's consider external MERGE, which is applied to two substructures α and β and produces a larger structure only if some syntactic relation holds between α and β. To paraphrase it in the alternative theory here, MERGE is a kind of union operation on two graphs, with a new ordered pair added to the union. It is formally defined as (8):

(8) Given two graphs G1 =(V1, E1) and G2=(V2, E2), MERGE (G1, G2) produces a graph G=(V, E) such that

(i) $V=V1 \cup V2$ and

(ii) for some $v1 \in V1$, $v2 \in V2$, $E=E1 \cup E2 \cup \{\langle v1, v2\rangle\}$ or

$E=E1 \cup E2 \cup \{\langle v2, v1\rangle\}$

(8i) is straightforward; the resulting set of nodes is simply the union of the node sets of the two input graphs. (8ii) essentially says that some node in one graph enters into a local syntactic relation with some node in the other graph, whereby the two graphs are combined. I assume the first member of each ordered pair to select the second member or to agree with it with its EPP feature.

A derivation starts with a set of minimal graphs, each of which consists of a single lexical item. (5), for example, starts with (9):[7]

(9) ・it ・will ・be ・raining

G1=(V1, V2): $V1=\{it\}$, $E1=\phi$

G2=(V2, E2): $V2=\{will\}$, $E2=\phi$

G3=(V3, E3): $V3=\{be\}$, $E3=\phi$

G4=(V4, E4): $V4=\{raining\}$, $E4=\phi$

Since the aspectual auxiliary *be* selects the progressive verb *raining*, the ordered pair <be, raining> is introduced as in (10):

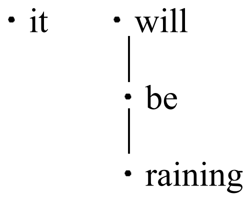
(10) ・it ・will ・be MERGE(G3, G4) = G5=(V5, E5)

|

・raining $V5=V3 \cup V4=\{be, raining\}$

$E5=E3 \cup E4 \cup \{\langle be, raining\rangle\}=\{\langle be, raining\rangle\}$

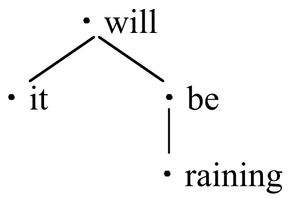
Will selects verbal and Case-checks nominative; thus, the recursive application of MERGE will convert (10) into (11) and then (12) (=(7)):

(11) 

MERGE (G2, G5)=G6=(V6, E6)

$V6=V2 \cup V5=\{will, be, raining\}$

$E6=E2 \cup E5 \cup \{\langle will, be\rangle\}=\{\langle will, be\rangle, \langle be, raining\rangle\}$

(12) 

MERGE (G1, G6)=G7=(V7, E7)

$V7=V1 \cup V6=\{it, will, be, raining\}$

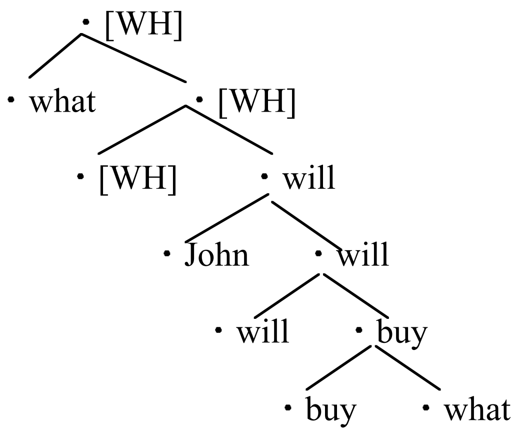
$E7=E1 \cup E6 \cup \{\langle will, it\rangle\}=\{\langle will, it\rangle, \langle will, be\rangle, \langle be, raining\rangle\}$

It does not matter which of the ordered pairs in E7 is added first. For example, it is easy to verify that adding the pair <will, it> before <will, be> or <be, raining> will make the same result.[8]

MERGE defined in (8) does not prevent a graph from being combined with itself, and it is exactly a case of internal MERGE (or movement). To illustrate this point, consider (13), a case involving wh-movement:

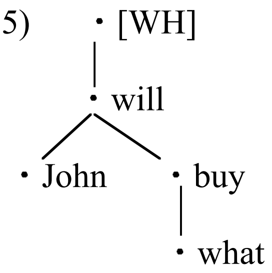
(13) (I wonder) what John will buy.

The bare phrase structure analysis of (13) is (14), where the *v*P structure and the movement of the subject/object are ignored for expository convenience here:[9]

(14) 

In the theory here, (13) has structure (15) before the movement of *what*:

(15)



V={[WH], will, John, buy, what}

E={<buy, what>,<will, buy>,<will, John>,<[WH], will>}

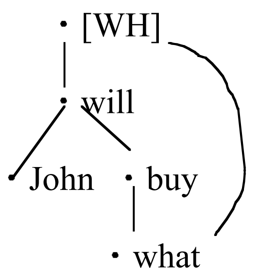
MERGE (G, G) will produce (16), where the wh-checking relation of <[WH], what> is added:

(16) MERGE (G, G)=G'=(V', E'):

V'=V ∪ V=V={[WH], will, John, buy, what}

E'=E ∪ E ∪ {<[WH], what>}={<buy, what>,<will, buy>,

<will, John>, <[WH], will>, <[WH], what>}



(16) violates one of the defining properties of tree, (3ii): *what* is immediately dominated by *buy* and [WH]. More generally, internal MERGE on a tree always introduces one closed route, and the result is not a tree by definition (see Balakrishnan and Ranganathan (2000; 71-72)). (16) might look too outrageous, given the widely accepted view that a sentence has a tree structure. Nothing, however, seems to be wrong with (16) as a syntactic structure. I will move on to discuss the PF interpretation of syntactic graphs in the next section, where a

graph with a closed route such as (16) is to be mapped into two linear orderings and the so-called overt/covert distinction in wh-movement is subsumed under these options.

