Minimal Recursion Semantics: An Application into Korean

Jong-Bok Kim
(Kyung Hee University)

Kim, Jong-Bok. 2006. Minimal Recursion Semantics: An Application into Korean. *The Linguistic Association of Korea Journal, 14*(2), 59-85. Minimal recursion semantics (MRS) is a framework for semantics within HPSG that is considerably suitable for parsing and generation. This paper illustrates the application of MRS into Korean and see how the nonrecursive semantic representation offers us descriptively adequate semantic representations for the various phenomena (including scope ambiguities) in the language.

Key Words: minimal recursion semantics, scope ambiguity, Korean, typed feature structure

1. Introduction

1.1. Motivations

Minimal Recursion Semantics (MRS), developed by Copestake et al. (2003), is a framework of computational semantics designed to enable semantic composition using only the unification of type feature structures (Carpenter 1992, Flickinger 2000). This allows the grammar to produce and generate for each phrase or sentence a description of the meaning representation sufficient to support logical inference. In particular, its development is couched upon achieving the following criteria as a framework of computational semantics (Copestake et al. 2003):

- Expressive Adequacy: the framework must allow linguistic meanings to be expressed correctly.

- Grammatical Compatibility: Semantic representations must be linked cleanly to other kinds of grammatical information (most notably syntax).

- Computational tractability: It must be possible to process meanings and to check semantic equivalence efficiently and to express relationships between semantic representations straightforwardly.

- Underspecifiability: semantic representations should allow underspecification, in such a way as to allow flexible, monotonic resolution of such partial semantic representations

To achieve these four main criteria, MRS introduces a syntactically 'flat' representation expressing meanings by feature structures. This flat, feature-based semantic representations make MRS suitable for large general purpose grammars for use in parsing, generation, and semantic transfer (cf. Flickinger and Bender 2003, Bender et al. 2002). Considering 'semantic transfer' in which a source utterance is parsed to give a semantic representation and a transfer component converts this into a target representation, we need to reduce spurious ambiguities, flat structures could make semantic transfer rules much simpler and avoids spurious ambiguities.

This paper is an attempt to adopt the MRS framework for Korean semantic representations and check the possibility of implementing it in a Korean grammar. We will see how the MRS system will work for representing its semantic compositions as well as scope ambiguities we often find in the language.

1.2. MRS Representations

To figure out the basics of MRS, let us consider one simple sentence:

(1) Every big white horse sleeps.

The semantic of this can be represented as a conventional calculus in(2):

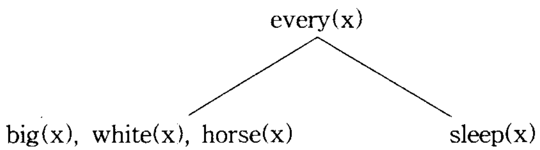
(2) $\forall_x[[[\text{big}(x) \wedge [\text{white}(x) \wedge \text{horse}(x)]]], \text{sleep}(x)]$

As pointed out by Copestake et al. (2003), One potential problem of such conventional semantic representations comes from the fact that the form of these semantic representations implicitly includes information about the syntactic structure even though it is irrelevant to their semantic representation. In particular the binary nature of the operator $\wedge$ leads to a spurious ambiguity in representation, since the bracketing is irrelevant to the truth conditions. This causes a main problem in semantic transfer approaches to MT (machine translation). Semantic transfer refers to an approach where a source utterance is parsed to give a semantic representation and a transfer component converts this into a target representation which is an input to a generator to produce a string in the target language. From a generation point of view, an efficient generator thus needs to accept the input logical form and generate a target sentence with no spurious ambiguities. As pointed One efficient way of achieving this goal is to refer to flat semantic representations like the following:

(3) every(x), big(x), white(x), horse(x), sleep(x)

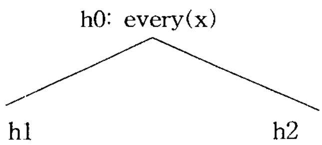
However, this does not capture the obvious scope relation of *every* as given in (4):

(4)



MRS is an attempt to introduce such flat semantic representations while underspecifying quantifier scopes. To do this, MRS considers the nodes of a tree independently of any parent or daughter; it reifies the links in the tree by using tags which match up scopal argument slots with the elementary predicates (EPs). These tags are 'handles' that enable us to grab hold of an EP. Each EP has a handle which identifies it as belonging to a particular tree node (label), as represented in the following tree representation:

(5)



This tree is equivalent to the following simple flat list of labelled EPs:

(6) h0: every(x, h1, h2), h1: big(x), h1: white(x), h1: horse(x), h2: sleep(x)

The same mechanism can be introduced for Korean. Let us consider one simple example.

(7) haksayng-tul-i sakwa-lul mek-ess-ta
student-PL-NOM apple-ACC eat-PST-DECL
'Students ate apples.'

