

## ALE를 이용한 한국어 문법의 설계

최운호    장석진  
서울대학교 언어학과  
서울시 관악구 신림동 산 56-1  
우: 151-742  
whchoi@clepsi.co.kr    sjchang@plaza.snu.ac.kr

### A Design for Korean Phrase Structure Grammar(KPSG) in ALE

Choi, Woon-Ho    Chang, Suk-Jin  
Department of Linguistics  
Seoul National University

#### 요약

본 논문에서는 한국어의 전산처리를 위한 문법 모형 개발의 일부분으로 HPSG에 기반한 문법 모형의 개발을 시도한다. 문법 모형의 개발에는 ALE(Attribute Logic Engine)를 이용하여, 보문 구조와 보조 용언 구문을 분석하기 위한 사전구조 및 문법 규칙을 제시한다. 그리고 문의 종류(Sentence Type:ST)와 문계(Sentence Level:SL), 시제, 존대 등을 분석해서 표상하기 위한 유형 계층 및 어휘부, 문법 규칙, 문법 원리 등을 제시한다.

#### 1 들어가기

한국어의 전산처리를 위한 문법 모형에 대한 연구의 일부분으로, 본 논문은 ALE(Attribute Logic Engine)를 이용하여 한국어 문법을 설계하고, 한국어 문법과 사전 구조의 설계 및 응용 가능성을 모색하는 것을 목적으로 한다.

ALE는 Bob Carpenter와 Gerald Penn에 의해서 자질 구조(Typed Feature Structure) 논리를 바탕으로 Prolog를 이용해 구현된 논리 프로그램 엔진으로, 문법의 설계가 용이하고, 문법의 적형성 및 계산가능성 검사를 문법 개발과 동시에 진행할 수 있는 장점이 있다.

ALE를 이용하여 설계된 한국어 문법(이하 KPSG)은 유형 계층 정의(Type Hierarchy Definition), 어휘 항목, 통사 규칙, 문법 원리, 형태-음운 규칙으로 구성되어 있다. 현재 KPSG는 몇 가지 실험용 문장을 바탕으로 응용 가능한 문법 구조의 원형을 설계하는 것을 목적으로 하기 때문에, 분석 가능한 문장에 제한을 두고 있다. 이 부분은 문법 및 어휘부의 확장을 통해서 차차

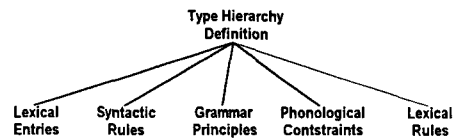
늘려가야 한다.

본 논문에서는, 보조 용언 구문, 원형 내포문 구성의 처리를 중심으로 ALE를 이용해 설계된 한국어 문법 구조를 제시한다.

#### 2 KPSG의 구조

KPSG는 기호(Sign)에 대한 유형 계층 정의를 바탕으로 어휘 항목 및 통사 규칙, 문법 원리를 구성하고, 일부 형태-음운 제약을 사용한다. KPSG의 기본 구조는 다음 (1)과 같다.

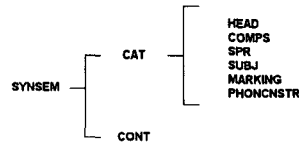
##### (1) KPSG의 구조



##### 2.1 유형 계층(Type Hierarchy) 정의

현재 KPSG의 유형 계층에서는 [SIGN:SYNSEM]을 최상위 자질로 설정했고, CAT(egory), CONT(ent) 자질이 설정되어 있다.

##### (2) KPSG의 유형 계층 정의



1) 기본적으로 KPSG는 [1]에 바탕을 두고 있지만, 문법 구조는 [1]에서 제시된 한국어 문법의 하위 부분이다.

(제 10회 한글 및 한국어 정보처리 학술대회)

CAT 자질에 의해 도입되는 HEAD 자질은 HEAD 유형을 값으로 가지며, HEAD 유형은 문법 기능을 담당하는 어휘(FUNC)와 실질어(SUBST)로 갈라진다. 현재까지 SUBST는 하위 유형으로 VERB(용언)와 NOUN(체언)을 가지며, VERB는 동사, 형용사, 지정사로 세분된다. FUNC는 현재 MARK 값만이 할당되어 있으며, MARK는 문법 기능을 담당하는 표지로 조사(격조사), 보문소 {고}, 보조용언 연결 어미 {고} 등이 할당되어 있다. (2)의 계층 구조는 [1]에서 제시된 화맥(Context) 등에 대한 구조가 아직 도입되지 않았다.