4. PF-interpretation of Graphs and Parametric Differences

4.1 Head-Parameter

Let us examine what conditions have to be imposed on the PF interpretation of a syntactic graph. One obvious requirement is that all the nodes corresponding to overt elements in a syntactic graph be pronounced.[10] Furthermore, if the interpretation is economical, each node is to be pronounced just once.[11] This kind of task is known as traversal. To traverse a graph is to perform a given operation on every node (e.g., to pronounce it) in the graph exactly once. A node can be traversed more than once but may undergo the operation exactly once.

A specific type of graph is associated with common orders in which its nodes undergo a given operation. If a directed graph is linear, the two common orders are forward and backward in its direction. Consider the graphical representations of the phrase "lean against the door to the backyard" and its Japanese counterpart, which are to be analyzed as (17a,b), respectively:

(17) a

• lean
↓
• against
↓
• the
↓
• door
↓
• to
↓
• the
↓
• backyard

b.

• motare (lean)
↓
• ni (against)
↓
• doa (door)
↓
• e-no (to-GEN)
↓
• uraniwa (backyard)

Each node in (17a) is pronounced before its child; you pronounce the verbal head *lean* first, pronouncing the other nodes in the downward direction. The opposite holds in (17b); the pronunciation starts at the nominal complement *uraniwa* (backyard) and proceeds in the upward direction. The downward and upward pronunciation options constitute the head-parameter, yielding the head-initial and head-final word orders, respectively.

A syntactic graph typically is a tree and its traversal is not so simple as that of a linear graph. Traversal of an ordered binary tree is one of the important issues in graph theories.[12] In its commonly-used traversals, the root is the starting and ending node, and they involve the following three parts:

(18) At a given node N:

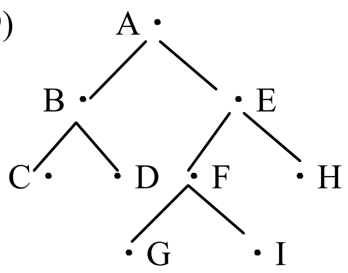
a. Recursively traverse its left subtree. When this step is finished, go back to N.

b. Recursively traverse its right subtree. When this step is finished, go back to N.

c. Perform the operation on N.

(18a,b) say that the left subtree of N is traversed before its right subtree. Traversals can be classified according to whether (18c) is done before, between or after (18a,b). They are called pre-order, in-order and post-order traversals, respectively. Let us assume that the operation mentioned in (18c) is to pronounce N. The pre-order traversal of (19) results in the pronunciation A-B-C-D-E-F-G-I-H:

(19)



The in-order and post-order traversals yield C-B-D-A-G-F-I-E-H and C-D-B-G-I-F-H-E-A, respectively.[13]

(18a,b) make crucial reference to the order of subtrees. Let us illustrate the point by traversing (19) in pre-order. You pronounce the nodes A, B and C in this order since B is the root of A's left subtree and C is the root of B's left subtree. Then, you come back to B and should visit D rather than go back to C. You never make a mistake at B since you know you have visited its left subtree. If subtrees are not ordered from left to right, the 'left' subtree does not exist, and we are not sure at a branching node which way to go.

As has been discussed in connection with (3iii) in Section 1, a syntactic tree has no distinction of right and left subtrees; thus, (18a-c) cannot be adopted as they are. Note that in the traversal of (19) under the assumption that it is ordered, the branching node B is traversed three times. When B is traversed for the first time, its two children are still not pronounced; one of them has been pronounced when B is traversed second time, and both of them have been pronounced at the point of B's last traversal. In this way, the number of unpronounced/pronounced nodes B immediately dominates changes each time B is traversed, and the same holds of branching nodes in general. The three types of traversal can be reformulated along this line.

Originally, each node immediately dominates no pronounced nodes.[14] If the node is pronounced at this stage, it is before any of its subtrees has been traversed; this is a pre-order traversal. If it is pronounced when it immediately dominates no unpronounced nodes, it is a post-order traversal. An in-order traversal is slightly more complicated; each node is pronounced at the stage where it immediately dominates no more than one unpronounced node. Pendant nodes such as C,D and G in (19) immediately dominate no nodes and they are pronounced when they are first traversed. Branching nodes like B are pronounced after one of its children has been pronounced. In this way, we can identify three kinds of traversal based of the number of unpronounced or pronounced nodes dominated by the node to be pronounced: no pronounced nodes (pre-order), no more than one unpronounced node (in-order), and no unpronounced nodes (post-order).

The upward pronunciation of the linear graph (17b) falls under the version of post-order traversal sketched above. Starting from *motare* (lean), you move to the pendant node *uraniwa* (backyard) without pronouncing the nodes en route; the latter immediately dominate an unpronounced node at this state. You pronounce *uraniwa*, which immediately dominates no (unpronounced) nodes, go back to *e*-(*no*) (to-GEN) and pronounce it since its only child has been pronounced. *Doa* (door), *ni* (against), and *motare* (lean) are pronounced in this order.

