Projections from Morphology to Syntax in the Korean Resource Grammar: Implementing Typed Feature Structures

Jong-Bok Kim[1] and Jaehyung Yang[2]

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

Abstract. Korean has a complex inflectional system, showing agglutinative morphology and using affixation as the major mechanism for word formation. A prerequisite to the successful development of any syntactic/semantic parsers for the language thus hinges on the efficient lexicon that can syntactically expand its lexical entries and map into syntax and semantics with robust parsing performance. This paper reports the system of the Korean Resource Grammar developed as an extension of HPSG (Head-driven Phrase Structure Grammar) and the results of implementing it into the Linguistic Knowledge Building (LKB) system (cf. Copestake 2002). The paper shows that the present grammar proves to be theoretically as well as computationally efficient enough in parsing Korean sentences.

1 Korean Resource Grammar

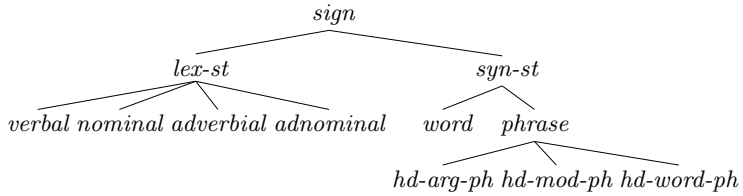
The Korean Resource Grammar (KRG) is a computational grammar for Korean currently under development since October 2002 (cf. Kim and Yang 2003). Its aim is to develop an open source grammar of Korean. The grammatical framework for the KRG is the constraint-based grammar, HPSG (cf. Sag, Wasow, and Bender 2003). HPSG (Head-driven Phrase Structure Grammar) is built upon a non-derivational, constraint-based, and surface-oriented grammatical architecture. HPSG seeks to model human languages as systems of constraints on typed feature structures. In particular, the grammar adopts the mechanism of type hierarchy in which every linguistic sign is typed with appropriate constraints and hierarchically organized. The characteristic of such typed feature structure formalisms facilitates the extension of grammar in a systematic and efficient way, resulting in linguistically precise and theoretically motivated descriptions of languages including Korean. The concept of hierarchical classification is essentially assigning linguistic entities such as phrases and words to specific types, and an assignment of those types to superordinate types. Each type is declared to obey certain constraints corresponding to properties shared by all members of that type. This system then allows us to express cross-classifying generalizations about phrases and words, while accommodating the idiosyncracies of individual types on particular subtypes of phrases or words.

As the basic tool for writing, testing, and processing the Korean Resource Grammar, we adopt the LKB (Linguistic Knowledge Building) system (Copes-

take 2002). The LKB system is a grammar and lexicon development environment for use with constraint-based linguistic formalisms such as HPSG.[3]

The Korean Resource Grammar (henceforth KRG) consists of grammar rules, inflection rules, lexical rules, type definitions, and lexicon. All the linguistic information is represented in terms of signs. These signs are classified into subtypes as represented in a simple hierarchy in (1):

(1)



The elements in *lex-st* type, forming the basic components of the lexicon, are built up from lexical processes such as lexical rules. Parts of these elements will be realized as *word* to function as a syntactic element. Phrases projected from *word* form basic Korean well-formed phrases such as *hd-arg-ph* (head-argument-ph) and *hd-mod-ph* (head-modifier-ph). In what follows, we will discuss how such projections are possible within a type-featured system.

2 Building the Lexicon through a Templatic Approach

The verb in Korean cannot be an independent word without inflectional suffixes. The suffixes cannot be attached arbitrarily to a stem or word, but need to observe a regular fixed order. Reflecting this, the verbal morphology has traditionally been assumed to be templatic. The template in (2) is a simplified one for the verbal suffixes in Korean, assumed in Cho and Sells (1994), among others.

(2) V-base + (Passive/Causative) + (Honorific) + (Tense) + Mood + (Comp)

As can be seen from the above template, verb suffixes, attaching to the preceding verb stem or word, mark honorific, tense, and mood functions. Morphologically, the inflectional suffixes preceding Mood are optional, but a Mood suffix obligatorily needs to be attached to a verb stem in simple sentences. Thus the verbal stem and the mood suffix are mutually bound in the sense that the bare verb stem cannot be used uninflected in any syntactic context and it should be inflected at least with the mood suffix, as seen in (3).