KPSG에서 CAT의 유형 계층 정의는 다음과 같다.

(3) CAT 유형 정의

```
cat sub [ ]
    intro [head:head,
           comps:list,
           spr:list,
           subj:list,
           marking:marking,
           phoncnstr:phsyltype].
```

2.2 통사 규칙

KPSG에서 현재 정의되어 있는 통사 규칙은 Complement-Head Rule, Head-Marker Rule 2가지이다.

Complement-Head Rule은 머리어와 머리어의 보충어를 결합시켜 주는 규칙이다.

(4) Complement-Head Rule

```
complement_head rule
(Mother, synsem:cat:comps:[])
==>
goal> comp_num_rstr(Comps),
cats> (Comps, ne_list),
cat> (HeadDtr, synsem:cat:comps:Comps),
goal> head_feature_principle(Mother, HeadDtr),
goal> marking_principle(Mother, HeadDtr),
goal> semantics_principle(Mother, HeadDtr).
```

(4)에서 (Mother, synsem:cat:comps:[])은 위 규칙이 적용되는 경우 Mother Node의 보충어 리스트 (comps-list)는 비어 있어야 한다는 것을 말한다. cats> (**Comps**, ne\_list)는 보충어 리스트를 말하며, cat> (HeadDtr, synsem:cat:comps:**Comps**)는 머리어 범주이다. 그리고, 보충어 리스트 Comps와 머리어의 보충어 자질 comps의 값인 Comps는 동일해야 한다.

(4)와 같이 단순한 통사 규칙 만으로는 과생성

의 우려가 있다. 이러한 과생성을 막아 주는 장치가 문법 원리(Grammar Principle)라고 하는 제약이다.

Head-Marker Rule은 명사구와 조사의 결합, 보문과 보문소의 결합 등에 적용되는 규칙이다.

(5) Head-Marker Rule

```
head_marker rule
(Mother, synsem:cat:(spr:[], comps:[]))
==>
cat> (HeadDtr,synsem:cat:(spr:[MarkerDtr],
                           comps:[])),
cat> (MarkerDtr, synsem:cat:(spr:[], comps:[])),
goal> phon_constraint(HeadDtr, MarkerDtr),
goal> spec_principle(HeadDtr, MarkerDtr),
goal> head_feature_principle(Mother,HeadDtr),
goal> marking_principle(Mother, MarkerDtr),
goal> semantics_principle(Mother,HeadDtr).
```

(5)에서 Mother Node의 spr(SPECIFIER)와 comps는 모두 비어 있어야 한다. Daughter Node에서 HeadDtr의 spr값으로는 그 뒤에 따르는 MarkerDtr가 들어가게 된다. (4)와 마찬가지로 과생성을 막아 주고 문법성을 보장하는 장치로 문법 원리라는 제약조건이 적용된다.

2.3 문법 원리(Grammar Principles)

KPSG에서는 현재 4개의 원리를 사용하고 있으며, 각각은 Head Feature Principle, Marking Principle, SPEC Principle, Semantics Principle이다.

(6) Head Feature Principle

```
% head_feature_principle(Mother, HeadDtr)
head_feature_principle(synsem:cat:head:Head,
                        synsem:cat:head:Head)
if true.
```

위 원리가 말하는 것은 Mother Node의 Head 자질의 값은 Head Daughter의 Head 자질의 값과 일치해야 한다는 것이다.

(7) Marking Principle

```
%marking_principle(Mother, MarkerDaughter)
marking_principle(synsem:cat:marking:Mark,
                  synsem:cat:marking:Mark)
if true.
```

Marking Principle은 Mother Node의 marking 값은 Marker Daughter의 marking 값이 일치해야

**(제 10회 한글 및 한국어 정보처리 학술대회)**

한다는 것을 말한다. 따라서, [[S]-고]와 같은 경우 [S]는 Head Daughter이고, [-고]는 Marker Daughter를 구성하는데, 이 두 구절이 결합하면서 이루어진 보문에서 synsem:cat:marking 자절의 값은 Marker Daughter에서 계승된다.

