Locality on Selection and Labeling

Toru Ishii
Meiji University

1. Introduction

Merge, which subsumes traditional phrase structure rules and transformations, is the fundamental structure-building operation in human language. In Chomsky's (1995) formulation (1), Merge is an asymmetric operation; it combines two elements into a set and projects one of the merged elements, providing the set with a label:

(1) Asymmetric Merge

\[
\text{Merge} (\alpha, \beta) = \{ \gamma, \{\alpha, \beta\}\}, \text{ where } \gamma \in \{\alpha, \beta\}
\]

(adapted from Chomsky 1995: 243)

In the asymmetric formulation (1), labeling is part of Merge, but this is a residue of phrase structure grammar. It is therefore better to separate the labeling part from Merge. Under this view, Chomsky (2004, 2008, 2013) formulates Merge as a symmetric operation as shown in (2). In (2), Merge combines two elements into a set, but labels are not created by Merge. Labels are rather determined by the two labeling algorithms in (3):

(2) Symmetric Merge

\[
\text{Merge} (\alpha, \beta) = \{\alpha, \beta\}
\]

(adapted from Chomsky 2004: 117)

(3) Labeling Algorithms (Chomsky 2008: 145)

a. In \{H, \alpha\}, H an LI, H is the label.

b. If \alpha is internally merged to \beta, forming \{\alpha, \beta\}, then the label of \beta is the label of \{\alpha, \beta\}.

(3a) states that it is always a head that projects. (3b) states that in Internal Merge, it is always the target that projects. The
symmetric formulation of Merge is also proposed by Boeckx (2008), Hornstein (2009), and Fukui (2011).

Under the symmetric Merge together with the labeling algorithms approach, Merge and labeling are independent operations so that we should expect that labeling may apply without Merge. This paper deals with complementizer stacking in Japanese and Korean exemplified by (4, 5), arguing that labeling without Merge applies during a derivation of complementizer stacking:

(4) John-wa Bill-ni [dare-ga kita ka to] tazuneta
    John-Top Bill-Dat who-Nom came Q that asked
    Lit. ‘John asked Bill that who came.’

(5) John-nun Mary-eykey [pro kumwuncey-lul phwul-ess
    John-Top Mary-Dat that problem-Acc solved
    nya ko] mulessta
    Q that asked
    Lit. ‘John asked Mary that whether she solved that
    problem.’

In (4) and (5), two complementizers are stacked at the right edge of the complement clause. In the Japanese example (4), the interrogative complementizer ka ‘Q’ and the declarative complementizer to ‘that’ are stacked. Similarly, nya ‘Q’ and ko ‘that’ are stacked in the Korean example (5). I argue that there are dual selections involved in (4) and (5), i.e. semantic selection between the matrix predicate tazuneta/mulessta ‘ask’ and the interrogative complementizer ka/nya ‘Q’, and syntactic selection between the matrix predicate and the declarative complementizer to/ko ‘that’. Given sisterhood condition on selection, which states that an element can only select its sister, no previous
analysis can account for why the matrix predicate tazuneta/mulessta 'ask' can semantically select the interrogative complementizer ka/nya 'Q' skipping over to/ko 'that' in (4) and (5). I propose that when a labeling conflict arises, relabeling may apply as part of LF-Transfer, which accounts for the syntactic and semantic selections in a straightforward way.

The organization of this paper is as follows. Section 2 investigates complementizer stacking in Japanese and Korean, showing that complementizer stacking involves dual selections. Section 3 presents evidence against a direct quotation analysis of complementizer stacking. Section 4 first explicates labeling conflicts and cartographic structure building, and then proposes a relabeling analysis of complementizer stacking. Section 5 makes concluding remarks.

