Geoffrey I. Webb
Xinghuo Yu (Eds.)

LNAI 3339

AI 2004: Advances in Artificial Intelligence

17th Australian Joint Conference on Artificial Intelligence
Cairns, Australia, December 2004
Proceedings

Springer

Feature Unification and Constraint Satisfaction in Parsing Korean Case Phenomena

Jong-Bok Kim[1], Jaehyung Yang[2], and Incheol Choi[3]

[1] School of English, Kyung Hee University, Seoul, Korea 130-701
[2] School of Computer Engineering, Kangnam University, Kyunggi, 449-702, Korea
[3] Language Research Institute, Kyung Hee University, Seoul, 130-701, Korea

Abstract. For a free-word order language such as Korean, case marking remains a central topic in generative grammar analyses for several reasons. Case plays a central role in argument licensing, in the signalling of grammatical functions, and has the potential to mark properties of information structure. In addition, case marking presents a theoretical test area for understanding the properties of the syntax-morphology interface. This is why it is no exaggeration to say that parsing Korean sentences starts from work on the case system of the language. This paper reports the project that develops a Korean Resource Grammar (KRG, Kim and Yang 2004), built upon the constrain-based mechanisms of feature unification and multiple inheritance type hierarchies as an extension of HPSG (Head-driven Phrase Structure Grammar), and shows that the results of its implementation in the Linguistic Knowledge Building System (cf. Copestake 2002) prove its empirical and theoretical efficiency in parsing case-related phenomena.

1 Formation of Case-Marked Elements

Nominal expressions allow various particles (including case markers) to be attached but in strict ordering relationships, as exemplified in the traditional template in (1)a and one example in (1)b (Kim 1998):

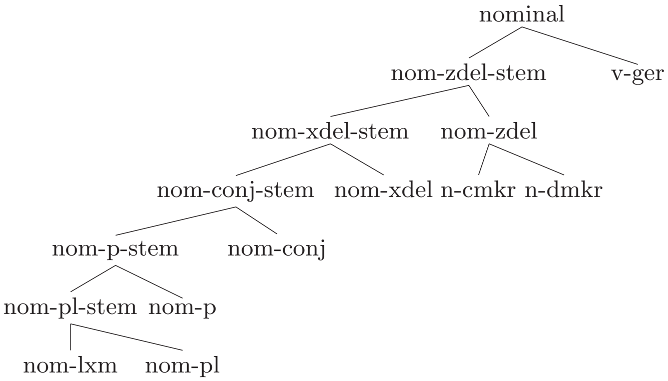
(1) a. N-base – (Hon) – (Pl) – (PostP) – (Conj) – (X-Delim) – (Z-Delim)
b. sensayng + (nim) + (tul) + (eykey) + (man) + (i)
teacher + Hon + Pl + Postp + only + NOM
'to the (honorable) teachers only'

As observed in (1)a, the GCASE markers such as NOM, ACC, and GEN can appear only in the final position, called Z-Delim(iter) position, whereas the SCASE markers (GOAL, LOC, etc) occupy the PostP position. We treat the particles as suffixes attached to the nominal stems in the lexicon by a step-by-step process based on the hierarchy in (2). The building process of nominal elements thus starts from the basic lexical elements of the type *nom-lxm* (nominal-lexeme), moving up to a higher type while any of these processes can be skipped and

G.I. Webb and Xinghuo Yu (Eds.): AI 2004, LNAI 3339, pp. 1160–1166, 2004.

then directly be realized as (pumped up to) a *word* element in syntax.[1] Thus the attachment of the plural suffix to the *nom-lxm* will generate *nom-pl*, and that of a postposition suffix will produce a *nom-p* element.

(2)



The constraints on each type place restrictions on the ordering relationship among nominal suffixes, as exemplified in (3):

(3) a. *nom-p* → [STEM *nom-pl-stem*]
b. *nom-zdel* → [STEM *nom-xdel-stem*]

These constraints mean that the type *nom-p* requires its STEM value to be a type of *nom-pl-stem*, and the type *nom-zdel* specifies its STEM value to be *nom-xdel-stem*. These constraints explain why (4)a is well-formed, but not (4)b:

(4) a. $[_{nom-p}$ $[_{nom-pl}$ sensayngnim-tul]-eykey] ‘teacher-PL-DAT’
b. *$[_{nom-p}$ $[_{nom-zdel}$ sensayngnim-nun]-eykey] ‘teacher-TOP-DAT’

The type *nom-pl* in (4)a is a subtype of *nom-pl-stem*, and this thus observes the constraint in (3)a. However, in (4)b, the type *nom-zdel* cannot serve as the STEM value of the postposition *-eykey* according to (3)a since it is not a subtype of *nom-pl-stem*.