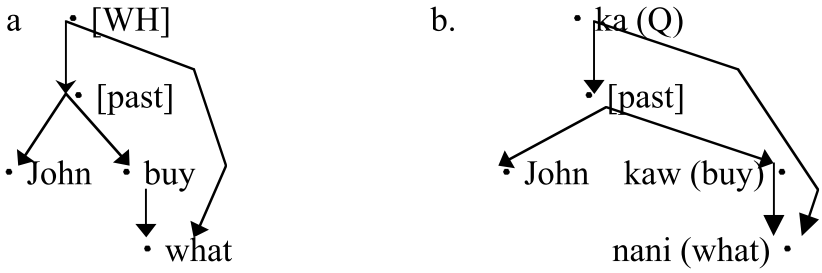
The downward pronunciation of the linear graph in (17a) can be subsumed under the pre-order for an obvious reason, but it also can be captured by the in-order traversal formulated here; the verbal head *lean* is pronounced first since it immediately dominates just one unpronounced node at the initial stage, its child node *against* is pronounced next for the same reason, and the pendant node *backyard*, which immediately dominates no unpronounced node, is pronounced after all its higher nodes have been pronounced.

Obviously, the English word order of specifier-head-complement is not obtainable in the pre-order. If the possibility of pre-order is excluded for some reason in natural languages, the head-parameter applying to binary syntactic trees is the option of in-order versus post-order. Applying the in-order traversal method to appropriate linear subparts of binary trees results in the downward or head-initial pronunciation pattern; nodes nearer to the root are pronounced earlier. The head-final pattern is deducible from the post-order traversal: nodes more distant from the root are pronounced earlier. The head-parameter will be discussed further in Section 4.3.

4.2 Overt *Wh*-movement and Wh in-situ

If a node is directly dominated by a single node as in (19), these nodes are simply to be pronounced in the neighborhood; one of them is pronounced immediately before or after the other according to a head-parameter value. This simple ordering is not applicable to a result of internal MERGE such as (20a) and its Japanese counterpart in (20b):

(20) a



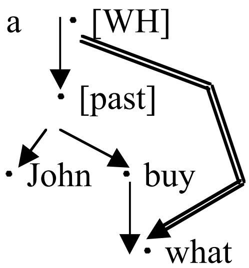
b.

What/nani are immediately dominated by two nodes, *buy/kaw* and [WH]/*ka*, which would mean that *what/nani* should be pronounced in the positions adjacent to *buy/kaw* and to [WH]/*ka*. Obviously, pronouncing *what/nani* in two distinct positions at the same time is impossible due to the limitation on our sensory-motor system. Pronouncing *what/nani* twice is possible but is not economical. In brief, it seems necessary to remove a closed route in (20a,b) and convert them into trees.

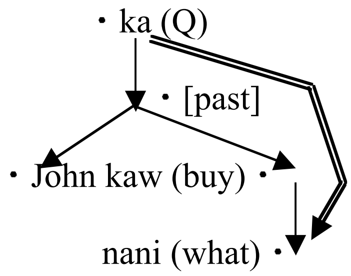
Given a connected graph with one closed route, deleting any one branch in the route will change it into a tree (see Balakrishnan, R. and K. Ranganathan (2000:72)). This operation is minimal in that it removes just one branch without changing the node set. An additional restriction is necessary in choosing a branch to be deleted: the overall structure should not be changed too radically. For instance, deleting the branch <[WH], [past]> in (20a) can eliminate the closed route, but [WH] will cease to dominate [past], *John* and *buy*, which means that it is no longer the root by definition. To maintain [WH] as the root, we should remove one of the incoming branches of the doubly-connected node: <buy, what> or <[WH], what>.

Note that the branch <[WH], what> constitutes the shortest path from [WH] to *what*, while <buy, what> is part of the longer path connecting them. Similarly, in (20b), <ka, nani> forms the shortest path between *ka* and *nani* and <kaw, nani> is part of the longer path connecting them. I assume that there can be constructed a simple algorithm to find out the shortest path from each doubly-connected node to the branching root and to mark it, for instance, as follows:

(21) a

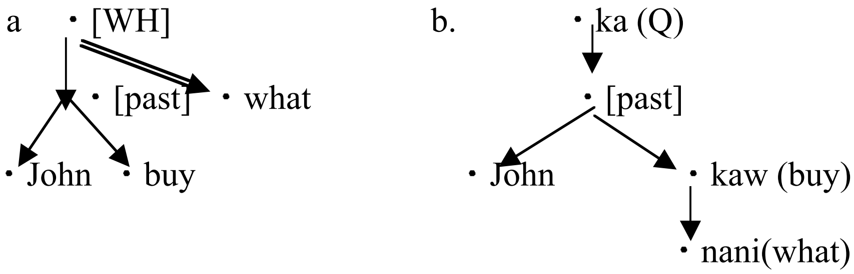


b.



In the downward pronunciation, nodes nearer to the root are to be pronounced earlier. Then, it is natural to pick up the shorter path from *what* to the root in (20a). On the other hand, (21b) is subject to the upward pronunciation, according to which nodes more distant from the root are pronounced earlier; the longer path from *nani* is to be chosen. Then, (21a,b) are to be modified as (21a,b), respectively:

(22) a b.



In (22a), we go downward from [WH] directly to *what* rather than through [past] and *buy.* This means that *what* is pronounced immediately after [WH] rather than *buy.* In (22b), in contrast, we backtrack from the object *nani* and go upward to *ka* (Q) through the verb *kaw* and [past]; *nani* is pronounced immediately before the verb selecting it rather than before *ka* (Q). In brief, a *wh*-phrase is pronounced in the neighborhood of the [WH]-COMP attracting it in head-initial languages such as English but it is pronounced in its original position in head-final languages such as Japanese.