2. Complementizer Stacking in Japanese and Korean

2.1 Complementizer Stacking in Japanese

2.1.1 Semantic Selection

Predicates like tazuneru 'ask' semantically select an interrogative clause. They can take a clause headed by the interrogative complementizer ka 'Q' but not a clause headed by the declarative complementizer to 'that' as shown by the contrast between (6a) and (6b):

(6) a. *John-wa Bill-ni [Mary-ga kita to] tazuneta
     John-Top Bill-Dat Mary-Nom came that asked
     Lit. 'John asked Bill that Mary came.'
b. John-wa Bill-ni [dare-ga kita ka] tazuneta
   John-Top Bill-Dat who-Nom came Q asked
   ‘John asked Bill who came.’

2.1.2 Complementizer Stacking

As pointed out by Fukui (1986), Saito (2010), Hoshi (2011), and Miyagawa (2011), these two complementizers ka ‘Q’ and to ‘that’ can be stacked in the complement clause selected by tazuneru ‘ask’ as shown in (4) (repeated here as (7)):

(7) John-wa Bill-ni [dare-ga kita ka to] tazuneta
    John-Top Bill-Dat who-Nom came Q that asked
    Lit. ‘John asked Bill that who came.’

Given that selection is local in that an element can only select its sister, a question arises how the matrix predicate tazuneru ‘ask’ can semantically select the interrogative complementizer ka ‘Q’ skipping over the declarative complementizer to ‘that’ in (7).

2.1.3 Syntactic Selection

We cannot simply assume that the declarative complementizer to ‘that’ is transparent for selection in (7). As pointed out by Saito (2010), not all matrix predicates allow complementizer stacking. Although predicates like siritagaru ‘want-to-know’, tyoosasuru ‘investigate’, and hakkensuru ‘discover’ semantically select an interrogative clause as shown in (8), they cannot take complementizer stacking clauses as shown in (9) (Saito 2010: 5):

(8) John-wa [dare-ga kita ka] siritagatteiru
    John-Top who-Nom came Q want-to-know
    Lit. ‘John wants to know who came.’
The contrast between (8) and (9) shows the difference in syntactic selection between *tazuneru* ‘ask’ and *siritagaru* ‘want-to-know’; predicates like *tazuneru* ‘ask’ can syntactically select a clause headed by *to* ‘that’ whereas predicates like *siritagaru* ‘want-to-know’ cannot.

2.1.4 Dual Selections

I argue that dual selections are involved in the complementizer stacking clause (7):

(10) Dual Selections

   a. Semantic selection between *tazuneru* ‘ask’ and the interrogative complementizer *ka* ‘Q’ at LF
   b. Syntactic selection between *tazuneru* ‘ask’ and the declarative complementizer *to* ‘that’ in overt syntax as a driving force of Merge

In the traditional analysis, there are two possible structures of the complementizer stacking clause (7), i.e. the head-complement structure (11a) and the adjunction structure (11b):

(11) a. The Head-Complement Structure

```
       that
       \\
   ask
   \\
     Q
     \\
   TP
```

b. The Adjunction Structure

```
       Q
       \\
   ask
   \\
     that
     \\
   Q
   \\
   TP
```
Given the sisterhood condition on selection, the dual selections cannot be captured by either the head-complement structure or the adjunction structure. The head-complement structure (11a) satisfies syntactic selection, but not semantic selection. The adjunction structure (11b), on the other hand, satisfies semantic selection, but not syntactic selection. What we need is a dual structure, where one syntactic object may have more than one structure, i.e. both the head-complement structure and the adjunction structure. One might argue that if we assume the head-complement structure (11a) and allow the Q-feature of the interrogative complementizer ka ‘Q’ to percolate up the that-phrase, then the matrix predicate tazuneru ‘ask’ can satisfy the dual selections. The that-clause, however, would end up having both an interrogative and non-interrogative feature, which is contradictory; this would result in an anomalous interpretation at LF. It should also be noted that under the bare phrase structure theory this paper is assuming, such partial percolation is not allowed. Hence, this paper does not adopt a partial percolation analysis.

2.2 Complementizer Stacking in Korean

2.2.1 Semantic Selection

In Korean complementizer stacking, a matrix verb semantically selects a mood marker within its complement clause as shown in (12-15):
(12) John-nun [Mary-ka ku mwuncey-lul phwul-ess
John-Top Mary-Nom that problem-Acc solved
*ta/*nya/*la/*ca ko/*nun] cwucangha-ess-ta
Decl/Q/Imp/Exh that claimed
‘John claimed that Mary solved that problem.’

