# 국어의 교착성과 형태소 분석기의 구현

이 민 행 김 성 묵

제주대학교 한국아이비엠
소프트웨어 연구소

# The Agglutination of the Korean Language and the Implementation of Korean Morphological Analyzer

Minhaeng Lee Seong MooK Kim

Cheju University KSDI, IBM Korea

## 요약

교착어(agglutinating language)에서는 다양한 통사정보가 독자적인 형태소에 내재되어 있다. 국어의 경우 형태소의 분석이 통사구조 분석에 선행되어 이루어져야 하는 이유가 바로 국어의 교착어적인 속성에 기인한다. 이 논문의 전반부에서 국어의 교착성을 명확히 보여주는 등위 접속구문을 핵심어 주도 구구조문법(HPSG)에 의하여 분석한다. 후반부에서는 PROLOG로 구현된 국어의 형태소 분석기와 통사구조 분석기(PARSER)를 소개한다.

## I. 머 리 말

한국어는 언어 유형상 교착어에 속한다. 교착어는 문법적 기능을 지니는 다수의 형태소들이 결합하여 단어의 구조나 통사적 구조를 나타내는 특징을 갖는다. 따라서 이러한 언어에서는 형태소들이 갖는 통사적 기능들로 인해서 형태소의 분석이 없이는 문장의 문법적 관계를 포착하기가 어려운 일이다. 최근 국어의 정보처리 분야에서는 이미 국어 형태소 분석기의 개발이 다양하게 시도되어 왔다. 이러한 시도와 더불어 전산언어학적 입장에서 형태소 분석의 동기와 그 이론적 타당성 제시가 병행되어야 하는 것은 자명한 일이다. 본 논문에서는 이러한 이유에서 국어의 교착성을 명확히 보여 주는 등위 접속구문을 대상으로 핵심어 주도 구구조문법 내에서 형태소 분석을 시도하겠다. 또한 이러한 논의에 근거한 국어 형태소 분석기와 통사분석기의 구현 과정을 제한적으로 소개하겠다.

## II. 본 론

### 1. 국어의 등위구조 접속구문

국어는 등위구조에서 조합성의 명제가 언어 보편적인 성격을 지닌다는 사실에 대한 강한 증거를 제시한다. 국어의 등위구조는 국어가 교착어의 전형적인 속성을 보여준다는 사실을 명확히 한다. Stechow(1990:48)에 따르면 교착어는 굴절어 보다 형태론적으로 본질적으로 명약관화하다고 한다. 즉, 교착어는 다양한 문법 정보가 독자적인 형태소 속에 내재되어 있는 경향을 보인다. 이러한 진술은 국어의 등위 표현을 통해 증명할 수가 있다.

국어의 등위구조를 위해 (1)a, b의 두 관할규칙과 (2)a, b의 두 선형 규칙을 가정하기로 한다.

(1)a. SRK1

```
SYN¦LOC¦CONJ  SUBST(1 ,2)
DTRS     HEAD-DTR¦SYN¦LOC¦HEAD¦CONJ   1
         CONN-DTR     PHON   2
                      SYN¦LOC¦HEAD¦CONJ   2
         where 2 =  { WA, KO, NA,... }
```

b. SRK2

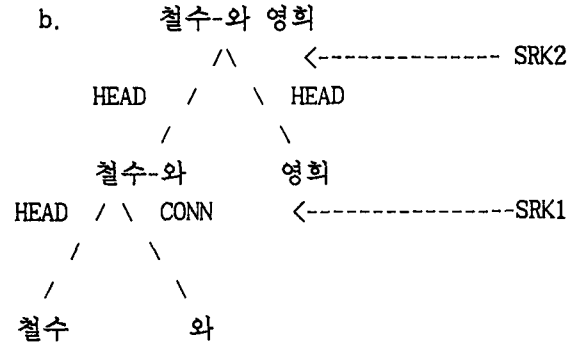
```
SYN¦LOC¦LEX       -
DTRS     HEAD-DTR¦SYN¦LOC
         HEAD-DTR¦SYN¦LOC¦HEAD¦CONJ 1
         where 1 =  { WA, KO, NA,... }
```

(2)a. [CONJ X] < Y
[LEX - 1]

b. Y < [MAJ CONJ]
[LEX +]

수형도 (3)b는 등위구조 '철수와 영희'의 분석을 위하여 관할규칙이 관여되어야 한다는 사실을 보여준다.

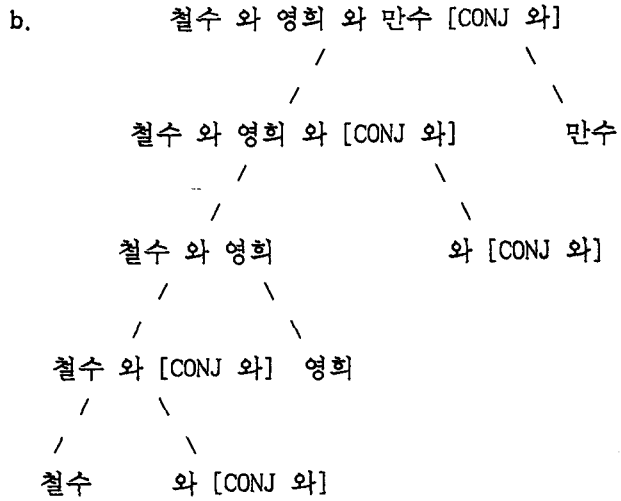
(3)a. 철수-와 영희

b.