The MRS of this sentence is given in (8):

(8) h1: student(x), h2: apple(x), h3: eat(x,y)}

Unlike the Davidsonian representation, no event variable is introduced, but each EP (elementary predicate) has a *handle*. This handle functions as a referring to the EP from other EPs. Consider another sentence with a quantifier and its MRS representation:

(9) a. sakwa-lul motun haksayng-tul-i mek-ess-ta
apple-ACC all student-PL-NOM eat-PST-DECL
'All students ate the apples.'

b. h0: all(x, h1, h3), h1: student(x), h2: apple(y), h3: eat(x,y)

In (9b), *all(x,h1,h3)* here means that the number of x's, satisfying both the EP with handle h1 and the EP with handle h3, is 'all'. This corresponds to the following generalized quantifier notation:

(10) $\forall_x$[student(x), $?_x \exists_y$[apple(y) & eat (x,y)]]

The MRS representation in (9b) is flatter than (10) in that in the former the quantifier takes only an individual variable and handles, while in the latter the quantifier takes two properties (of type <e,t>) as arguments. This kind of flat semantics is also well-suited to capture semantic ambiguities too:

(11) motun haksayng-tul-i sakwa-lul mek-ci anh-ass-ta
all student-PL-NOM apple-ACC eat-COMP not-PST-DECL
'All students didn't eat the apples.'

This sentence has two interpretations with respect to the scope of negation *anh-* 'not' and quantifier *motun* 'all'.

(12) a. $\forall_x$[student(x), $?_x \exists_y$[apple(y) & eat (x,y)]]
b. $\forall_x$[student(x), $?_x \exists_y$[apple(y) & eat (x,y)]]

(12a) induces the reading such that there are all students who didn't eat an apple/apples. Meanwhile, (12b) has the reading such that it is not the case that all the students ate an apple/apples. In the MRS representation, these two readings will be expressed in the following way:

(13) h0: all(x, h1, h7), h1: student(x), h2: apple(y), h3: eat(x,y), h4: not(h8)

One thing to note here is that the handles h7 and h8 are not associated with any EP. If we make h7 = h4 and h8 = h3, then we have the interpretation (12a) in which *all* has a wide scope and thus outscopes the negator *not*. Meanwhile, if we make h7=h3 and h8=h0, we have the interpretation (12b) in which the negation has a wider scope over *all*.

In sum, MRS uses semantic representations with a flat syntax, eliminating syntactic embedding. This brings us a simple and efficient method for underspecification of quantifier scope. This is achieved by handle variable values: scope features are associated with handle variable values in the underspecified cases. These variables are associated with constraints concerning scope. The system eventually allows us to capture the semantic ambiguity in a monotonic way without resorting to different syntactic structures at LF or additional semantic device such as Cooper's storage.[1)]

2. MRS in Typed Feature Structure

MRS, a system of semantic representation, can be systematically represented in terms of typed feature structures (Bender et al. 2002). The semantic representations assigned to each word or phrase in MRS has at least the following basic feature structures:[2)]

---

1) See Gunji (2005) for Japanese examples.

2) The type *mrs* has two subtypes *nom-obj* and *psoa*, corresponding to semantic representations of nominal signs and predicative signs.

(14)
$$\begin{bmatrix} \textit{mrs} \\ \text{HOOK } \textit{hook} \\ \text{RELS } \textit{diff-list} \\ \text{HCONS } \textit{diff-list} \end{bmatrix}$$

As given in the feature structure, the type mrs has three basic attributes: HOOK, RELS, and HCONS. Let us consider the role of each attribute in detail.[3]

RELS: The attribute RELS, the heart of an mrs, is basically a bag of elementary predications (EP) whose value is a relation. Each relation has at least three features LBL, PRED, and ARG0:

(15)
$$\begin{bmatrix} \text{RELS} \left\langle \ldots, \begin{bmatrix} \textit{relation} \\ \text{PRED } \textit{string} \\ \text{LBL } \textit{handle} \\ \text{ARG0 } \textit{individual} \end{bmatrix}, \ldots \right\rangle \end{bmatrix}$$

The value of LBL is a handle, which is a token to its EP. The value of PRED is a string, serving to distinguish particular relations.[4] All EP will have at least one argument ARG0 whose value is either *ref-ind* for nominal expressions or *event-ind* for verbal expressions.[5] Depending on the type of EP, additional arguments will be added. The following is the basic relation value of sample lexical elements:

---

3) '*diff-list*' is different from '*list*' in that it allows us to point to the last element in a list. To be more precise, this type introduces LIST and LAST as its attributes.

4) Each part of speech will have the following relations: noun: *n_rel*, verb: *v_rel*, adjective: *j_rel*, adverb: *r_rel*, determiner: *q_rel*, message: *m_rel*, all other closed class: *x_rel*

5) This in turn means that these two types are subtypes of individual.

(16)

a. $\begin{bmatrix} \text{PRED } \_student\_n\_rel \\ \text{LBL } h1 \\ \text{ARG0 } x \end{bmatrix}$ b. $\begin{bmatrix} \text{PRED } \_apple\_n\_rel \\ \text{LBL } h2 \\ \text{ARG0 } y \end{bmatrix}$

c. $\begin{bmatrix} \text{PRED } \_eat\_v\_rel \\ \text{LBL } h3 \\ \text{ARG0 } e1 \\ \text{ARG1 } x \\ \text{ARG2 } y \end{bmatrix}$ d. $\begin{bmatrix} \text{PRED } \_happy\_j\_rel \\ \text{LBL } h4 \\ \text{ARG0 } e1 \\ \text{ARG1 } x \end{bmatrix}$

The RELS value also includes a type *message* when a clause is involved. The semantic type that clauses express (e.g., commands, questions, or propositions) triggers the inclusion of one of the *message* types in the RELS value:[6]

(17) $\begin{bmatrix} message \\ \text{PRED } prpstn\_m\_rel \\ \text{LBL } handle \\ \text{MARG } handle \end{bmatrix}$

This message relation introduces the attribute MARG (message argument) whose value is the highest scoping handle of the clause.