(13) John-nun Mary-eykey [pro kumwuncey-lul phwul-ess
John-Top Mary-Dat that problem-Acc solved
*ta/*nya/*la/*ca ko/*nun] mul-ess-ta
Decl/Q/Imp/Exh that asked
‘John asked Mary whether she solved that problem.’

(14) John-nun Mary-eykey [pro ku mwuncey-lul phwul
John-Top Mary-Dat that problem-Acc solve
*ta/*nya/*la/*ca ko/*nun] myengryengha-ess-ta
Decl/Q/Imp/Exh that ordered
‘John ordered Mary to solve that problem.’

(15) John-nun Mary-eykey [pro ku mwuncey-lul phwul
John-Top Mary-Dat that problem-Acc solve
*ta/*nya/*la/*ca ko/*nun] ceyanha-ess-ta
Decl/Q/Imp/Exh that suggested
‘John suggested to Mary to solve that problem.’

Verbs like *cwucangha* ‘claim’ semantically select the declarative mood marker *ta* as shown in (12). Verbs like *mul* ‘ask’ semantically select the interrogative mood marker *nya* as shown in (13). Verbs like *myengryengha* ‘order’ semantically select the imperative mood marker *la* as shown in (14). Verbs like *ceyanha* ‘suggest’ semantically select the exhortative mood marker *ca* as shown in (15). Similarly, a matrix noun semantically selects a mood marker in its complement clause as shown in (16-19):
(16) [John-i ku mwuncey-lul phwul-ess ta/*nya/*la/*ca
John-Nom that problem-Acc solved Decl/Q/Imp/Exh
*ko/nun] cwucang
that claim
‘the claim that John solved that problem’

(17) [John-i ku mwuncey-lul phwul-ess *ta/nya/*la/*ca
John-Nom that problem-Acc solved Decl/Q/Imp/Exh
*ko/nun] cilmwu
that question
‘the question whether John solved that problem’

(18) [pro ku mwuncey-lul phwul *ta/*nya/la/*ca *ko/nun]
that problem-Acc solve Decl/Q/Imp/Exh that
myenglyeng
order
‘the order to solve that problem’

(19) [pro ku mwuncey-lul phwul *ta/*nya/*la/ca
that problem-Acc solve Decl/Q/Imp/Exh
*ko/nun] ceyan
that suggestion
‘the suggestion to solve that problem’

2.2.2 Syntactic Selection

A question again arises how the matrix predicate can
semantically select a mood marker skipping over ko/nun ‘that’ in
(12-19). We cannot claim that ko and nun are transparent for
selection. Matrix verbs take ko but not nun as shown in (12-15).
Matrix nouns, on the other hand, take nun but not ko as shown in
(16-19). This shows that matrix verbs syntactically select ko
‘that’ whereas matrix nouns syntactically select nun ‘that’. Hence, dual selections are also involved in Korean complementizer stacking; semantic selection between a matrix predicate and a mood marker at LF and syntactic selection between a matrix verb/noun and ka/nun ‘that’ in overt syntax as a driving force of Merge.

3. Evidence against a Direct Quotation Analysis

This section presents evidence to show that complementizer stacking clauses in Japanese and Korean are not direct quotations but complement clauses.

3.1. Polite Forms

First, Miyagawa (1987) observes that direct wh-questions with ka ‘Q’ in Japanese are deviant if the verb is in the plain form without the polite suffix -masu, as shown by the contrast between (20a) and (20b):

(20) a. *Dare-ga kita ka (plain form)
    who-Nom came Q
    ‘Who came?’

b. Dare-ga kimasita ka (polite form)
    who-Nom came Q
    ‘Who came?’