We have just deduced an important correlation between word order and *wh*-movement. This generalization has been pointed out by Bach (1970) and Bresnan (1972). In particular, Bresnan (1972: 42) observes that only languages with clause-initial COMP permit a COMP-attraction transformation. Logically possible but non-existent are languages with a clause-final COMP that attracts a *wh*-phrase rightward. Ikawa (1996: Chapter 4) extends this correlation to NP-movements: English and a number of other head-initial languages exhibit obligatory overt leftward movements, but no comparable rightward movements are attested in head-final languages. The theory advocated here offers one natural account of the leftwardedness of overt movement.

To recapitulate the discussion so far, a doubly-connected node in a graph causes a PF problem and one of its incoming branches needs to be ignored during the PF interpretation of the graph. When the graph is minimally modified, the head-parameter value is respected: the shorter path is chosen in a head-initial language, where a node nearer to the root is pronounced earlier, and the opposite holds in a head-final language. The idea behind this is Fukui's (1993:400) Parameter Value Preservation (PVP) measure cited in (23):

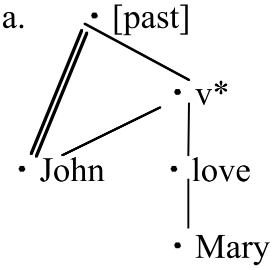
(23) A grammatical operation (Move α, in particular) that creates a structure that is inconsistent with the value of a given parameter in a language is

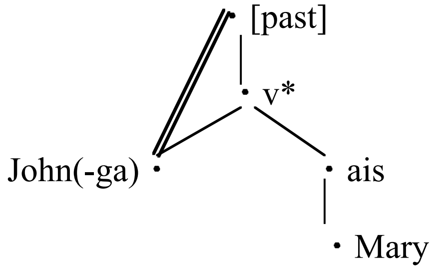
> costly in the language, whereas one that produces a structure consistent with the parameter value is costless.

Using (23), Fukui explains scrambling phenomena in Japanese; Japanese is head-final, and moving a constituent to the left does not alter the head-finality, hence scrambling is costless. If a grammatical operation in (23) covers deletion of a branch exemplified in (22a,b), that operation is costless.[15]

4.3 IP-Specifiers

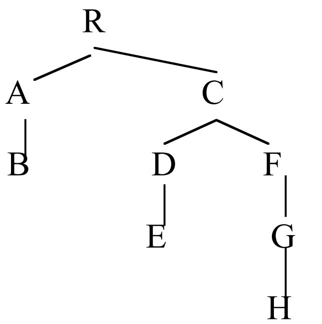
It is illustrated by (21) and (22) in the previous section that the shortest path from a wh-specifier to a [WH]-COMP attracting it is retained in English but is removed in Japanese on the PF side. The same is true of IP specifiers:[16]

(24) a.


b.


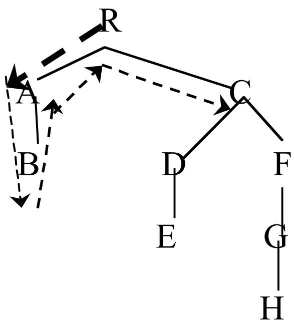
<[past], John> constitutes the shortest path from *John* to the root in both cases, and it is retained in (24a) but is removed in (24b); the nearness to the root has significance in the former. Japanese has a syntactic specifier just like English but, in some sense, loses it in the PF interpretive processes. Then, it is natural that a specifier occupies a special position only in the word order of English.

More generally speaking, the root of (25) is branching and it produces path ambiguities:

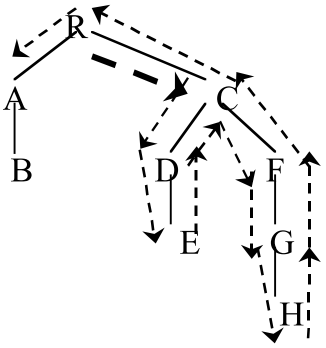
(25)


Choosing either A or C at R yields distinct subtraversals such as (25a, b), respectively, where the initial step is highlighted by a heavy dotted line:

(26) a



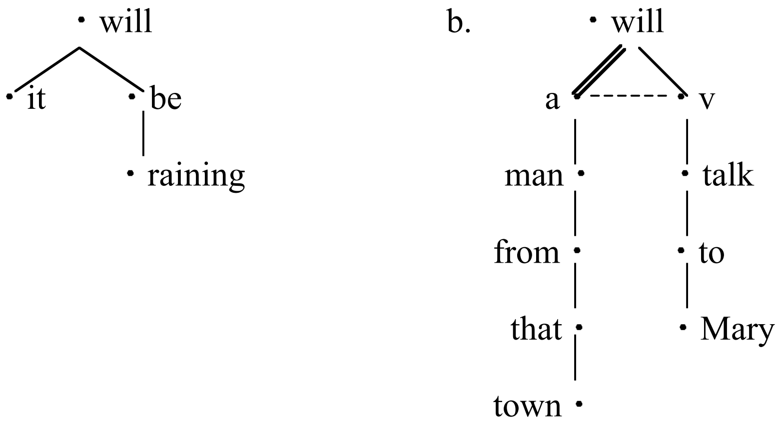
b



If the subgraph rooted by A is the specifier of R and the one rooted by C is its complement, applying the in-order/downward pronunciation to the subtraversal in (26a) will give the correct order in English: Specifier-Head-Complement. The subtraversal in (26b) will yield the order Head-Complement-Specifier. If Specifier-Head-Complement is the only order (of head-initial languages) as claimed by Kayne (1994), a specifier needs to be distinguished from a complement, which is possible thanks to its nearness to the head. Specifically, the node A has initially been connected to R and also to some other node; at the (post-)syntactic level, the incoming branch from R has been marked by the algorithm mentioned in Section 4.2 and the other branch has been removed. If the specially marked branch is given priority at any branching node, the order of Specifier-Head-Complement can be properly obtained for English.