2 Case Constraints in Syntax

Once we have the right generation of nominal elements with case information, the next issue is how argument-selecting heads and grammar rules contribute their case information to nominal elements. Phenomena such as case alternation illustrated in (5) make it hard to attribute to the case as lexical properties:

[1] This is one main difference from *verb-lxm*. As noted, only *v-free* elements can become a *v-word* that can appear in syntax.

(5) a. John-i nokcha-ka/*lul coh-ta
John-NOM green.tea-NOM/*ACC like-DECL
'John is fond of green tea.'

b. John-i nokcha-lul/*ka coh-a hanta
John-NOM green.tea-ACC/*NOM like-COMP do
'John likes green tea.'

Our analysis adopts the lexeme-based lexicon where all the verbal lexemes will minimally have the following information:

(6)
$$\begin{bmatrix} \textit{v-lxm} \\ \text{ORTH } \langle \text{ilk-} \rangle \\ \text{ARG-ST } \langle \text{NP}\begin{bmatrix}\text{GCASE } \textit{vcase}\end{bmatrix}, \text{NP}\begin{bmatrix}\text{GCASE } \textit{vcase}\end{bmatrix} \rangle \end{bmatrix}$$

This means that any element in the ARG-ST gets the value *vcase* as its GCASE value: the *vcase* value can be either *nom* or *acc* in syntax. The elements in the ARG-ST will, in accordance with a realization constraint, be realized as SUBJ and COMPS in syntax as indicated in the following:

(7)
$$\begin{bmatrix} \langle \text{ilk-ess-ta `read-PST-DECL'} \rangle \\ \text{SYN}\begin{bmatrix} \text{HEAD} \mid \text{POS } \textit{verb} \\ \text{VAL}\begin{bmatrix} \text{SUBJ } \langle \boxed{1} \rangle \\ \text{COMPS } \langle \boxed{2} \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST } \langle \boxed{1}\text{NP}\begin{bmatrix}\text{GCASE } \textit{vcase}\end{bmatrix}, \boxed{2}\text{NP}\begin{bmatrix}\text{GCASE } \textit{vcase}\end{bmatrix} \rangle \end{bmatrix}$$

With this declarative verb *ilk-ess-ta* 'read-PST-DECL', the SUBJ element can be *nom* whereas the COMPS can be *acc*, but not the other grammatical case value as noted in (8):

(8) John-i/*ul chayk-ul/*i ilk-ess-ta
John-NOM/ACC book-ACC/NOM read-PST-DECL
'John read a book.'

Then, the question is which part of the grammar makes sure the SUBJ is *nom* whereas COMPS is *acc*. The determination of case value in the VAL is not by a lexical process but imposed by syntactic rules. That is, we assume that Korean X$'$ syntax includes at least the Head-Subject Rule encoded in the LKB as the following feature description:

```
head-subj-rule := hd-arg-ph &
 [ SYN.VAL [ SUBJ <>,
             COMPS #2 ],
   ARGS < #1 & [ SYN.HEAD [ CASE.GCASE nom, PRD - ] ],
          [ SYN.VAL [ SUBJ < #1 >,
                      COMPS #2 ] ] > ].
```

The rule simply says that when a head combines with the SUBJ, the SUBJ element is *nom*. As for the case value of a complement, it is a little bit more complicated since there are cases where the nonsubject argument gets NOM rather than ACC as in (5). In the language, nonagentive verbs like *coh-* assign NOM to their complements. Reflecting this type of case assignment, we adopt the head feature AGT (AGENTIVITY) and ramify the Head-Complement Rule into two as the following:[2]

(9) a. Head-Complement Rule A:

$$\begin{bmatrix}\textit{hd-comp-ph}\end{bmatrix} \Rightarrow \boxed{1}\begin{bmatrix}\text{CASE}\,|\,\text{GCASE } \textit{acc}\end{bmatrix}, \mathbf{H}\begin{bmatrix}\text{HEAD}\,|\,\text{AGT} + \\ \text{COMPS}\left\langle ..., \boxed{1}, ...\right\rangle\end{bmatrix}$$

b. Head-Complement Rule B:

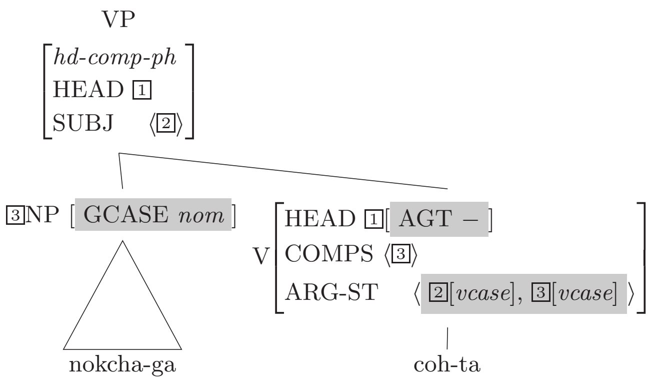
$$\begin{bmatrix}\textit{hd-comp-ph}\end{bmatrix} \Rightarrow \boxed{1}\begin{bmatrix}\text{CASE}\,|\,\text{GCASE } \textit{nom}\end{bmatrix}, \mathbf{H}\begin{bmatrix}\text{HEAD}\,|\,\text{AGT} - \\ \text{COMPS}\left\langle ..., \boxed{1}, ...\right\rangle\end{bmatrix}$$

Within this system, we then do not need to specify *nom* to the nonsubject complement of psych verbs, diverging from the traditional literature. Just like other verbs, the complement(s) of such psych verbs like *coh-ta* 'like-DECL' will bear just *vcase*, as a general constraint on verbal elements as represented in (10)a:

(10)
$$\begin{bmatrix}\text{HEAD}\begin{bmatrix}\text{POS } \textit{verb} \\ \text{AGT} -\end{bmatrix} \\ \text{ARG-ST}\left\langle \text{NP}\begin{bmatrix}\text{GCASE } \textit{vcase}\end{bmatrix}, \text{NP}\begin{bmatrix}\text{GCASE } \textit{vcase}\end{bmatrix}\right\rangle\end{bmatrix}$$

This lexical information would then project the following structure for (5):

(11)



[2] The positive value of the AGT (AGENTIVITY), similar to STATIVITY, is assigned to the verbs that have an external argument whereas the negative value is assigned to those with no external argument.

As noted here, the verb *coh-ta* ‘like’ bears the head feature [AGT −]. This means that the complement of this verb will get NOM even though in the ARG-ST its case value is *vcase*. This is guaranteed by the Head-Complement Rule B in (9).

3 Some Merits of the Feature Unification

3.1 Two Nominative Cases

One tricky case pattern in the language is the double occurrence of nominative markers:

(12) sensayngnim-kkeyse-man-i o-si-ess-ta
teacher-HON.NOM-only-NOM came
‘Only the honorable teacher came.’

The marker *-kkeyse* here functions as a honorific subject marker and falls the same morpholoigcal slot as the postposition marker. This marker cannot mark nominative objects or adjuncts: It marks only honorable nominative subjects. This implies that the stem produced by the attachment of *kkeyse* carries at least the following information:

(13)
$$\begin{bmatrix} \langle \text{sensayngnim-kkeyse `teacher-HON.NOM'} \rangle \\ \text{HEAD} \begin{bmatrix} \text{POS noun} \\ \text{HON} + \\ \text{CASE} \,|\, \text{GCASE } \textit{nom} \end{bmatrix} \end{bmatrix}$$

The [GCASE *nom*] value accounts for why this stem can combine only with the nominative marker. If we attach an accusative marker there will be a clash between [GCASE *acc*] and [GCASE *nom*]. This is not a possible feature unification:

(14)
$$* \begin{bmatrix} \langle \text{sayngkakha-kkeyse-man-ul `teacher-HON.NOM-DEL-ACC'} \rangle \\ \text{HEAD} \begin{bmatrix} \text{POS noun} \\ \text{HON} + \\ \text{CASE} \begin{bmatrix} \text{GCASE } \textit{nom} \\ \text{GCASE } \textit{acc} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

3.2 Case Omission and Delimiters

Another welcoming consequence of the present analysis in which the unification and subsumption operations of feature structures play key roles in the KRG comes from phenomena where case markers are not realized or replaced by delimiters. One main property of case markers is that they can be omitted or can be replaced by delimiters in proper context:

(15) haksayng-(tul) chayk-(to) ill-ess-e
student-PL book-even read
'Students even read a book.'