In the complementizer stacking clause (7) (repeated here as (21)), ka ‘Q’ is used with the plain verb form kita ‘came’. This shows that the complementizer stacking clause (21) does not involve a quoted direct wh-question but a complement clauses:
(21) John-wa Bill-ni [dare-ga kita ka to] tazuneta
    John-Top Bill-Dat who-Nom came Q that asked
    Lit. John asked Bill that who came.’ (plain form)

3.2. Extraction

Second, direct quotations are opaque to extraction as shown below:

(22)* What did Mary say, “I am going to buy t?”

(23) Japanese

*Sono situmon-ni Mary-ga, “Dare-ga t tadasiku
that question-Dat Mary-nom who-Nom correctly
kotaeta no kasira” to tazuneta rasii
answered Q Part(icle) that asked seem
Lit. ‘That question, it seems that Mary asked, “Who
answered t correctly?”’

(24) Korean

??Ku mwuncey-lul Meyli-ka “Nwu-ka t cenghwakhi
that problem-Acc Mary-Nom “who-Nom correctly
phwulessulkka?” hako mwul-un kes kathta.
solved that ask fact seems
Lit. ‘That question, it seems that Mary asked, “Who
solved t correctly?”’

Extraction out of a complementizer stacking clause, on the other
hand, is possible in both Japanese (25) and Korean (26). This also
shows that complementizer stacking clauses are not direct
quotations but complement clauses:
(25) Sono situmon-ni Mary-ga [dare-ga t tadasiku
that question-Dat Mary-Nom who-Nom correctly
kotaeta ka to] tazuneta rasii
answered Q that asked seem
Lit. 'That question, it seems that Mary asked that who
answered t correctly.'

(26) Ku kyengchal-eykey Con-i [Meyli-ka t pemin-ul
that policeman-Dat John-Nom Mary-Nom criminal-Acc
sinkohayss nya ko] Pil-eykey mwulessta
reported Q that Bill-Dat asked
Lit. 'That policeman, John asked Bill that whether Mary
reported the criminal t.'

3.3. Pronominal Binding

Third, direct quotations are opaque to pronominal binding as
shown in (27–29):

(27) Mary\textsubscript{1} asked John, "Who cheated her\textsubscript{1/2}?"

(28) Japanese
Mary\textsubscript{1}-ga John-ni, "Dare-ga kanozyo\textsubscript{1/2}-o damasita
Mary-Nom John-Dat who-Nom she-Acc cheated
no kasira," to tazuneta rasii
Q Part that asked seem
'It seems that Mary asked John, "Who cheated her?"'
(29) Korean
Meyli₁-ka pil-eykey “Con-i kunye₁/₂-lul sokyess nya?”
Mary-Nom Bill-Dat John-Nom she-Acc cheated Q
hako mwulessta
that asked
“Mary asked Bill, “Did John cheat her?””

In (27-29), the pronoun within the direct quotation cannot be coreferential with the matrix subject Mary. In (30, 31), on the other hand, the pronoun kanozyo ‘she’ within the complementizer stacking clause can be coreferential with Mary; complementizer stacking clauses are not opaque to pronominal binding:

(30) Mary₁-ga John-ni [dare-ga kanozyo₁/₂-o damasita
Mary-Nom John-Dat who-Nom she-Acc cheated
ka to] tazuneta rasii
Q that asked seem
Lit. ‘It seems that Mary asked John that who cheated her.’

(31) Meyli₁-ka pil-eykey [con-i kunye₁/₂-lul sokyess nya ko]
Mary-Nom Bill-Dat John-Nom she-Acc cheated Q that
mwulessta
asked
Lit. ‘Mary asked Bill that whether John cheated her.’

3.4. De re Readings

Fourth, descriptions in direct quotations may not be interpreted as de re as shown below:

(32) a. Oedipus said, “My mother is pretty.” (Unambiguous)

De dicto: Oedipus knows she is his mother.

*De re: Oedipus doesn’t know she is his mother.
b. Oedipus said his mother is pretty. (ambiguous)

In (32a), *my mother* in the direct quotation can only be interpreted as *de dicto* but not as *de re*. This is in contrast with (32b), where *his mother* in the complement clause can be interpreted as either *de dicto* or *de re*. Similarly, in the Japanese example (33), *sensei* 'teacher' in the direct quotation cannot be interpreted as *de re*; (33) is deviant in the given context. In the Korean example (34), *Meyli* 'Mary' in the direct quotation cannot be interpreted as *de re*, either; (34) is also deviant in the given context.