(3) a. ilk-(ess)-ta 'read-(Past)-Decl'
b. *ilk-ess 'read-Past'

Also, as expected from the template, the verbal suffixes observe the rigid ordering restrictions: the template ordering cannot be violated.

(4) a. *cap-ass-si-ta 'catch-Past-Hon-Decl'
b. *cap-ta-ass 'catch-Decl-Past'

[3] The LKB is freely available with open source (http://ling.stanford.edu).

The template given in (2) appears to capture the ordering generalizations as well as combinatory possibilities of verbal suffixes. However, the template alone could generate some ill-formed combinations, as given in (5).

(5) a. ka-(*si)-(*ess)-ca 'go-Hon-Past-Prop'
b. ka-(*si)-(*ess)-la 'go-Hon-Past-Imper'

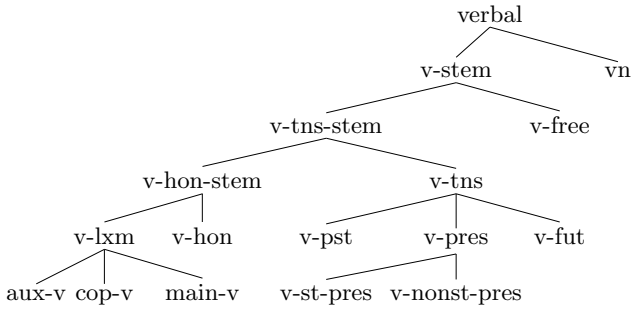
If we simply assume the template in (2) with the given suffixes in each slot, we would allow the ill-formed combinations here. The propositive mood suffix *-ca* and imperative mood suffix *-la* cannot combine either with the honorific suffix or with the tense suffix. They can combine only with a verb root as in *ka-ca* 'go-Prop' and *ka-la* 'go-Imper'. This means that verbal suffixes like *-ca* and *-la* have their own selectional or co-occurrence restrictions in addition to their being positioned into the Mood slot. The template alone thus fails to describe all the combinatory possibilities, demanding additional mechanisms. In addition, taking into consideration other types of verbal elements such as complementizer words or subordinator words, more templates are called upon. Leaving aside the issue of empty elements when optional suffixes are not realized, a templatic approach appears not to reflect the morphological structure of Korean inflections (cf. Kim 1998).

3 A Type-hierarchy Approach

3.1 Verbal Morphology

The starting point of structuring the lexicon is parts of speech in the language. Like the traditional literature, the KRG assumes *verbal*, *nominal*, *adverbial*, and *adnominal* as the language's basic categories. These are further subclassified into subtypes. For example, the type *verbal* is taken to have the hierarchy in (6):

(6)



Such a classification aims to capture the basic verbal morphology of Korean. In turn, it means a verbal element will be built up step by step as in the following:

(7) a. [[[[[(cap) + ((hi))]+((si))] +((ess))]+ (ta)] + (ko)] 'catch-CAUS-HON-TNS-DECL-COMP'
b. v-lxm → v-hon (v-hon-stem) → v-tns (v-tns-stem) → v-free (v-stem) → v-comp

Such building processes are constrained by the type declarations, some of which are given in (8):[4]

(8)

a. v-hon: $\begin{bmatrix}\text{ORTH } \boxed{1} + \text{si} \\ \text{STEM } \begin{bmatrix}\textit{v-base} \\ \text{ORTH } \boxed{1}\end{bmatrix} \\ \text{SYN.HEAD.HON } +\end{bmatrix}$ b. v-tns-stem: $\begin{bmatrix}\text{STEM } \begin{bmatrix}\textit{v-hon-stem} \\ \text{SYN } \boxed{1} \\ \text{SEM.RELS } \boxed{A}\end{bmatrix} \\ \text{SYN } \boxed{1} \\ \text{SEM.RELS } \boxed{A} \oplus \boxed{B}\end{bmatrix}$

c. v-free: $\left[\text{STEM } \textit{v-tns-sten} \text{ SYN.HEAD.IC } \textit{bool}\right]$

The constraints in (8)a mean that the type *v-hon* will take *v-base* as its stem; those in (8b) mean that the type *v-tns-stem* will take an instance of *v-hon-stem* as its stem. One thing to note here is that any subtypes of *v-hon-stem* can serve as the stem of *v-tns-stem* in accordance with the type hierarchy system. The grammar makes only the instances of *v-free* serve as an input to syntax.