In Korean, this *message* relation is introduced together with the mood marker. Thus the verb *mek-ess-ta* 'eat' will introduce the following EP in addition to the one in 16c:[7]

6) In addition, message has subtypes such as *command_m_rel* and *question_m_rel*.

7) In English and Japanese, this message value is introduced by a construction rule (cf. Bender et al. 2002, Siegel 2000, Siegel and Bender 2002). However, the message type in Korean is marked by a mood marker, it better be introduced by the type of mood marker.

(18)

$$\begin{bmatrix} \textit{message} \\ \text{PRED } \textit{prpstn\_m\_rel} \\ \text{LBL } \textit{h0} \\ \text{MARG } \textit{h3} \end{bmatrix}$$

HOOK: In obtaining the semantics of a phrase, we often need to make a given feature value externally visible. This information is encoded in the feature HOOK, a group of distinguished externally visible attributes of the atomic predications in RELS, used in combining the semantics of this sign with the semantics of other signs. It basically has the following feature information:

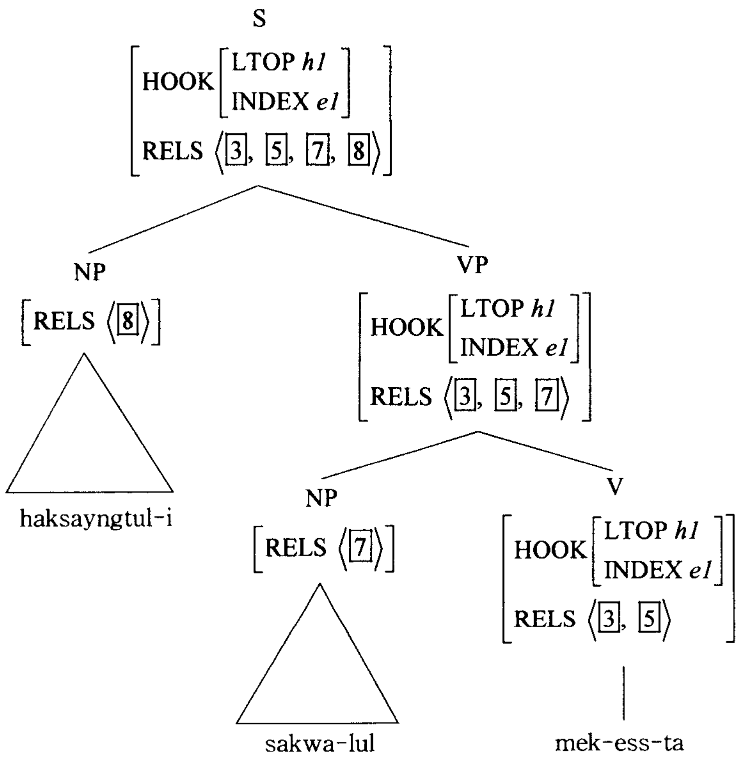
(19)

$$\begin{bmatrix} \text{HOOK} \begin{bmatrix} \textit{hook} \\ \text{LTOP } \textit{handle} \\ \text{INDEX } \textit{individual} \\ \text{XARG } \textit{individual} \end{bmatrix} \end{bmatrix}$$

The value of LTOP is the local top handle, the handle of the relations with the widest scope within the constituent. This is also accessed by semantic heads in phrasal constructions in order to impose further scopal constraints. The value of INDEX is identified with the INDEX of the semantic head daughter. The value of XARG (external argument) is the index of the single argument in a phrase. This information will be accessed by semantic heads in raising and control constructions.

Together with these feature declarations so far, let us consider a simple sentence like (7) *Haksayng-tul-i sakwa-lul mek-ess-ta* 'Students ate apples'. The semantic composition process of this sentence can be more clearly represented in a tree format:

(20)



The syntactic combination of the given expressions is guaranteed by Korean grammar rules (see Kim 2004 and Kim and Yang 2004)[8]: The Head-Complement rule will allow the combination of the head *mek-ess-ta* and its complement *sakwa-lul*.[9]

The Head-Subject rule allows the resulting VP phrase to combine with the subject *haksayngtul-i*. In the structure (19), we can notice that each expression has its own RELS value. These values will be passed up to the mother. The mother's RELS value is the sum of its daughters RELS value. The mother's LTOP and INDEX values are

8) Kim (2004) posits Korean grammar rules (similar to Korean X' rules) such as Head-Subject, Head-Complement, Head-Modier Rule, and Head-Filler Rule. See Kim (2004) for details.