(33) Context: Mary saw Jack talking with the man who was a stranger to her. She asked me who Jack was talking with. She doesn't know Jack is a teacher. In the classroom, I say to someone else:

# Mary-ga, “Sensei-wa dare-to hanasiteita no
Mary-Nom teacher-Nom who-with was-talking Q
kasira,” to boku-ni tazuneteita yo
Part that I-Dat asked Part
'Mary asked me, “Who was the teacher talking with? ”’

(34) Context: A teacher pointed a girl and asked her to solve the problem. However, the teacher does not know the girl is Mary.

# Sensayngnim-kkeyse Meyli-eykey “Meyli-ka
teacher-Dat Mary-Dat Mary-Nom
ku mwuncey-lul phwules ny?
that problem-Acc solved Q that asked
'A teacher asked Mary, “Did Mary solve the problem? ”’

Turning to complementizer stacking clauses, (35) and (36) are acceptable in the same contexts as (33) and (34) respectively. This
indicates that sensei ‘teacher’ and Meyli 'Mary' in the complementizer stacking clauses can be interpreted as de re. Hence, complementizer stacking clauses are not direct quotations but complement clauses:

(35) In the same context as (33):
Mary-ga [sensei-ga dare-to hanasiteita kato]
Mary-Nom teacher-Nom who-with was-talking Q that
boku-ni tazuneteita yo
I-Dat asked Part
‘Mary asked me that who the teacher was talking with.’

(36) In the same context as (34):
Sensayngnim-kkeye Meyli-eykey [Meyli-ka
teacher-Dat Mary-Dat Mary-Nom
ku mwuncey-lul phwuless nya ko] mwulessta
that problem-Acc solved Q that asked
Lit. ‘A teacher asked that whether Mary solved that problem.’

3.5. Temporal Modifiers

Fifth, temporal modifiers in direct quotations are evaluated relative to subjects whereas those in complement clauses are evaluated relative to utters:

(37) a. Mary said, “I will come to the party tomorrow.”
b. Mary said she would come to the party tomorrow.

In (37a), the temporal modifier tomorrow in the direct quotation is evaluated relative to the subject Mary; tomorrow is interpreted as the day after Mary’s saying or asking. In (37b), on the other hand, tomorrow in the complement clause is evaluated relative to
the utter; tomorrow is interpreted as the day after speech act. In a similar vein, asita ‘tomorrow’ in the Japanese example (38) and nayil ‘tomorrow’ in the Korean example (39) in the direct quotations are evaluated relative to the subject Mary:

(38) Mary-ga John-ni, “Dare-ga asita paatii-ni kuru
      Mary-Nom John-Dat who-Nom tomorrow party-to come
      no kasira,” to tazuneteita yo
      Q Part that asked Part
      ‘Mary asked John, “Who comes to the party tomorrow?”’

(39) Meyli-ka Con-eykey, “Ney-ka nayil phathi-ey ka la,”
      Mary-Nom John-Dat you-Nom tomorrow party-to go Imp
      hako myenglyenghayssta
      that ordered
      ‘Mary ordered John, “You should go to the party tomorrow.”’

Turning to complementizer stacking clauses, asita ‘tomorrow’ in the Japanese example (40) and nayil ‘tomorrow’ in the Korean example (41) are interpreted as the day after speech act. This shows that complementizer stacking clauses are not direct quotations but complement clauses:

(40) Mary-ga [dare-ga asita paatii-nikuru ka to]
      Mary-Nom who-Nom tomorrow party-to come Q that
      John-ni tazuneteita yo
      John-Dat asked Part
      ‘Mary asked John that who would come to the party the next day.’
(41) Meyli-ka Con-eykey [pro nayil phathi-ey ka la ko]
Mary-Nom John-Dat tomorrow party-to go Imp that
myenglyenghayssta.
ordered
'Mary ordered John to go to the party the next day.'