These constraints restrict the possible word internal structures in Korean word formation. The system could provide a clean account for the ill-formed combinations without employing mechanisms such as templates. Observe the following:

(9) a. $*_{v-hon-stem}[_{v-tns-stem}$[cap-ass]-si]-ta 'catch-Past-Hon-Decl'
b. $*_{v-free}[_{v-hon-stem}$[cap-usi]-ta]-ess 'catch-Hon-Decl-Pas'
c. $*[_{v-hon-stem}[_{v-hon-stem}$[cap-usi]-usi]-ess]-ta 'catch-Hon-Hon-Past-Decl'

(9a) is ruled out because the honorific suffix co-occurs with the *v-tns-stem*, violating (8a); (9b) is ill-formed since the passive suffix *-ess* is attached to the *v-free* stem. This violates (8b) that requires its stem value be *v-hon-stem* or any of its subtypes. In the same vein, (9c) is not generated because the second honorific suffix occurs not with a *v-base*, but with a *v-hon* stem.

One important question arises: why do we need the notions of types in the morphological theory? The reason is simply that any morphological theory for Korean needs certain notions similar to types. We can find cases where we should have some notions referring to a specific group of morphological objects, so as to predict that a certain morphological phenomena applies only to this group. As noted, only instances of *v-free* can be pumped up to *v-word* occurring in syntax.[5] Being its subtype, *v-sug-infm* (v-suggestive-informal) requires its STEM value to be *v-base* as represented in the following:

(10) a. $*_{v-hon-stem}$[ilk-usi]-ca 'read-Hon-Sug'
b. $_{v-base}$[ilk]-ca 'read-Sug'

In a template analysis like (2), this would mean the honorific and tense slots should be empty. This would surely make the grammar much complicated. However, the present type-based system can efficiently avoid such an issue by simply referring to the type *v-base* as the STEM value of the type *v-sug-infm*.

[4] The implemented feature descriptions in the LKB system are slightly different from those represented here.

[5] The type *v-free* is further subtyped into *v-ind*, *v-dep*, and *v-ger*. Each of these functions as an independent syntactic element. *v-ind* functions as a predicate in the independent clause, *v-dep* words are used as dependent verbs such as complementizer or subordinator predicates.

3.2 Nominal Morphology

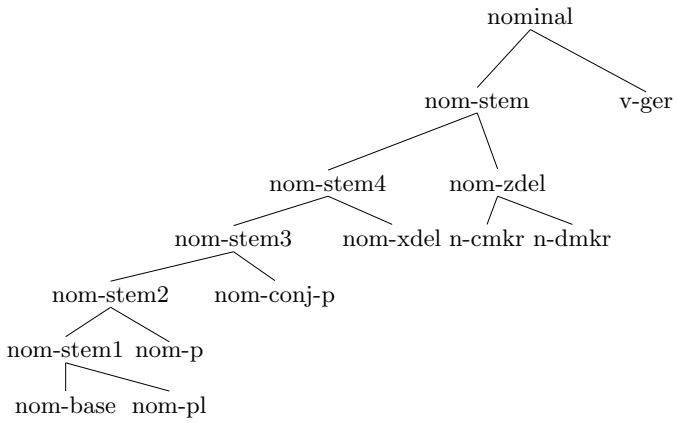
Nominal inflection is basically different from verbal inflection. Even though like verbal inflections, nominal suffixes are also under tight ordering restrictions, all the nominal suffixes are optional as represented in the following template and a true example:

(11) N-base – (Hon) – (Plural) – (Postp) – (Conj) – (X-Delim) – (Z-Delim)

(12) sensayng + (nim) + (tul) + (eykey) + (man) + (un)
teacher + Hon + Pl + Postp + X-Delim + Z-Delim
'to the (honorable) teachers only'

All the suffixes (often called particles) here, decoding various grammatical functions, need not be realized. Traditionally particles are treated as independent words even though they act more like verbal suffixes in terms of strict ordering restrictions, no intervention by any word element, and so forth. Our grammar, following lexicalist perspectives (cf. Cho and Sells 1994, Kim 1996), takes a quite different approach: we take particles not to exist as independent words but to function as optional inflectional suffixes. As a starting point, the KRG sets up different types of nominals corresponding to meaningful classes as represented in the hierarchy: (13):

(13)