(8) SPEC Principle

```
% spec_principle(HeadDtr, SprDtr)
spec_principle(X, synsem:cat:head:(mark,
    spec:X))
    if true.
```

동일한 Mother Node의 Daughter Node 사이에서 Marker Daughter는 그것이 결합하는 Head Daughter를 명시하고 있고, 명시된 Head Daughter와만 결합할 수 있다는 것을 말한다. 따라서, 조사 {-이/가} 등은 명사구와만 결합할 수 있고, 보문소 {-고}는 완형 보문만을 이루게 된다. 그리고 '-(시)고 있다'와 같은 보조 용언 구문을 구성하는 {-고}는 용언과만 결합하도록 제약한다.

(9) Semantics Principle

```
% semantics_principle(Mother, SemHead)
semantics_principle(synsem:cont:Cont,
    synsem:cont:Cont)
    if true.
```

Semantics Principle은 Mother Node와 Semantic Head Daughter의 CONT(ent)값이 동일해야 한다는 것을 제약한다.

**2.4 어휘 항목**

어휘 항목 중 '믿다'와 보조용언 '있다'는 다음과 같이 구성되어 있다.

(10) '믿다'에 대한 KPSG 사전 항목

```
mit --->
synsem:((cat:(head:(verbal,
    vform:base,
    infl:bfalse,
    hon:bfalse,
    tnsf:bfalse,
    morphword:bfalse,
    regtype:irt_reg,
    vtype:vb_verb,
    sl:sl_default,
    st:st_default),
    spr:[],
    subj:[(@np_gf(gf_subj), @np_ind(Index1))],
```

```
comps:[(@np_gf(gf_subj),
    @np_ind(Index1)),
    (@ comp_s(comp_ko, Prop))],
    marking:unmarked,
    phoncnstr:opensyl)),
    marking:unmarked,
    phoncnstr:opensyl)),
    (cont:nucleus:(qfpsoa_believe,
    (qfp_believer:Index1,
    soa_arg:Prop) ))).
```

(10)의 '믿다'는 주어와 보문소 '-고'가 결합된 완형 보문을 보충어로 취하도록 사전 항목이 구성되어 있다. CONT(ent)는 의미 부분으로 QFP\_BELIEVER는 문장의 주어와 동일한 지표를 가지도록 되어 있고, 믿음의 대상은 보충어인 완형 보문이 된다.

(11) 보조용언 '있다'에 대한 KPSG의 사전 항목

```
iss --->
synsem:((cat:(head:(verbal,
    vform:base,
    infl:bfalse,
    hon:bfalse,
    tnsf:bfalse,
    morphword:bfalse,
    regtype:irt_reg,
    vtype:vb_adj,
    sl:sl_default,
    st:st_default),
    spr:[],
    subj:[(@np_gf(gf_subj),
    @np_ind(Subj))],
    comps:[synsem:((cat:(head:(verbal,
        vform:base,
        morphword:bfalse),
        subj:[(@np_gf(gf_subj),
            @np_ind(Subj))],
        comps:[],
        marking:aux_ko)),
        (cont:Prop))],
    marking:unmarked,
    phoncnstr:closedsyl)),
    cont:(nucleus:(qfpsoa_state_progress,
    (soa_arg:Prop))))).
```

보조용언 '있다'는 사전 항목에서 보조용언어미 '-고'가 첨가된 절을 보충어로 취하며, CONT(ent)는 보충어로 취한 절의 상태 진행을 나타낸다.

**2.5 음운 제약 및 어휘 규칙**

(제 10회 한글 및 한국어 정보처리 학술대회)

음운 제약은 실사부와 허사부의 결합에서 매개 모음의 삽입 등과 같은 처리를 하기 위한 제약 규칙이다. 어휘 규칙은 용언의 활용과 관련된 처리를 담당하며, 어간을 바탕으로 어휘 생성을 제약한다.

문의 종류(Sentence Type: ST) 및 문계(Sentence Level: SL)의 결정은 용언 생성 어휘 규칙부에서 생성과 함께 결정된다. 즉, 어간에 결합되는 어미의 종류 및 유형에 따라서 문장의 종류 및 문장의 계층이 결정된다. 따라서, '있다'라는 어휘를 바탕으로 생성된 어휘 '있었다'의 HEAD 자질-값을 보면 다음과 같다.

(12) 문의 종류 및 문계

HEAD verbal

HON bfalse  
INFL btrue  
REGTYPE irt\_reg  
MORPHWORD btrue  
SL sl\_plain  
ST st\_dcl  
TNSF btrue  
VTYPE vb\_adj  
VFORM past

(12)에서 보듯이 '있었다'는 문의 종류로 평서문(st\_dcl), 문계로 평대(sl\_plain)를 값으로 갖는다.