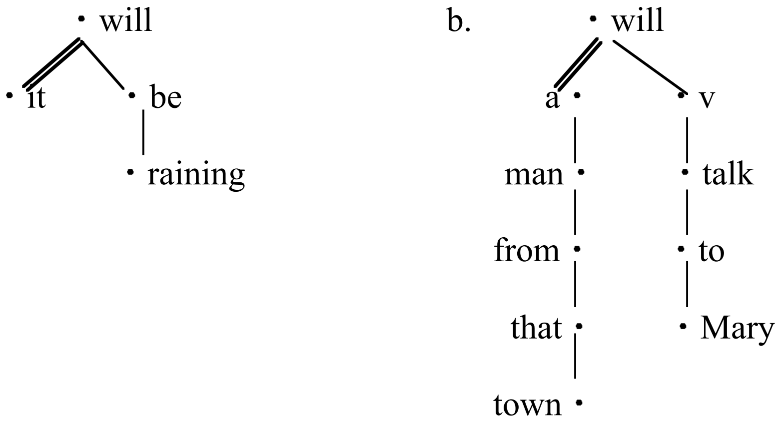
A specifier created by movement or internal MERGE is directly connected with the head triggering that operation. An expletive like *it* is also directly connected with a finite tense by external MERGE. Its incoming branch, however, is not specially marked since only a doubly-connected specifier created by internal MERGE has been assumed to be subject to the algorithm at stake. Still, an expletive stands out naturally in always being the nearest pendant node to the root. Compare the expletive in (27a) with the doubly-connected IP-specifier in (27b):

(27) a.



The expletive *it* constitutes a minimal maximal projection and it is the nearest pendant node to the root.[17] *The man from New York* in (27b) is a phrasal specifier and its head *a* is connected to the root directly and indirectly through the light verb. Note that the pendant node *town* is more distant from the root than *Mary.* I will assume that the algorithm mentioned above can be extended to mark the incoming branch from the root to an expletive on a par with the shortest path to a doubly-connected node; the former itself is the nearest pendant node to the root, and the latter retains its shortest path to the root:

(28) a.



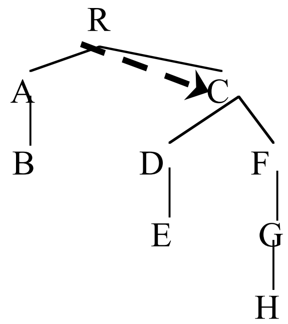
Then, the expletive subject in (28a) can be properly pronounced before the other nodes just like the phrasal subject in (28b) is. Japanese lacks expletives presumably because it has syntactic specifiers but no phonetic specifiers.

4.4 Word Order Variation

Head-final languages are subject to the post-order/upward pronunciation.

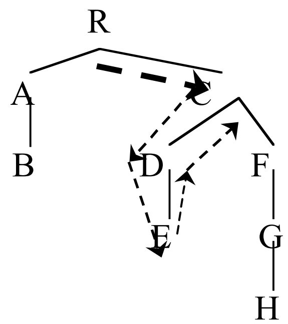
In (25) repeated below as (29), let us start at R and visit C as follows:

(29)

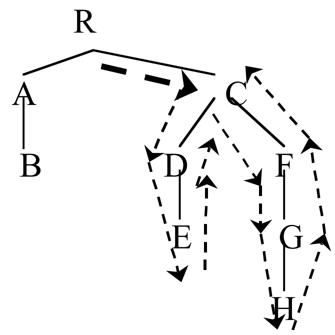


Traversing the subtree rooted by D and coming back to C will yield (30):

(30)

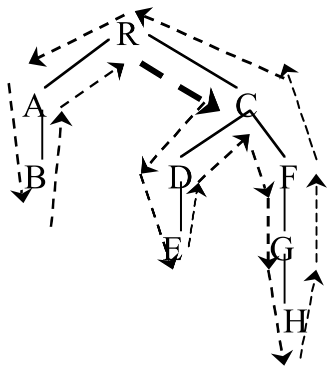


(31)



If you visit the other subtree rooted by C as in (31), all the nodes dominated by C have been visited. Visiting the subtree rooted by A and coming back to the root results in (32):

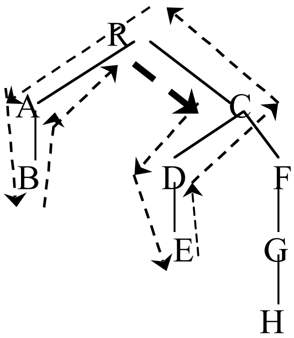
(32)



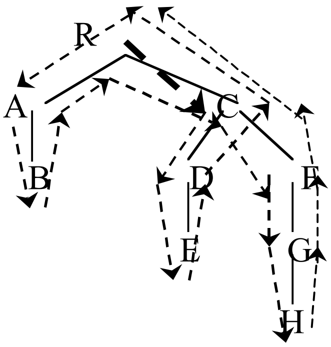
Each branch is traversed once downward and once upward in (32). In the upward pronunciation, each node is pronounced at the point where it immediately dominates no unpronounced nodes; namely it is pronounced in its backtracking traversal. (32) renders the pronunciation E-D-H-G-F-C-B-A-R.

Going back to (30), let us visit R, A, and B before visiting the remaining nodes within the subgraph rooted by C as in (33a):

(33) a.