The building up process of nominal elements starts from the type *nom-base* that includes subtypes such as *vn, n-bn, n-cn, n-cl, n-prop* (verbal nouns, bound nouns, common nouns, classifiers, proper nouns). Just like the process of building verbal elements, nominal word formation observes this hierarchical process:

(14) nom-base → nom-stem1 → nom-stem2 → nom-stem3 → nom-stem4 → nom-stem

One crucial difference from the forming process of verbal elements is that any of these processes can directly be realized as (pumped up to) a *word* element in syn-

tax.[6] The constraints on each type place restrictions on the ordering relationship among nominal suffixes:

(15)

nom-pl: $\begin{bmatrix}\text{ORTH } \boxed{1} + \text{tul} \\ \text{STEM } \begin{bmatrix}\textit{nom-base} \\ \text{ORTH } \boxed{1}\end{bmatrix}\end{bmatrix}$ nom-conj-p: $\begin{bmatrix}\text{STEM}\begin{bmatrix}\textit{nom-stem2} \\ \text{SEM}\begin{bmatrix}\text{INDEX } \boxed{1} \\ \text{RELS } \boxed{A}\end{bmatrix}\end{bmatrix} \\ \text{SEM}\begin{bmatrix}\text{INDEX } \boxed{1} \\ \text{RELS } \boxed{A}\end{bmatrix}\end{bmatrix}$

nom-p: $\begin{bmatrix}\text{STEM } \textit{nom-stem1} \\ \text{SYN.HEAD.CASE pcase}\end{bmatrix}$ nom-zdel: $\begin{bmatrix}\text{STEM } \textit{nom-stem4}\end{bmatrix}$

These constraints on the nominal types can place ordering restrictions among nominal particles:

(16) a. *[$_{nom-stem}$[sensayngnim-tul-un]-eykey] 'teacher-PL-DEL-POST'
b. *[$_{nom-stem3}$[sensayngnim-tul-kwa]-eykey] 'teacher-PL-CONJ-POST'
c. *[$_{nom-stem}$[sensayngnim-tul-un]-i] 'teacher-PL-DEL-NOM'

The so-called postposition *eykey* requires its STEM value to be an instance of *nom-stem*. This explains why (16a) and (16b) are not generated in the system. The nominative marker can combine only with *nom-stem4* or its subtypes. This explains why the system generates cases like (16c). However, it correctly generates cases like the following:

(17) a. $_{nom-base}$[sensayngnim]-i 'teacher-NOM'
b. $_{nom-stem1}$[$_{nom-base}$[sensayngnim]-tul]-kwa 'teacher-CONJ'

As noted, the type hierarchy system allows the STEM value to be any subtypes of the originally required one. For example, even though the case marked nominal (nom-cmkr) element would have its STEM value *nom-stem4*, *nom-base* can also satisfy this satisfaction since it is a subtype of *nom-stem4*.

In sum, the morphological system we have shown makes the Korean morphology much simpler and can capture the ordering restrictions as well as cooccurrence restrictions. Other welcoming consequences of adopting the typed feature system come from the treatment of well-known mixed constructions such as sentential nominal and light verb constructions. Both of these have received much attention because their mixed properties.

4 Multiple Inheritance Hierarchy: Advantages

One main property of the typed feature system we developed here is that it allows to adopt multiple inheritance hierarchies, commonly used in the object-oriented programming paradigm to organize multiple dimensions of information about

[6] The grammar specifies only *v-free* to be realized as *v-word* whereas for nouns it permits all the instances of type *nominal* to be realized as *n-word*. This in turn means any subtype of *nominal* can serve as a syntactic element in accordance of the type hierarchy in (13).

objects in particular knowledge domains. In particular, this multiple inheritance system provides a straightforward and efficient method of capturing the mixed properties of phenomena such as light verb and nominalizations constructions, both of which are most common phenomena and notorious for their syntactic complexities.