9) The boxed numbers in the tree here match with those in (21).

identical to those of its head daughter, resulting in the final MRS representation on the top S here given in (21):

(21)

$$\begin{bmatrix} \text{HOOK} \begin{bmatrix} \text{LTOP } h1 \\ \text{INDEX } e1 \end{bmatrix} \\ \text{RELS} \left\langle \begin{array}{l} \boxed{5} \begin{bmatrix} \text{PRED } prpstn\_m\_rel \\ \text{LBL } h1 \\ \text{MARG } h2 \end{bmatrix}, \boxed{3} \begin{bmatrix} \text{PRED } \_eat\_v\_rel \\ \text{LBL } h2 \\ \text{ARG0 } e1 \\ \text{ARG1 } i \\ \text{ARG2 } j \end{bmatrix}, \\ \boxed{8} \begin{bmatrix} \text{PRED } \_student\_n\_rel \\ \text{LBL } h3 \\ \text{ARG0 } i \end{bmatrix}, \boxed{7} \begin{bmatrix} \text{PRED } \_apple\_n\_rel \\ \text{LBL } h4 \\ \text{ARG0 } j \end{bmatrix} \end{array} \right\rangle \end{bmatrix}$$

The noun *haksayng-tul-i* 'student-PL-NOM' and *sakwa-lul* 'apple' will introduce the appropriate EPs as given here. The verb *mek-ess-ta* introduces *_eat_v_rel* as well as *_prpstn_m_rel*. The LBL (LABEL) value of these two EPs will eventually function as INDEX and LTOP value of the whole sentence as indicated here. That is, this sentence is a propositional message denoted by the event *e1*. The event is an eat event in which two arguments *i* and *j* participate.[10)]

HCONS: When quantifiers are involved, the RELS values are also enriched. First of all, quantifiers select different features. The quantifiers (which has a *quant-relation*) will have RESTR and BODY as additional arguments, as given in (22):

---

10) Throughout the paper, in order to avoid the complexity, we suppress the undefined quantificational information each bare NP has. To have a complete system, we need to have a system in which each NP has quantificational information. See Bender et al. 2002 and Siegel and Bender 2002.

(22)

$$\text{a.}\begin{bmatrix}\text{PRED } \_all\_q\_rel \\ \text{LBL } h0 \\ \text{ARG0 } i \\ \text{RESTR } h1 \\ \text{BODY } h7\end{bmatrix} \qquad \text{b.}\begin{bmatrix}\text{PRED } \_some\_q\_rel \\ \text{LBL } h0 \\ \text{ARG0 } i \\ \text{RESTR } h1 \\ \text{BODY } h7\end{bmatrix}$$

The RESTR value is related to the top handle of the quantifier's restriction (the N′ that combines with the given quantifier) whereas the BODY is left unbounded.

In order to resolve the unbounded value, MRS introduces the feature HCONS. This value can be resolved in such a way that the quantifiers '리oat in' wherever there is a 'space' left by a *qeq* (equality modulo quantifiers) constraint. This handle constraint is represented by the feature HCONS whose value is a bag of *qeq* relations:

(23)

$$\begin{bmatrix}\text{HCON}\left\langle \ldots, \begin{bmatrix}qeq \\ \text{HARG } handle \\ \text{LARG } handle\end{bmatrix}, \ldots \right\rangle\end{bmatrix}$$

HARG is identified with the handle-taking argument position (usually quantifier) whereas LARG is identified with the LBL of the outscoped N‘ relation. The value of HARG and LARG, like that of LBL and all role features (e.g., ARG0, etc), are all objects of type *semarg*, which in turn has two subtypes *handle* and *individual*.[11] This system then assigns more enriched MRS representations to quantifiers like *motun* 'all' than those in (22):

11) The type *individual* has two subtypes *ref-ind* for nominal elements and *event-ind* for verbal elements. The type *event-ind* will have TENSE, ASPECT, and MOOD features whereas *ref-ind* will have PER, NUM, and GEN features. For the reason of simplicity, we do not discuss these features here. See Copestake et al. 2004.

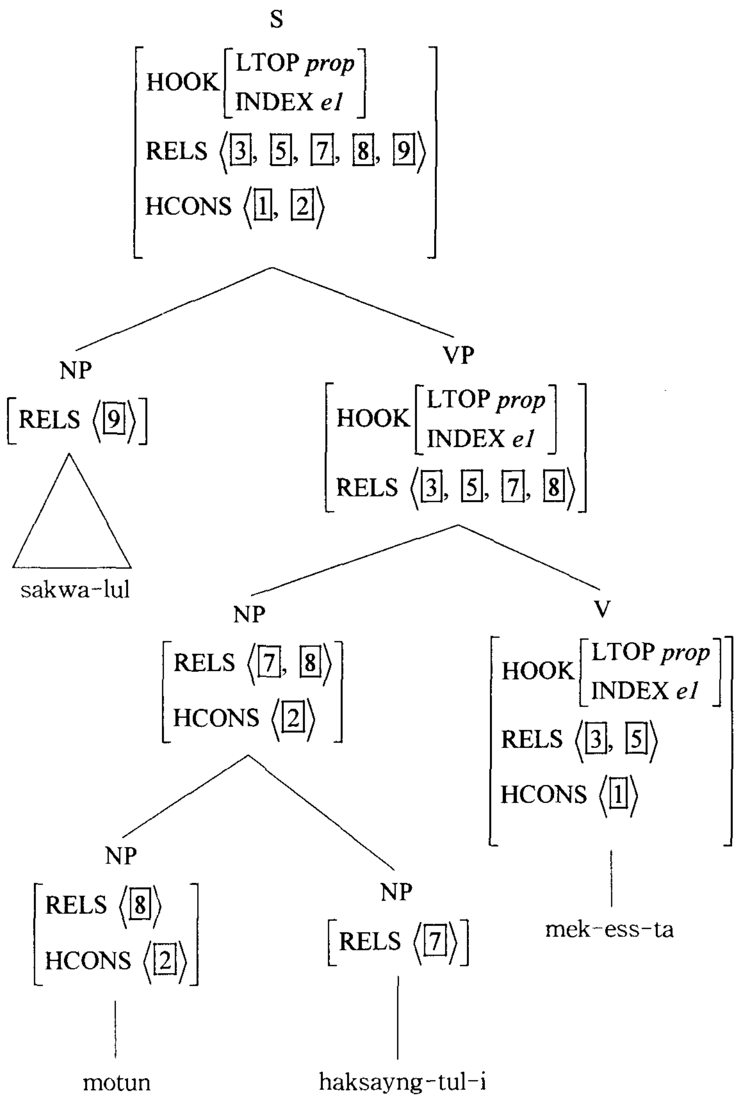
(24)