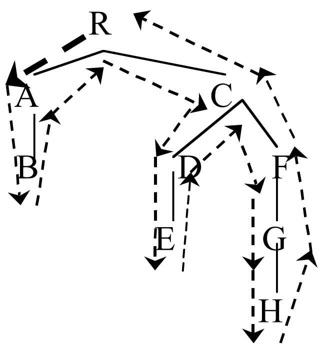
b



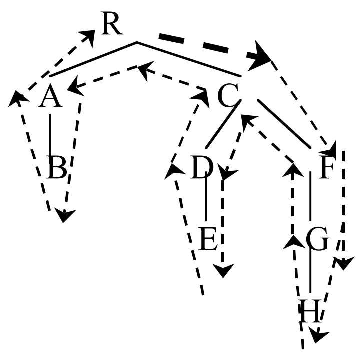
You will be back to R after traversing the remaining unpronounced subtree rooted by C, as in (33b); the sequence of traversed nodes is R->C->D->E->D->C->R->A->B->A->R->C->F->G->H->G->F->C->R and the pronunciation is E-D-B-A-H-G-F-C-R. Unlike in (32), the branch <R, C> is unnecessarily traversed four times. Since a traversal has been assumed to start and end at the root, the natural economy condition is that each branch should be traversed downward once and backtracked once, which is the case in (32) but is not in (33b).

Beside (32), there are other traversals of (29) that satisfy the economy condition just mentioned. (29) has two other pendant nodes, B and H. Visiting one of these nodes before the other pendant nodes yields three distinct sequences of visited nodes: (i) R->A->B->A->R->C->D->E->D->C->F->G->H->G->F->C->R; (ii) R->A->B->A->R->C->F->G->H->G->F->C->D->E->D->C->R; and (iii) R->C->F->G->H->G->F->C->D->E->D->C->R-> A->B->A-> R. (i) and (iii) are given below:

(34) a.



b.



The post-order/upward pronunciations of traversals in (34a,b) result in B-A-E-D-H-G-F-C-R and H-G-F-E-D-C-B-A-R, respectively. Choosing F at C before D in (34a) produces (ii) and it is phonetically realized as B-A-H-G-F-E-D-C-R . Each branch is traversed downward once and upward once in all the three traversals.

I assume that any of the three orderings obtained above are basic word orders. It follows that a head-final language, which takes the post-order/upward option in the linearization of its syntactic graphs, allows its word order to be quite free. As is well-known, Japanese confirms this prediction. Besides (35a), for example, (35b-d) are equally acceptable:

(35) a. [John-ga [Mary-ga hon-o katta to] itta]

NOM NOM book-ACC bought COMP said

'John said that Mary had bought the book.'

b. [John-ga [hon-o Mary-ga katta to] itta]

Lit. 'John said that the book, Mary had bought.'

c. [[Mary-ga hon-o katta to] John-ga itta]

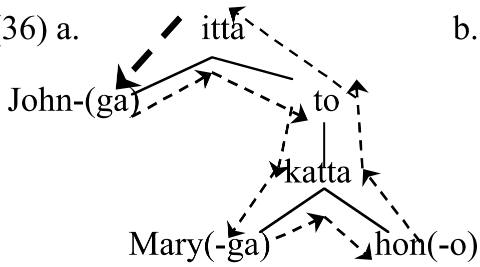
'That Mary had bought the book, John said.'

d. [[hon-o Mary-ga katta to] John-ga itta]

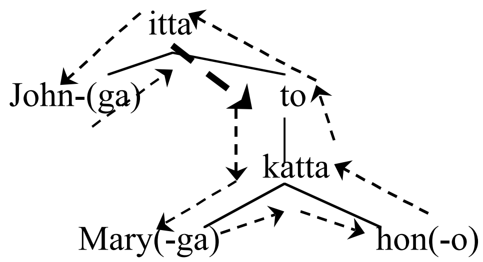
Lit. That the book, Mary had bought, John said.'

According to the standard analysis, (35a) has the basic word order; clause-internal scrambling of the embedded object results in (35b) and scrambling of the embedded clause in front of the matrix subject in (35a,b) produces (35c,d), respectively. (35a-d) are all obtained from the same syntactic graph here without running counter to the economy condition on traversal. (35a,c), for instance, are based on the traversals described in (35a,b), respectively:[18]

(36) a.



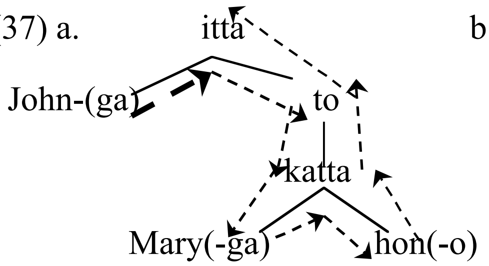
b.



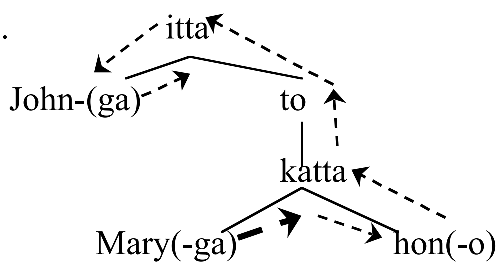
The traversal in (36a) underlies (35a), which is generally regarded as the basic word order. (35c), which is deduced from the traversal in (36b), is also treated as basic here. This accords with the claim by Hale (1980) among others.

Actually, clause-internal scrambling of non-subjects exemplified in (35b,c) does not require any heavy stress on them or special discourses. What is more, most native speakers of Japanese prefer (35c) to (35a) since the latter involves a center-embedding of the complement clause. If the traversal starts with the first-pronounced pendant node rather than the root, which does not affect the resultant pronunciation, the first few steps from the root to the pendant in (36a,b) are removed as follows:

(37) a.



b.