4.1 Nominalization

One of the main puzzles in the treatment of Korean sentential nominalizations or verbal gerundive phrases (VGP) is that they display verbal properties internally and nominal properties externally. Internal verbal properties are prevalent. One telling piece of evidence comes from the inheritance of arguments from the lexeme verb from which the gerundive verb is derived. As shown in (18), the gerundive verb takes the same arguments, the nominative subject and accusative object:

(18) [John-i ecey ku chayk-ul/*uy
John-NOM yesterday that book-ACC/*GEN
ilk-ess-um]-i myonghwak-hata
read-PAST-NMLZ-NOM clear-do
'John's having read the book yesterday is clear'

Various other phenomena also show that such gerundive phrases are internally similar to VPs. They can include a sentential adverb as in (19a); an adverbial element can modify the gerundive verb as in (19b); the phrase can include the sentential negation marker *an* as in (19c); it also can contain the full range of auxiliaries as in (19d), the phrase allows free scrambling of its elements as in (19e):

(19) a. John-i **papokathi** ku chayk-ul ilk-ess-um (Sent. Adv)
John-NOM foolish that book-ACC read-PAST-Nmlz
'John's having read the book foolish'

b. John-i chayk-ul **ppalli/*ppalun** ilk-um (Adv Mod)
John-NOM book-ACC fast(adv)/*fast(adj) read-Nmlz
'John's reading books fast.'

c. John-i chayk-ul **an** ilk-um (Sentential Neg)
John-NOM book-ACC NEG read-Nmlz
'John's not reading books.'

d. John-i chayk-ul ilk-ko **siph-um** (Aux verb)
John-NOM book-ACC read-COMP want-Nmlz
'John's wanting to read books'

e. **ku chayk-ul** John-i __ ilk-ess-um-(i nollapta) (Scrambling)
book-ACC John-NOM __ read-PST-NM-NOM surprising
'It is surprising that John read the book.'

Meanwhile, its external structure is more like that of NPs. VGPs can appear in the canonical NP positions such as subject or object as in (20a) or as a postpositional object in (20b).

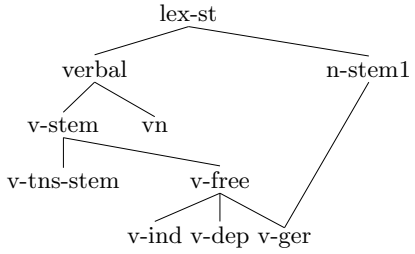
(20) a. [ai-ka chayk-ul ilk-um]-i nollapta
child-NOM book-ACC read-NMLZ-NOM surprising
'That child's reading a book is surprising'

b. [John-i enehak-ul kongpwuha-m]-**eytayhay** mollassta
John-NOM linguistics-ACC study-Nmlz-about not.know
'(We) didn't know about John's studying linguistics.'

These mixed properties of Korean sentential nominalization have provided a challenge to syntactic analyses with a strict version of X-bar theory. Various approaches (see Malouf 1998 and references cited therein) have been proposed to solve this puzzle, but they all have ended up abandoning or modifying fundamental theoretical conditions such as endocentricity, lexicalism, and null licensing.

In the KRG with the multiple inheritance mechanism, the type *v-ger* is classified as the subtype of both *v-free* and *n-stem1* as represented in the following hierarchy:[7]

(21)



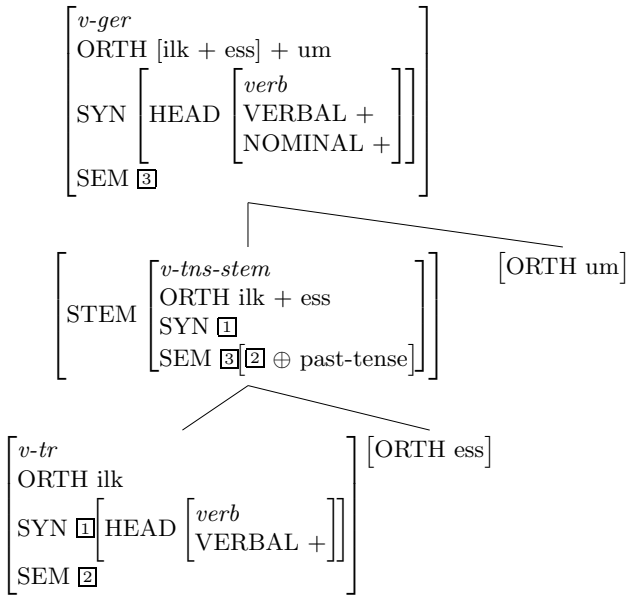
Such a cross-classification, allowing multiple inheritance, is also reflected in the feature descriptions in the LKB. The following represents a sample source cord:

```
v-ger := v-free & n-stem1 &
 [ SYN #syn & [ HEAD.MOD <> ],
   SEM #sem,
   ARGS < v-tns-stem & [ SYN #syn, SEM #sem ] > ].
```

As observed here, being a subtype of *v-free* and *n-stem1* implies that *v-ger* will inherit their properties. Since it is a subtype of *v-free*, *v-ger* will act just like as a verb: selecting arguments and assigning case values to them. In addition, *v-ger* can undergo the same nominal suffixation process since it is a subtype of *n-stem1*. For example, the gerundive verb *ilk-ess-um* will be generated through the following informally represented structure in the KRG.