$$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{POS } \textit{det} \\ \text{SPEC} \left\langle \begin{bmatrix} \text{POS } \textit{noun} \\ \text{LTOP } \textit{h4} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{HOOK} & \begin{bmatrix} \text{LTOP } \textit{handle} \\ \text{INDEX } \textit{i} \end{bmatrix} \\ \text{RELS} & \left\langle \begin{bmatrix} \text{PRED } \textit{\_all\_q\_rel} \\ \text{LBL } \textit{h0} \\ \text{ARG0 } \textit{i} \\ \text{RESTR } \textit{h1} \\ \text{BODY } \textit{handle} \end{bmatrix} \right\rangle \\ \text{HCONS} & \left\langle \begin{bmatrix} \textit{qeq} \\ \text{HARG } \textit{h1} \\ \text{LARG } \textit{h4} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

Syntactically, the determiner *motun* species a nominal element as indicated by the feature SPEC. Semantically, it restricts an EP with the handle value *h1* which in turn is, through the HCONS value, linked to the nominal element it specifies.

With these enriched feature structures, let us consider the structure of (9a):

(25)



As we have seen so far, all the linguistic objects, words and phrases, have these three features (HOOK, RELS, and HCONS). As the expressions are combined according to grammar rules such as

Head-Subject and Head-Complement Rule, these features are accordingly accumulated by the Semantic Principles (cf. Bender et al. 2002, Siegel and Bender 2002):

(26) Semantic Principles:

a. The value for RELS on the mother of a phrase is the result of appending the RELS values of all its daughters.

b. The value for HCONS on the mother of a phrase is the result of appending the HCONS values of all of its daughters.

c. The value for HOOK on the mother of a phrase is identified with the HOOK value of its semantic head daughter.[12)]

The principles will then generate the following final MRS:

12) Where each phrase type uniquely determines which of the daughters is the semantic head.

(27)

[HOOK [LTOP *h1*, INDEX *e1*]

RELS ⟨ 5 [PRED *_prpstn_m_rel*, LBL *h1*, MARG *h9*], 3 [PRED *_eat_v_rel*, LBL *h3*, ARG0 *e1*, ARG1 *i*, ARG2 *j*],
8 [PRED *_all_q_rel*, LBL *h7*, ARG0 *i*, RESTR *h2*, BODY *h10*], 7 [PRED *_student_n_rel*, LBL *h4*, ARG0 *i*],
9 [PRED *_apple_n_rel*, LBL *h8*, ARG0 *j*] ⟩

HCONS ⟨ 1 [*qeq*, HARG *h9*, LARG *h3*], 2 [*qeq*, HARG *h2*, LARG *h4*] ⟩]

As we can notice here, the quantifier all induces a *qeq* value in the HCONS. The *qeq* identifies the RESTR value of the quantifier with its HARG value and the outscoped N's LBL value as its LARG. The introduction of *qeq* for semantics, though rather cumbersome, allows the right ranger of variation in quantifier scope. (cf. Copestake et al. 2003). The semantic principles I and II eventually allow the meaning representation to keep pace as the syntactic analysis of a grammar grows in complexity.

3. Applications to Complex Cases

3.1. Verbal Modifiers

Let us consider an example with a sentential modifier:

(28) motun haksayng-tul-i ama o-kess-ney
all student-PL-NOM probably come-FUT-DECL
'All students probably come.'