The total number of branched traversed is 9 in (37a) and 7 in (37b); the word order of (35c) is obtainable with fewer steps than that of (35a). This is likely to make (35c) easier to process than (35a). The same contrast holds between the traversals underlying (35b,d). I will assume that sentences with clause-internal scrambling are all basic word orders and that those with fewer branches traversed tend to be preferred at the PF-interface level.[19]

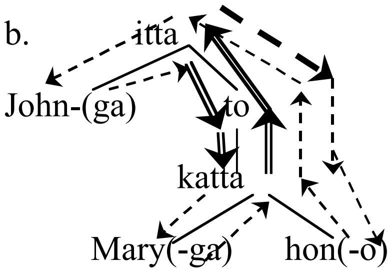
(38a) involves long-distance scrambling of the embedded object in (35a) and is deduced from the traversal described in (38b):

(38) a. [hon-o John-ga [Mary-ga katta to] itta].

book-ACC NOM NOM bought COMP said

'The book, John said that Mary had bought.'

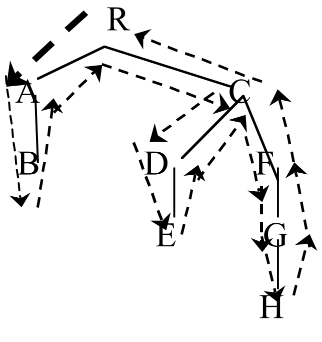
b.



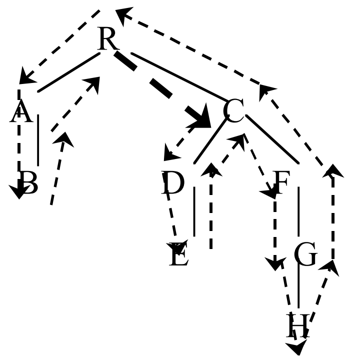
You first traverse the embedded object and backtrack to the embedded verb *katta* (bought). Next, instead of visiting the embedded subject, you go upward to the matrix verb *itta* (said) and visit the matrix subject. Then, you backtrack to the embedded verb to visit the embedded subject and finally go back to the root. (38b) violates the economy condition of traversal discussed above, with its superfluous subtraversals expressed by the double arrows. This accords with the generally accepted view that cases involving long-distance scrambling such as (38a) do not have basic word order, and specifically with the observation by Saito (1992) and others that long-distance scrambling behaves differently from clause-internal scrambling. We can conclude either that a violation of the economy condition of traversal is allowed by some discourse-level factors such as focus or that (38a) is deduced from a syntactic graph distinct from the one underlying (35a-d).

Let us go back to the instances of the in-order/downward pronunciation in English. The two traversals described in (26a,b) repeated below as (39a,b) are consistent with the economy condition:

(39) a.



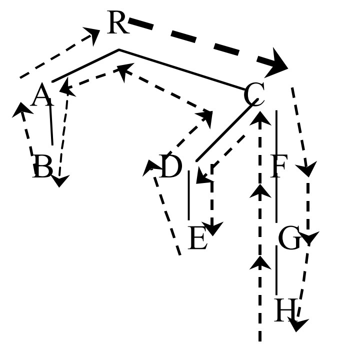
b



No branches in (39a,b) are traversed three times. There is one more downward traversal consistent with the condition, which has the sequence of nodes

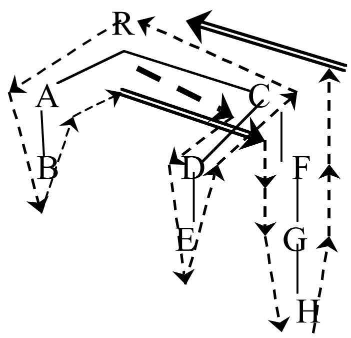
R->C->F->G->H->G->F->C->D->E->C->R->A->B and the resultant ordering F-G-H-C-D-E-R-A-B:

(40)



(39b) and (40) are filtered out by the specifier-first condition discussed in Section 4.3. A violation of the economy condition on traversal in the in-order/downward pronunciation is (41), with the sequences of traversed nodes R->C->D->E->D->C->R->A->B->A->R->C->F->G->H->G->F->C->R and the resultant pronunciation D-E-C-R-A-B-F-G-H:

(41)



After going down the left path from C and backtracking to C, the traversal goes up to R and scans its left subtree instead of traversing the right subtree of C. The branch <R, C> has been traversed four times. The economy condition blocks the traversal in (41). What is interesting is that the specifier-first condition imposed on the in-order/downward pronunciation is stricter, filtering out (39b) and (40) as well as (41). In this way, English does not exhibit the word order freedom Japanese does.[20]

5. Remaining Issues

This paper has been mainly concerned with parametric differences in word order: phonetic realizations of specifiers such as attracted wh-phrases, argumental subjects and expletives if any, and the flexibility or rigidity of basic word order. One important operation that greatly affects word order but cannot be discussed here is head-movement. Presumably, it can be analyzed as what is called edge contraction in graph theories, which deletes a branch and identifies the two nodes directly connected by the deleted branch. Depending on which of the two nodes is retained, the overall graphical structure greatly differs. A analysis along this line appears to be promising. Another issue, which has not been explored here or solved in other syntactic studies, is the treatment of adjuncts. It is not clear whether they are introduced by external MERGE or some comparable operation at the edge of the graph already formed, or they are inserted into the graph post-cyclically. I leave these problems for future research.

Notes

---

[1] Downward lines are used instead of arrows, as has been the convention in syntactic trees.

[2] A path is any linear subset of a tree.

[3] A terminology more familiar in graph theories is: α is a predecessor of β and β is a successor of α. See, for example, Balakrishnan, and Ranganathan (2000), and Hopcroft and Ullman (1979).