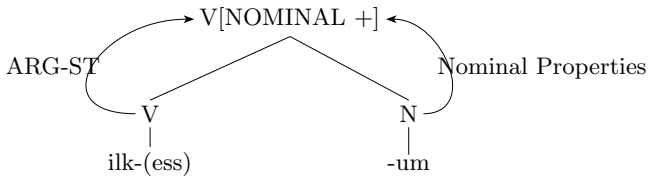
[7] In capturing the mixed properties, the KRG system adopts the binary-valued features VERBAL and NOMINAL. Nominalized verbs are assigned to have [VERBAL +] and [NOMINAL +] with the HEAD value *verb*. Meanwhile, the verbal nouns are different form nominalized verbs with respect to the HEAD value: They are *noun*.

(22)



The gerundive verb starts from a transitive lexeme *ilk* 'read' and forms a *v-tns-stem* after the attachment of the past tense suffix *ess*. When this *v-tns-stem* is attached with the nominalizer suffix, it inherits [NOMINAL +] feature. As such, various verbal properties are inherited from *v-tran-lxm* whereas the nominal properties coming when it attaches to the nominalizer. This is a reflection of how information flow occurs in sentential nominalization:

(23)



As can be seen in (23), the ARG-ST information is coming from the left element since the nominalized N still needs to combine with the complement(s) of the verb, while the categorial information comes from the righthand nominalizer.

Such a treatment is a clear advantage over previous theoretical or computational approaches in which nominalized verbs are simply taken to be either verbs or nouns. If they are taken to be verbs, ad hoc mechanisms are required to generate nominal suffixed nominalized verbs, causing heavy parsing loads. If they are simply taken to be nouns, we could not account for why gerundive verbs can be also inflected with tense and honorific and function just like verbs. The multiple inheritance system, designed with fine-grained feature declarations, can avoid such an issue.

4.2 Light Verb Constructions

As the name implies, VNs (verbal nouns) in Korean also display both nominal and verbal properties. The case markings on the VNs and the genitive case marking on its argument indicate that they have nominal properties:

(24) John-i mullihak-uy yonkwu-lul hayessta
John-NOM physics-GEN study-ACC did
'John studied physics.'

They also have verbal properties in the sense that they select arguments and assign case markings on its arguments independently.

(25) a. John-i mwullihak-ul yonkwu (cwung)
John-NOM physics-ACC study (while)
'John is in the middle of studying physics.'
b. John-i ku ceyphwum-ul mikwuk-eye yelshimhi swuchwul-ul hayessta
John-NOM the item-ACC US-LOC diligently export-ACC did
'John diligently exported the item to US.'

Just like the treatment of gerundive verbs, the multiple inheritance mechanism plays an important role in capturing the mixed properties. In the KRG, verbal nouns are also cross-classified as a subtype of both *n-base* and *verbal*.

```
vn := n-base & verbal &
 [ SYN.HEAD.TYPE t-none,
   SEM [ MODE statement,
         INDEX event ] ].
```

This feature description implies that *vn*, being a subtype of *n-base* and *verbal*, will inherit their properties. For example, the structure of the VN *swuchwul* 'export' would be something like the following:

(26)
$$\begin{bmatrix} \textit{nom-zdel} \\ \text{ORTH swuchwul-ul} \\ \text{STEM} \begin{bmatrix} \textit{vn} \\ \text{ORTH swuchwul} \\ \text{SYN } \boxed{1} \begin{bmatrix} \text{HEAD} \begin{bmatrix} \text{noun} \\ \text{NOMINAL } + \\ \text{VERBAL } + \end{bmatrix} \\ \text{ARG-ST } \langle \text{NP[nom], NP[acc], NP[dat]} \rangle \end{bmatrix} \end{bmatrix} \\ \text{SYN } \boxed{1} \begin{bmatrix} \text{HEAD } \begin{bmatrix} \text{CASE acc} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

As a subtype of *n-base*, the HEAD feature of the VN will be *noun* and [NOMINAL +], and as a subtype of *verbal*, it will also inherit [VERBAL +] feature and ARG-ST value. This then would allow the VN to appear in any nominal position while internally acting like a verbal element.