Since the adverb *ama* 'probably' is scopal, we need to represent this in the MRS representations too. The EP *ama* will have the following feature structure:

(29)

$$\left[\begin{array}{ll} \text{SYN} & \left[\text{HEAD} \left[\begin{array}{l}\text{POS } adv \\ \text{MOD}\left\langle\left[\begin{array}{l}\text{POS } verb\\ \text{INDEX } e1\\ \text{LTOP } h8\end{array}\right]\right\rangle\end{array}\right]\right] \\ \text{SEM} & \left[\begin{array}{ll} \text{HOOK} & \left[\begin{array}{l}\text{LTOP } h3\\ \text{INDEX } e1\end{array}\right]\\ \text{RELS} & \left\langle\left[\begin{array}{l}\text{PRED } \_probably\_adv\_rel\\ \text{LBL } h3\\ \text{ARG1 } h9\end{array}\right]\right\rangle \\ \text{HCONS} & \left\langle\left[\begin{array}{l} qeq\\ \text{HARG } h9\\ \text{LARG } h8\end{array}\right]\right\rangle\end{array}\right]\end{array}\right]$$

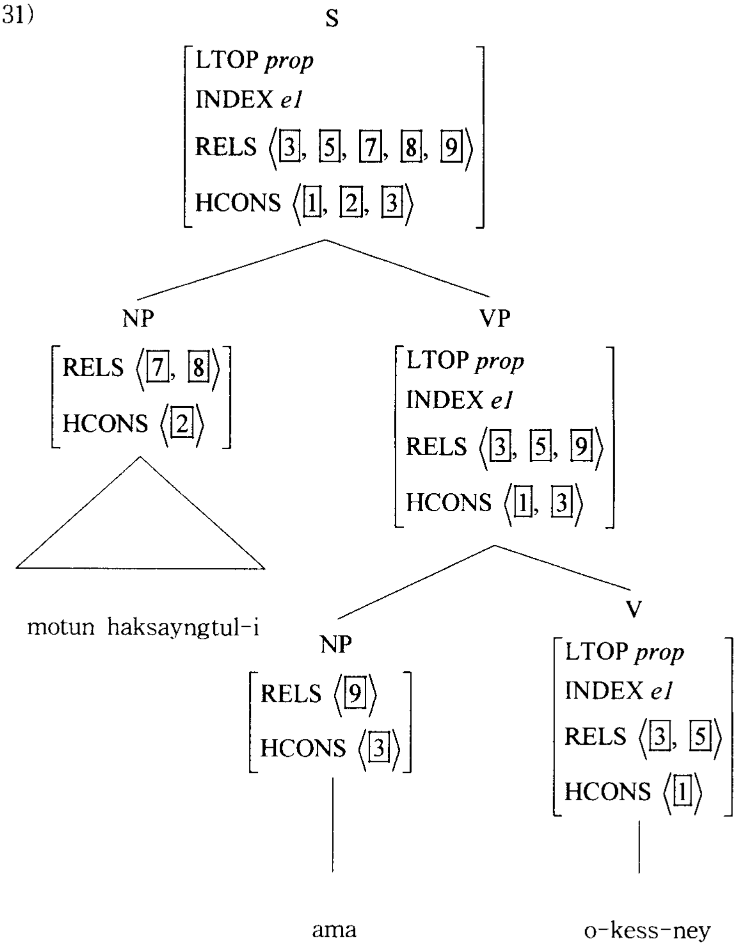
As noted here, the adverb modifies a verbal element whose INDEX is *e1* with the LTOP value *h8*. The scopal adverb *ama* will also induce a *qeq* relation in the HCONS. Given this lexical entry, the final MRS of the sentence will then look like the following:

(30)

```
[ HOOK [ LTOP h1
         INDEX e1 ]
  RELS ⟨ [3][ PRED _prpstn_m_rel
              LBL h1
              MARG h9 ],
         [5][ PRED _come_v_rel
              LBL h8
              ARG0 e1
              ARG1 x ],
         [6][ PRED _every_q_rel
              LBL h2
              ARG0 i
              RESTR h4
              BODY handle ],
         [7][ PRED _student_n_rel
              LBL h6
              ARG0 i ],
         [9][ PRED _probably_adv_rel
              LBL h7
              ARG1 h5 ] ⟩
  HCONS ⟨ [1][ qeq
               HARG h9
               LARG h8 ],
          [2][ qeq
               HARG h4
               LARG h6 ],
          [3][ qeq
               HARG h5
               LARG h8 ] ⟩ ]
```

There are several things to be noted here: The INDEX value of *every* and that of *students* are identical. The LARG of *probably* is coindexed with the LTOP of the VP it modifies. Meanwhile, the LARG of *every* is coindexed with the LTOP of the N0 *student* whereas its ARG0 is coindexed with its INDEX value. To clearly see how the coindexation is propagates through the structure, consider its structure:

(31)



Given the definitions of semantic composition in feature structures, the composition of such a sentence is straightforward: how the coindexation propagates through the structure. Both every and probably introduces a *qeq* relation. This relation is passed up through the tree by the virtue of the simple append of HCONS. The LAG of the *qeq* in ama 'probably' is coindexed with the LTOP of the structure it modifies.

3.2. Ambiguity from Two quantifiers

Then, consider the following:

(32) motun haksayng-tul-i sakwa-lul mek-ci anh-ass-ta
all student-PL-NOM apple-ACC eat-COMP NEG-PST-DECL

As noted earlier, such a sentence will have at least two readings with respect to the scope of the quantifier *motun* and the negator *anh-ass-ta*:

(33) a. It is not the case that all the students ate the apples.
b. As for all the students, they did not eat the apples (just part of the students ate apples).