3.6. Evaluative Predicates

Sixth, evaluative predicates in direct quotations are evaluated relative to subjects whereas those in complement clauses are evaluated relative to utters:

(42) a. Mary said, “I love that idiot.”
    b. Mary said she loves that idiot.

In (42a), where *that idiot* is within the direct quotation, it is evaluated relative to the subject *Mary*; it is *Mary* who thinks that he is an idiot. In (42b), on the other hand, *that idiot* is in the complement clause so that it is evaluated relative to the utter; it is the utter who thinks that he is an idiot. Similarly, *ano orokamono ‘that idiot’* in (43) and *ku papo.casik ‘that idiot’* in (44), which are within the direct quotations, are evaluated relative to the subject *Mary*:

(43) Mary-wa John-ni, ”Dare-ga ano orokamono-ni
Mary-Top John-Dat who-Nom that idiot-Dat
taikin-o watasita no kasira,” to tazuneta
a lot of money-Acc gave Q Part that asked
'Mary asked John, “Who gave a lot of money to that idiot?”'
When *that idiot* is in a complementizer stacking clause, it is evaluated relative to the utter as shown in (45, 46). This shows that complementizer stacking clauses are complement clauses:

(45) Mary-wa  John-ni [dare-ga  ano orokamono-ni
Mary-Top John-Dat who-Nom  that idiot-Dat
taikin-o  watasitaka to]  tazuneta
a lot of money-Acc gave  Q  that asked
Lit. ‘Mary asked John that who gave a lot of money to that idiot.’

(46) Meyli-ka Con-eykey  ku papo.casik-eykey kekum-ul
Mary-Nom John-Dat  that idiot-Dat  money-Acc
cwuess nya ko  mwulessta
gave  Q  that asked
Lit. ‘Mary asked John that whether he gave money to that idiot.’

### 3.7. Deictic Terms

Finally, deictic terms within direct quotations are evaluated relative to subjects whereas those in complement clauses are evaluated relative to utterers:

(47) a. Mary said, “I want this picture.”

b. Mary said that she wanted this picture.

In (47a), where the deictic term *this* appears within the direct
quotation, it is evaluated relative to the subject Mary; this picture is near Mary. In (47b), on the other hand, this appears within the complement clause so that it is evaluated relative to the utter; this picture is near the utter. In a similar vein, when a deictic term appears in a direct quotation in Japanese and Korean, it is evaluated relative to the subject, as shown in (48, 49):

(48) Mary-wa John-ni, “Dare-ga kono e-o kaita
Mary-Top John-Dat who-Nom this picture-Acc drew
no kasira,” to tazuneta
Q Part that asked
‘Mary asked John, “Who draw this picture?”’

(49) Meyli-ka Con-eykey “i kulim-ul Saym-eykey
Mary-Nom John-Dat this picture-Acc Sam-Dat
cwuess nya” hako mwulessta
gave Q that asked
‘Mary asked John, “Did you give this picture to Sam?”’

When the deictic term kono e ‘this picture’ or i kulim ‘this picture’ appears in a complementizer stacking clause, it is evaluated relative to the utter as shown in (50, 51). Hence, complementizer stacking clauses are complement clauses:

(50) Mary-wa John-ni [dare-ga kono e-o kaita
Mary-Top John-Dat who-Nom this picture-Acc draw
ka to] tazuneta
Q that asked
Lit. ‘Mary asked John that who draw this picture.’
(51) Meyli-ka Con-eykey i kulim-ul Saym-eykey
Mary-Nom John-Dat this picture-Acc Sam-Dat
cwuess nya ko mwulessta
gave Q that asked
Lit. ‘Mary asked John that whether he gave this picture to Sam.’