5 Testing the Feasibility of the System

The grammar we have built within the typed-feature structure system here, eventually aiming at working with real-world data, has been first implemented into the LKB.[8] In testing its performance and feasibility, we used the SERI Test Suites '97 after the successful parsing of the self-designed 250 sentences. The SERI Test Suites (Sung and Jang 1997), carefully designed to evaluate the performance of Korean syntactic parsers, consists of total 472 sentences (292 test sentences representing the core phenomena of the language and 180 sentences representing different types of predicate). In terms of lexical entries, it has total 440 lexemes (269 nouns, 125 predicates, 35 adverbs, and 11 determiners) and total 1937 word occurrences. As represented in the following table, the testing results of the KRG prove quite robust:

(27)

	# of Lexemes	# of Words	# of Sentences
SERI	440	1937	472
KRG Parsing Results	440	1937	423
Coverage (%)	100	100	89.5

As the table shows, the system correctly generated all the lexemes in the test suites and inflected words. In terms of parsing sentences, the grammar parsed 423 sentences out of total 472. Failed 49 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 423 parsed sentences marks 1.67, 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 suites 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.

[8] The space does not allow us to explicate the morphological and semantic system of the KRG in Korean. As for morphology, we integrated MACH (Morphological Analyzer for Contemporary Hangul) developed by Shim and Yang (2002). This system segments words into sequences of morphemes with POS tags and morphological information.

As for semantics, we adopted the Minimal Recursion Semantics developed by Copestake et al. (2001). In the multilingual context in which this grammar has been developed, a high premium is placed on parallel and consistent semantic representations between grammars for different languages. Ensuring this parallelism enables the reuse of the same downstream technology, no matter which langauge is used as input. The MRS well suits for this purpose.

6 Conclusion

It is hard to deny the fact that in building up an efficient grammar, expressive accuracy has often been scarified in order to achieve computational tractability (Oepen et al. 2002). However, putting linguistic generalizations aside has brought difficulties expanding the coverage and eventually building up a large scale of grammar. To build up any efficient parsing system for languages like Korean which displays an intriguing morphological properties, a prerequisite is a system that can build up morphological elements in a systematic way and project them into syntax and semantics to achieve proper grammatical compatibility. Conventional forms of standard morphological representations have proved problematic, neither being able to capture linguistic generalizations nor pinning down descriptive adequacy. In contrast, the morphological and syntactic system we have developed here with typed feature structures solve such preexisting problems while keeping linguistic insights, thus making the Korean morphology much simpler (e.g., in capturing the ordering restrictions as well as co-occurrence restrictions). Other welcoming consequences of the present system come from the treatment of well-known mixed constructions such as sentential nominal and light verb constructions. Both of these have received much attention because their mixed properties and even have been impediments to theoretical as well as computational linguistics. We have seen that once we have a rigorously defined type feature structure system of grammar, all these fall out naturally with high efficient parsing performance.

References

Cho, Young-Mee Yu, and Peter Sells. 1995. A lexical account of inflectional suffixes in Korean. *Journal of East Asian Linguistics*4, 119-174.

Copestake, Ann, Dan Flickinger, Ivan Sag and Carl Pollard. 2001. Minimal Recursion Semantics: An introduction. Ms. Stanford University.

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 Jahyung Yang. 2003. Korean Phrase Structure Grammar and Implementing it into the LKB (Linguistic Knowledge Building) System (In Korean). *Korean Linguistics* 21: 1–41.

Malouf, Robert. 1998. *Mixed Categories in the Hierarchical Lexicon*. Stanford: CSLI Publications.

Oepen, Stephan, Dan Flickinger, Jun-ichi Tusujii, and Hans Uszkoreit. 2002. *Collaborative Language Engineering.* Stanford: CSLI Publications.

Sag, Ivan, Tom Wasow, and Emily Bender. 2003. *Syntactic Theory: A Formal Approach.* Stanford: CSLI Publications.

Shim, Kwangseob and Yang Jaehyung. 2002. MACH: A Supersonic Korean Morphological Analyzer. In *Proceedings of Coling-2002 International Conference*, pp.939-45, Taipei.

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