As we have seen, the key point of capturing scope ambiguity in the MRS is to use the mechanism of underspecification without any additional mechanism. Consider the MRS of this sentence:

(34)

```
[ HOOK  [ LTOP h1
          INDEX e1 ]
  RELS  < [ PRED _prpstn_m_rel      [ PRED _eat_v_rel
            LBL h1                ,   LBL h5
            MARG h9 ]                 ARG0 e1          ,
                                      ARG1 i
                                      ARG2 j ]
          [ PRED _student_n_rel     [ PRED _all_q_rel
            LBL h4                ,   LBL h0
            ARG0 i ]                  ARG0 i           ,
                                      RESTR h3
                                      BODY h7 ]
          [ PRED _apple_n_rel       [ PRED _neg_rel
            LBL h2                ,   LBL h6
            ARG0 j ]                  ARG0 e1
                                      ARG1 h8 ]           >
  HCONS < [ qeq          [ qeq          [ qeq
            HARG h9   ,    HARG h3   ,    HARG h8
            LARG h6 ]      LARG h4 ]      LARG h5 ] > ]
```

As noted here, there is only one semantic representation that can induce two readings. The message value *prpstn_m_rel*, indicating the illocutionary force of the utterance, bears the top handle of the sentence as its LBL value and takes the label of 'eat' as its argument. It is also mediated by a *qeq* (equality modulo quantifiers) constraint in the HCONS list of handle constraints. The sentence introduces two scopal elements: *all* and *not*. Since the quantifier scope is underspecified, the BODY of all is left with unbound values. If *h7* is *h6* and *h8* is *h5*, *all* will have wider scope. Meanwhile, if *h7* is *h5* and *h8* is *h0*, the negator not then outscopes all. Such a mechanism of underspecification allows the MRS system to capture the semantic ambiguity without resorting to different syntactic structures at LF or additional semantic device such as Cooper's storage (cf. Copestake et al. 2003).

3.3. Control Predicates

As we noted before, we introduced the feature XARG (External argument) to pick out the index of the subject argument. This feature is of use in referring to the unexpressed subject in the various control environments. Consider one simple control example:

(35) haksayng-un sengkongha-lyeko noleykhayessta
student-NOM succeed-COMP tried
'The student tried to succeed.'

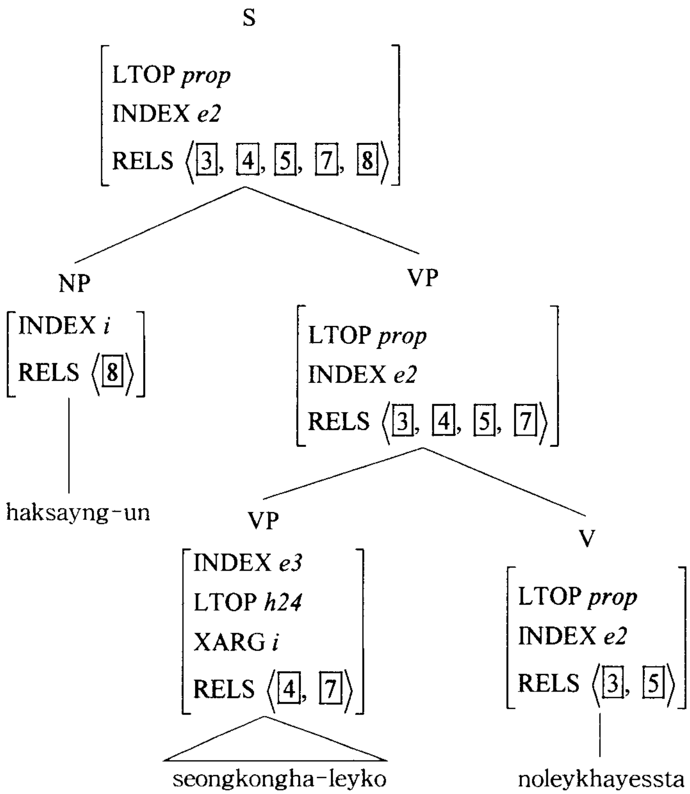
The lexical entry for the control verb try ensures that the ARG1 of *_try_v_rel* is identified with the external argument of the infinitival complement:

(36)

$$\begin{bmatrix} \text{ORTH}\langle \text{nolyekhayessta} \rangle \\ \text{SUBJ}\left\langle \begin{bmatrix} \text{HOOK.INDEX } \boxed{1} \end{bmatrix} \right\rangle \\ \text{COMPS}\left\langle \begin{bmatrix} \text{HOOK.XARG } \boxed{1} \end{bmatrix} \right\rangle \\ \text{SEM.RELS}\left\langle \begin{bmatrix} \text{ARG1 } \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix}$$

The VP complement of the verb *noleykhayessta* 'tried' identifies its semantic external argument (XARG) with the subject's semantic index value of the verb. This external value is also identified with the semantic role of the verb's first argument (ARG1). This lexical entry will project a structure like the following:

(37)



The construction of the MRS here follows the general compositional constraints. Syntactically, the head-complement rule unifies the verb *try* with its VP complement *sengkonghaleyko*. Semantically, this combination results in the identification of the VP's XARG value with the subject's INDEX value too. The head-subject rule will allow the resulting VP to combine with the subject *haksayng*. This unification also ensures the VP's XARG value is identified with the subject's INDEX value. The final MRS will be the one given in 38:

(38)

$$
\begin{bmatrix}
\text{HOOK} \begin{bmatrix} \text{LTOP } h1 \\ \text{INDEX } e2 \end{bmatrix} \\
\text{RELS} \left\langle
\begin{array}{l}
\boxed{3}\begin{bmatrix} \text{PRED } \_prpstn\_m\_rel \\ \text{LBL } h1 \\ \text{MARG } h4 \end{bmatrix}, \boxed{5}\begin{bmatrix} \text{PRED } \_try\_v\_rel \\ \text{LBL } h21 \\ \text{ARG0 } e2 \\ \text{ARG1 } i \\ \text{ARG2 } h22 \end{bmatrix}, \\
\boxed{8}\begin{bmatrix} \text{PRED } \_student\_n\_rel \\ \text{LBL } h4 \\ \text{ARG0 } i \end{bmatrix}, \boxed{7}\begin{bmatrix} \text{PRED } \_succeed\_v\_rel \\ \text{LBL } h24 \\ \text{ARG0 } e3 \\ \text{ARG1 } i \end{bmatrix}, \\
\boxed{4}\begin{bmatrix} \text{PRED } \_prpstn\_m\_rel \\ \text{LBL } h22 \\ \text{MARG0 } h23 \end{bmatrix}
\end{array}
\right\rangle \\
\text{HCONS} \left\langle \begin{bmatrix} qeq \\ \text{HARG } h4 \\ \text{LARG } h21 \end{bmatrix}, \begin{bmatrix} qeq \\ \text{HARG } h23 \\ \text{LARG } h24 \end{bmatrix} \right\rangle
\end{bmatrix}
$$

4. Conclusion

A computational grammar can be valuable only when they can assign correct semantic representations if it seeks applications that require natural language understanding (Copestake et al. 2001, Oepen et al. 2002). MRS, basically designed for computational semantics, produce a description of the meaning representation sufficient to support such aims.

This paper is an inceptive attempt to incorporate MRS for Korean and see the feasibility of implementing it into a computational Korean grammar. Though there remain issues of expanding this system into more complicated phenomena such as wh-questions and coordinations along with the expansion of grammar, we have seen that MRS is validate enough to build a semantically rich Korean grammar. Needless to say, the efficiency and validity of the MRS system for Korean needs to be validated by further applications and computational implementations.

References

Bender, Emily, and Dan Flickinger. (1999). *Peripheral constructions and core phenomena.* In G. Webelhuth, A. Kathol, and J.-P. Koenig (Eds.), *Lexical and Constructional Aspects of Linguistic Explanation.* Stanford: CSLI Publications.

Bender, Emily M., and Dan Flickenger, and Stephan Oepen. (2002). The Grammar Matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th Internation Conference on computational Linguistics*, 8-14, Taipei, Taiwan.

Carpenter, Bob. (1992). *The Logic of Typed Feature Structures.* Cambridge, UK: Cambridge University Press.

Copestake, Ann. (2002). *Implementing Typed Feature Structures* Stanford: CSLI Publications.

Copestake, Ann, Daniel P. Flickenger, Ivan A. Sag, and Carl Pollard. (2003). *Minimal Recursion Semantics. An introduction.* Unpublished ms.

Copestake, Ann, Alex Lascarides, and Dan Flickinger. (2001). An algebra for semantic construction in constraint-based grammars. In *Proceedings of the 39th Meeting of the Association for*

Computational Linguistics, Toulouse, France.

Flickinger, Dan. (2000). On building a more efficient grammar by exploiting types. Natural Language Engineering 6(1) (Special Issue on Efficient *Processing with HPSG*):15-28.

Flickinger, Dan, and Emily M. Bender. (2003). Compositional semantics in a multilingual grammar resource. In *Proceedings of the ESSLLI 2003 Workshop* "Ideas and Strategies for Multilingual Grammar Development", Vienna, Austria.

Gunji, Takao. (2005). *Measurement and Quantification Revisited.* Theoretical and Applied Linguistics at Kobe Shoin 8:21-36

Kim, Jong-Bok. (2004). *Korean Phrase Structure Grammar* (In Korean). Hankook Publishing.

Kim, Jong-Bok, Jaehyung Yang. (2004). Projections from Morphology to Syntax in the Korean Resource Grammar: Implementing Typed Feature Structures, *Lecture Notes in Computer Science,* Vol.2945, pp.13-24, Springer-Verlag, 2004.2.

Oepen, Stephan, Daniel Flickinger, J. Tsujii, and Hans Uszkoreit (Eds.). (2002). *Collaborative Language Engineering. A Case Study in Efficient Grammar-based Processing*. Stanford, CA: CSLI Publications.

Siegel, Melanie. (2000). *HPSG analysis of Japanese.* In Wolfgang Wahlster, editor, Vermobil: Foundations of Speech-to-Speech Translation. Springer, Berlin.

Siegel, Melanie, and Emily M. Bender. (2002). Ecient deep processing of Japanese. In *Proceedings of the 3rd Workshop on Asian Language Resources and Standardization at the 19th International Conference on Computational Linguistics*, Taipei, Taiwan.

Jong-Bok Kim
School of English
Kyung Hee University
Seoul 130-701, Korea
Phone: 82-2-961-0892
Email: jongbok@khu.ac.kr

Received: 30 Mar, 2006
Revised: 30 May, 2006
Accepted: 19 Jun, 2006