4. A Proposal

4.1. Labeling Conflicts and Cartographic Structure Building

Before turning to an analysis of the dual selections in complementizer stacking, I will explicate labeling conflicts and cartographic structure building. The notion of labeling conflict was proposed by Donati (2006), Chomsky (2008), and Cecchetto and Donati (2010; 2011). They argue that a labeling conflict arises in cases like (52). In (52), α is a lexical item, i.e. a head, and β is not a head, and α undergoes Internal Merge with β:

(52) [α [β α]]
The labeling algorithms (3) (repeated here as (53)) make conflicting predictions:

(53) Labeling Algorithms (Chomsky 2008: 145)

a. In {H, α}, H an LI, H is the label.
b. If α is internally merged to β, forming {α, β}, then the label of β is the label of {α, β}.

According to (53a), α, which is a head, should become the label. According to (52b), however, β, which is the target of Internal Merge, should become the label. They claim that a labeling conflict makes the two different labels available, which creates an ambiguous structure.
Based on their labeling conflict idea, I argue that a labeling conflict creates not only an ambiguous structure but also a dual structure in the sense that one syntactic object has different structures in overt syntax and LF. Such a dual structure is created by relabeling as part of LF-Transfer. It is reasonable to assume that LF-Transfer can trigger relabeling, where the asymmetric relation between merged elements changes. It should be noted that PF-Transfer alters the (a)symmetric relation between merged elements, converting an unordered set \{X, Y\} into an ordered pair \langle X, Y \rangle or \langle Y, X \rangle. Hence, I claim that Transfer can alter the (a)symmetric relation between merged elements due to linearization in PF-Transfer and relabeling in LF-Transfer.

This paper claims with Shlonsky (2006) that cartographic structure in the sense of Rizzi (1997) is created by self-attachment of C as stated in (54):

(54) a. C is associated with an ordered set of lexical items (or bundles of features if C is null) \langle C_1, \ldots, C_n \rangle. These precompiled lexical items (or bundles of features) correspond to Rizzi's (1997) Fin, Foc, Top, etc.

b. The computational system activates these lexical items (or bundles of features) one by one respecting the order of compilation; (i) Initial merger of C activates the leftmost lexical item (or bundle of features) in the set, (ii) Each time C is internally merged to itself, i.e., self-attachment of C takes place, the leftmost inactive lexical item (or bundle of features) in the set is activated.
c. Once $C_m$, $m > 1$, is activated, $C_{m-1}$ is no longer visible to the computational system.

4.2 A Relabeling Analysis of Complementizer Stacking

Let us look at how relabeling derives the dual structure property of complementizer stacking, thereby accounting for its dual selections. Let us consider the Japanese example (7) (repeated here as (55)) as an example. The derivation of (55) proceeds as represented in (56):

(55) John-wa Bill-ni [dare-ga kita ka to] tazuneta
John-Top Bill-Dat who-Nom came Q that asked
Lit. 'John asked Bill that who came.'

(56) a. TP [ka 'Q', to 'that']

b. ka 'Q'

TP [ka 'Q', to 'that']

c. Self-Attachment of C

ka 'Q' [ka 'Q', to 'that']

TP [ka 'Q', to 'that']

d. Labeling in Overt Syntax

to 'that'

ka 'Q' [ka 'Q', to 'that']