[4] A closed route is called 'cycle' in the graph theory, but I will reserve the terminology to be used for 'cycle' in rule application and so on.

[5] vP structure, if any, is ignored in (5). See note 16.

[6] Chomsky (1995: 244) distinguishes nodes with the same label by indices, which is an undesirable departure from the Inclusiveness Condition.

[7] Nodes will be expressed with dots henceforth.

[8] Thus, the crash-proof selection pursued in Frampton and Gutmann (2000) does not constitute a significant issue here. See Yasui (2002).

[9] Following Lopez (2000), I will assume in Section 4 that an object is Case-checked in its original position.

[10] In the theory here, Move does not produce a copy of the moved element, so that phoneti-

cally non-empty elements in a graph can all be pronounced, which is a desirable result.

11 Relaxing this economy condition presumably results in clitic doubling and similar phenomena.

12 A binary tree is a rooted tree where each node immediately dominates at most two nodes (i.e., zero, one or two children). Each child can be identified as either a left or right child. If a node has just one child as in a linear graph, identifying it as left or right does not affect the traversal if it is pre- or post-order. See Sahni (2001) among others.

13 In any of the three traversals, the nodes are traversed in the same order according to the 'left-first' rule: A-B-C-B-D-B-A-E-F-G-F-I-F-E-H-E-A.

14 There are two cases: it immediately dominates just unpronounced nodes or no nodes at all. The latter case covers pendant nodes.

15 We can assume either that this operation takes place in syntax and its output is subject to the PF interpretive processes, or that (23) governs PF as well as syntactic operations.

16 Following Lopez (2001), I assume that the object stays in its original position and Case-checked. Given this assumption, we can maintain the generalization that each node has at most two downward branches. A tense selects a light verb and agrees with D; COMP selects T (and agrees with a wh-phrase if it is [WH]); D selects N (and agrees with a DP-specifier if it is the genitive marker). Similarly, each lexical category could be analyzed as selecting at most one category if the object of a predicate taking two arguments such as *put* is assumed to be Case-checked in-situ. This generalization collapses if Chomsky's analysis of the transitive verbal structure is taken, where v* is connected to three distinct categories: the object, the verb it selects, and the subject it selects. The choice does not seriously affect the discussion here.

17 The expletive in "it will rain" is one of the nearest pendant nodes to the root if the sentence is analyzed on a par with (28a), or it is the only nearest node if an intransitive verb is selected by a light verb.

18 The verb stem *kaw* (buy) and the past tense morpheme are assumed here to constitute a single node. More exactly, the subject is connected to the light verb selecting *kaw* (buy) by external MERGE and doubly connected to the past tense by internal MERGE. Since internal MERGE has no PF effect in upward pronunciation languages as discussed in Section 4.2, the simplified analysis in (31a,b) suffices.

19 According to Fukui's PVP, not only clause-internal scrambling but also long-distance scrambling are costless. In the theory here, the former produces equally economical PF realizations but the letter yields less economical PF realizations.

---

[20] Fukui (1993) claims that English exhibits rightward cost-free movement, which is the counterpart of Japanese scrambling. The present theory denies it.

References

Balakrishnan, R. and K. Ranganathan (2000) *A Textbook of Graph Theory.* Springer.

Bach, Emmon (1971) "Questions." *Linguistic Inquiry* 2, 153-166.

Bresnan, Joan (1972) *Theory of Complementation in English Syntax*. Doctoral dissertation, MIT.

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

_____ (1999) *Derivation by Phase: MIT Occasional Papers in Linguistics no.18*.

_____ (2000) "Minimalist Inquiries: The Framework." In *Step by Step: Essays on Minimalist Syntax in Honor of Howard Lasnik*, eds. Roger Martin, David Michaels, and JuanUriagereka. MIT Press, Cambridge.

_____ (2001) "Beyond Explanatory Adequacy." ms., MIT.

Frampton, John and Sam Gutmann (1999) "Cyclic Computation, a Computationally Efficient Minimalist Syntax." *Syntax* 2, 1-27.

_____ (2000) "Crash-Proof Syntax. " ms., Northeastern University.

Fukui, Naoki (1993) "Parameters and Optionality." *Linguistic Inquiry* 24, 399-420.

Hale, Kenneth (1980) "Remarks on Japanese Phrase Structure: Comments on the Papers on Japanese Syntax." In *Theoretical Issues in Japanese Linguistics : MIT Working Papers in Linguistics* Vol. 2, eds. Yukio Otsu and Ann Farmer. MIT.

Holte, Robert C.(2002) "Data Structures Course Lecture Notes." University of Ottawa (http://www.csi.uottawa.ca/~holte/T26/top.html)

Hopcroft, John E. and Jeffrey D. Ullman (1979) *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley.

Ikawa, Hajime (1996) Overt Movement as a Reflex of Morphology. Doctoral dissertation, University of California, Irvine.

Kayne, Richard S. (1994) *The Antisymmetry of Syntax*. The MIT Press.

Lopez, Luis (2001) "On the (Non)complementarity of θ-Theory and Checking Theory." *Linguistic Inquiry* 32, 694-716.

Sahni, Sartaj (2001) *Data Structures, Algorithms and Applications in JAVA*. McGraw-Hill

Saito, Mamoru (1992) "Long Distance Scrambling in Japanese." *Journal of East Asian Linguistics* 1, 69-118.

Yasui, Miyoko (2002) "Crash-Proofness of the Lexicon and Bare Phrase Structure." ms., Dokkyo University. (http://kasiwagi.dokkyo.ac.jp/~myasui/general/profile/publications/F&G.pdf)