3. 분석 예

문장 "김-이 운호-가 밥-을 먹었다-고 믿-고 있었다"에 대한 최종 분석결과는 (12)와 같다.

(13) "김-이 운호-가 밥-을 먹었다-고 믿-고 있었다"의 분석 결과

STRING:

0 kim 1 i 2 wunho 3 ka 4 pap 5 ul 6 mekessta  
7 ko 8 mit 9 ko 10 issessta 11

CATEGORY:

sign

SYNSEM synsem

CAT cat

COMPS e\_list

HEAD verbal

HON bfalse  
INFL btrue  
REGTYPE irt\_reg  
MORPHWORD btrue  
SL sl\_plain  
ST st\_dcl

TNSF btrue  
VTYPE vb\_adj  
VFORM past  
MARKING marking  
PHONCNSTR phsylytype  
SPR list  
SUBJ list  
CONT psoa  
NUCLEUS qfpsoa\_state\_progress  
SOA\_ARG psoa  
NUCLEUS qfpsoa\_believe  
QFP\_BELIEVER ref  
SOA\_ARG psoa  
NUCLEUS brel\_mekta

EATEN ref

EATER ref

내포된 문장 "운호-가 밥-을 먹었다"까지의 중간 분석 결과는 (13)과 같다.

(14) "운호-가 밥-을 먹었다"의 분석 결과

sign  
SYNSEM synsem  
CAT cat  
COMPS e\_list  
HEAD verbal

HON bfalse  
INFL btrue  
REGTYPE irt\_reg  
MORPHWORD btrue  
SL sl\_plain  
ST st\_dcl  
TNSF btrue  
VTYPE vb\_verb  
VFORM past

MARKING marking  
PHONCNSTR phsylytype  
SPR list  
SUBJ list  
CONT psoa  
NUCLEUS brel\_mekta  
EATEN ref  
EATER ref

Edge created for category above:

from: 2 to: 7

string: wunho ka pap ul mekessta

rule: complement\_head

# of dtrs: 1

(15) "운호-가 밥-을 먹었다-고"의 분석 결과는 다음과 같다.

```

sign
SYNSEM synsem
  CAT cat
  COMPS e_list
  HEAD verbal
    HON bfalse
    INFL btrue
    REGTYPE irt_reg
    MORPHWORD btrue
    SL sl_plain
    ST st_dcl
    TNSF btrue
    VTYPE vb_verb
    VFORM past
  MARKING comp_ko
  PHONCNSTR phsyltype
  SPR e_list
  SUBJ list
  CONT psoa
    NUCLEUS brel_mekta
      EATEN ref
      EATER ref

```

```

Edge created for category above:
  from: 2 to: 8
  string: wunho ka pap ul mekessta ko
  rule: head_marker
  # of dtrs: 2

```

#### 4. 마무리

ALE를 이용한 한국어 문법 설계의 가능성에 대해서 살펴 보았다. 언어처리를 위한 한국어 문법은 사전 구조, 문법 규칙 등이 일치가 되어서 계산 가능해야 한다. KPSG는 현재 용언의 형태-음운부와 통사부가 어휘 생성 규칙에 의해서 결합되어 있다. 따라서, 선어말 어미나, 어말 어미에 대해서는 통사적 자립성을 허용하지 않는다. 앞으로의 문법 개발에서는 이 부분을 수정 보완해 나가야 할 필요가 있다. KPSG는 앞으로 통합 문법에 바탕을 두고 의미-화용 부분까지 포함해서 하나의 틀 안에 설계할 계획이며, 전산 처리 가능한 문법의 설계를 목적으로 확장해 나갈 것이다.

#### 참고 문헌

- [1] 장석진, 정보기반 한국어 문법, 1995, 한신문화사
- [2] Carpenter, B., G. Penn, The Attribute Logic Engine User's Guide, Version 3.0 Beta, 1998
- [3] Davis, A., Lexical Semantics and Linking and the Hierarchical Lexicon, 1996, Doctoral Dissertation, Stanford University
- [4] Pollard, C., I. A. Sag, Head-Driven Phrase Structure Grammar, 1994, CSLI
- [5] Sag, I., T. Wasow, Syntactic Theory: Formal Introduction, 1997, CSLI(Draft)