TP [ka 'Q', to 'that']
C consists of the ordered set \(<\text{ka 'Q', to 'that'}\>). External Merge applies to TP and C \(<\text{ka 'Q', to 'that'}\>). By this initial merger of C \(<\text{ka 'Q', to 'that'}\>\), the leftmost lexical item in the set, i.e. the interrogative complementizer ka 'Q', becomes activated. This merger yields \{TP, \(<\text{ka 'Q', to 'that'}\>\}\} as represented in (56a). In (56a) and relevant representations to follow, the activated lexical item (or bundle of features) is underlined. The labeling algorithms in (53) require that \(<\text{ka 'Q', to 'that'}\>\), which is a lexical item, should become the label of \{TP, \(<\text{ka 'Q', to 'that'}\>\}\} in accordance with the labeling algorithm (53a); (56b) is labeled ka 'Q' (= \(<\text{ka 'Q', to 'that'}\>\)). Next, we apply self-attachment of C, i.e. attachment (movement) of \(<\text{ka 'Q', to 'that'}\>\) to itself. In other words, \(<\text{ka 'Q', to 'that'}\>\) is internally merged to the label ka 'Q' (= \(<\text{ka 'Q', to 'that'}\>\)). When \(<\text{ka 'Q', to 'that'}\>\) is internally merged to itself, the leftmost inactive lexical item in the set to 'that' becomes activated and ka 'Q', which was activated in the previous Merge, becomes no longer visible to the computational system; this results in \(<\text{ka 'Q', to 'that'}\>\). Hence, self-attachment of C yields \{ka 'Q' (= \(<\text{ka 'Q', to 'that'}\>\), ka 'Q', to 'that'>\}, as represented in (56c). A labeling conflict arises in (56c); the labeling algorithm (53a) requires that \(<\text{ka 'Q', to 'that'}\>\), which
is a lexical item, should become the label, whereas the labeling algorithm (53b) requires that the label ka ‘Q’ (= ka ‘Q’, to ‘that’), which is the target of Internal Merge, should become the label. I argue that this labeling conflict creates a dual structure. We have two labeling options; either to ‘that’ (= ka ‘Q’, to ‘that’) or ka ‘Q’ (= ka ‘Q’, to ‘that’)) becomes the label. We take the former option here; to ‘that’ becomes the label in accordance with (53a) in overt syntax, as represented in (56d). This labeling drives Merge with the matrix predicate tazuneru ‘ask’, satisfying the syntactic selection of tazuneru ‘ask’. When we come to the stage where LF-Transfer applies, relabeling applies as part of LF-Transfer; ka ‘Q’ (= ka ‘Q’, to ‘that’) becomes the label in accordance with (53b) as represented in (56e). This label satisfies the semantic selection of the matrix predicate tazuneru ‘ask’ at LF. Hence, relabeling due to the labeling conflict creates a dual structure where the complementizer stacking clause (55) is labeled to ‘that’ in overt syntax but ka ‘Q’ at LF; this accounts for the puzzling dual selections. The dual selections in Korean complementizer stacking can be accounted for in the same way.

Before closing this section, let us consider predicates like omou ‘think’, yuu ‘say’, and sakebu ‘scream’. These predicates syntactically and semantically select a complement clause headed by the declarative complementizer to ‘that’ as shown in (57):

(57) a. John-wa [Mary-ga kita to] omotta/itta/sakenda
     John-Top Mary-Nom came that thought/said/screamed
     ‘John thought/said/screamed that Mary came.’
b. *John-wa [dare-ga kita ka] omotta/itta/sakenda
   John-Top who-Nom came Q thought/said/screamed
   Lit. 'John thought/said/screamed who came.'

As pointed out by Saito (2010), these predicates can also take a complementizer stacking clause as shown in (58):

(58) John-wa [dare-ga kita ka to] omotta/itta/sakenda
   John-Top who-Nom came Q that thought/said/screamed
   Lit. 'John thought/said/screamed that who came.'

Under our analysis, the derivation of (58) proceeds almost in the same way as derivation (55). The only difference between these two derivations is that relabeling due to a labeling conflict, which is optional, does not apply in the derivation of (58). The declarative complementizer to 'that', which is the label in overt syntax, remains as the label at LF; this satisfies the syntactic and semantic selections of these predicates.

5. Conclusion

This paper has investigated complementizer staking in Japanese and Korean. It was shown that complementizer stacking involves dual selections, which cannot be accounted for by either the traditional head-complement or adjunction structure. I have then proposed relabeling as part of LF-Transfer due to a labeling conflict, showing that it accounts for the dual selections. The proposed analysis presents further evidence for the symmetric Merge together with labeling algorithms approach.
References:


Miyagawa, Shigeru. 2011. Agreements that Occur Mainly in the Main Clause. Ms., MIT.


Toru Ishii
School of Arts and Letters
Meiji University
1-1 Kandasurugadai, Chiyoda-ku
Tokyo, 101-8301, JAPAN
tishii@meiji.ac.jp