The basic lexical entries for the words in (15) would be something like the following:

(16)

a.
$$\begin{bmatrix} \langle \text{ilk-ess-e `read-PST-DECL'} \rangle \\ \text{HEAD} \mid \text{AGT} + \\ \text{ARG-ST } \langle \text{NP}\begin{bmatrix}\text{GCASE } \textit{vcase}\end{bmatrix}, \text{NP}\begin{bmatrix}\text{GCASE } \textit{vcase}\end{bmatrix}\rangle \end{bmatrix}$$

b.
$$\begin{bmatrix} \left\langle \text{haksayng-tul `student-PL'} \right\rangle \\ \text{HEAD}\begin{bmatrix}\text{POS } \textit{noun} \\ \text{CASE } \begin{bmatrix}\text{GCASE } \textit{gcase}\end{bmatrix}\end{bmatrix} \end{bmatrix}$$

c.
$$\begin{bmatrix} \left\langle \text{chayk-to `book-also'} \right\rangle \\ \text{HEAD}\begin{bmatrix}\text{POS } \textit{noun} \\ \text{CASE } \begin{bmatrix}\text{GCASE } \textit{gcase}\end{bmatrix}\end{bmatrix} \end{bmatrix}$$

Note that the nouns here, projected to NPs, are not specified with any grammatical case value even though they may have semantic information coming from the delimiters. The present analysis generates the structure (17) to the sentence (15). As represented in the tree structure, since *gcase* is supertypes of *nom* and *acc*, there is no unification failure between the case information on the lexical element and the case requirement imposed by the Head-Subject and Head-Complement Rule. For example, in accordance with the Head-Complement Rule A, the complement of the agentive head must be *acc*, but the complement itself bears *gcase.* Since *gcase* is the supertype of *acc*, there is no feature clash. The case hierarchy, together with the feature unification and subsumption, thus allows us to capture no realization of the case markers in a straightforward manner.

4 Testing the Feasibility of the System and Conclusion

The KRG we have built within the typed-feature structure system and well-defined constraints, eventually aiming at working with real-world data, has been first implemented into the LKB. In testing its performance and feasibility, we used the 231 (grammatical and ungrammatical) sentences from the literature and 292 sentences from the SERI Test Suites '97 (Sung and Jang 1997) designed to evaluate the performance of Korean syntactic parsers:

(17)

	# of Sentences	# of Words	# of Lexemes
SERI	292	1200	2679
Literature	231	1009	2168
Total	523	2209	4847

Of the 2209 words, the number of nominal elements is 1,342. These nominal elements include total 1,348 particles, which can be classified as follows:

(18)

	NOM	ACC	GEN	Delimiter	Semantic cases	Vocative	Total
Number	514	401	14	152	265	2	1,348

As the table shows, the system correctly generated all the GCASE or SCASE marked words as well as delimiter-marked elements in the literature and Test Suites. The KRG lexicon, build upon the type hierarchy with relevant constraints on each type, generate all these elements and the Case Constraints in syntax properly licensed these in the grammar. In terms of parsing sentences, the KRG correctly parsed 274 sentences out of 292 Seri Test Suites and 223 out of 231 literature sentences, failing 26 sentences (497 out of 523 sentences). Failed sentences are related to the grammar that the current system has not yet written. For example, the SERI Test Suites include examples representing phenomena such as honorification, coordination, and left dislocation of subject. It is believed that once we have a finer-grained grammar for these phenomena, the KRG will resolve these remaining sentences. Another promising indication of the test is that its mean parse (average number of parsed trees) for the parsed sentences marks 2.25, controlling spurious ambiguity at a minimum level.

As noted here, the test results provide clear evidence that the KRG, built upon typed feature structure system, offers high performance and can be extended to large scale of data. Since the test sentences here include most of the main issues in analyzing the Korean language, we believe that further tests for designated corpus will surely achieve nearly the same result of high performance too.

References

Copestake, Ann. 2002. *Implementing Typed Feature Structure Grammars.* CSLI Publications.

Kim, Jong-Bok. 1998. Interface between Morphology and Syntax: A Constraint-Based and Lexicalist Approach. *Language and Information* 2: 177-233.

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

Sung, Won-Kyung and Myung-Gil Jang. 1997. SERI Test Suites '95. In *Proceedings of the Conference on Hanguel and Korean Language Information Processing.*