(1)b의 관할규칙 SRK6는 두개의 HEAD 성분이 서로 조합하는 것을 허용한다.(1)a의 지배규칙 SRK1은 하나의 구성성분과 접속어가 서로 복합성분을 형성할 수 있음을 보장한다. 이때 접속어는 보족어나 부가어가 아니라 핵심어 성분에 대해 상대적으로 연결성분(CONNECTOR)으로 표시된다.규칙 SRK1의 적용에 따라 접속어는 새로 형성된 성분 내에 존재한다는 정보가 지배된다.

두 규칙의 귀환적 적용을 통해 '철수-와 영희-와 만수'와 같은 복합표현도 다음과 같이 기술될 수 있다.

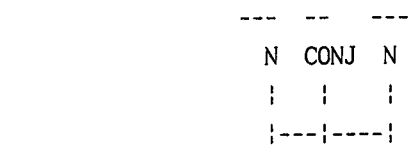
(4) a. 철수와 영희와 만수

b.



다음에서는 한국어에서 어떠한 성분들이 서로 등위 접속 가능한지를 살펴 보겠다.

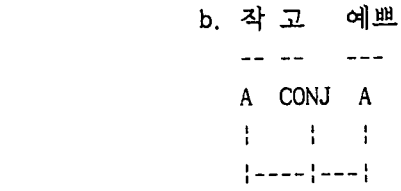
(5)

a. 철수 와 영희가 서로 좋아한다.



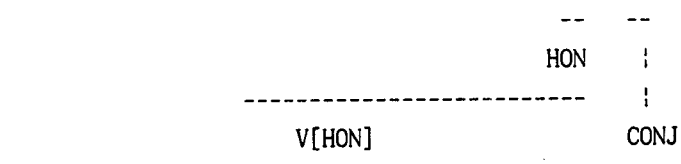

두개의 명사(구)가 서로 등위접속하고 있다.

b. 작 고 예쁘 ㄴ 영희

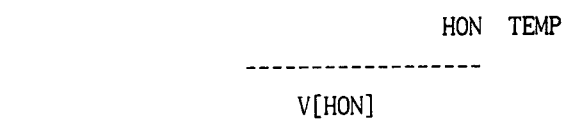



두개의 형용사가 등위접속하고 있다.

c. 학생시절에도 부지런하 - 시 - 고

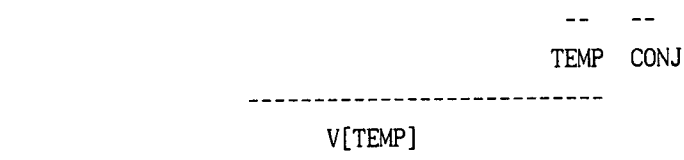



지금도 부지런하-시- ㄴ 김 교수님

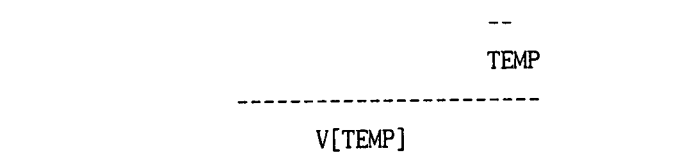



존칭 형태소를 지닌 두개의 동사구가 서로 등위접속하고 있다.

d. 학생시절에도 부지런하 - 였 - 고

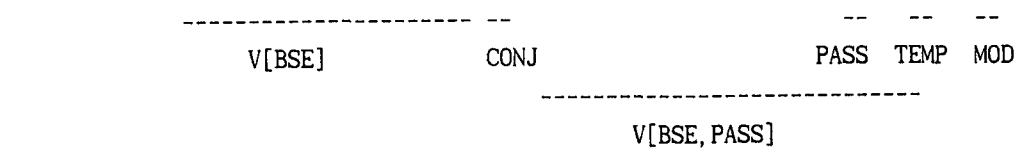



지금도 부지런하 - ㄴ 철수



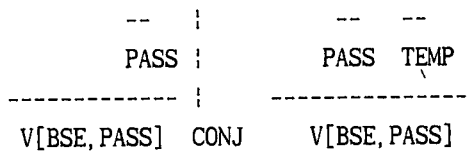

시제형태소를 지닌 두개의 동사구가 서로 등위접속하고 있다.

e. 소녀가 아이스크림을 사-고 그것이 그녀에게 건네 -지 - 었 - 다.




두개의 동사의 어간이 서로 등위접속하고 있다.

f. 아이스크림이 소녀에게 팔- 리- 고 건네 -지 - 었 - 다.

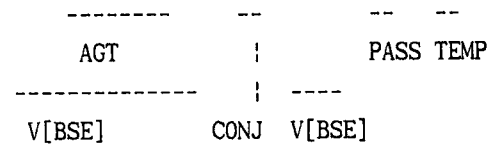



피동 형태소를 가진 두개의 동사의 어간이 서로 등위접속하고 있다.

g. * 철수 -가- 와 영희 -가 서로 좋아 한- ㄴ - 다.

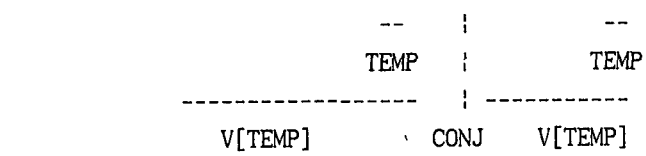

두개의 주격 명사구는 서로 등위접속될 수 없다.

h. * 아이스크림 - 이 소녀에 의해 사 - 고 건네 - 지 - 었 - 다.




피동 형태소는 어떠한 등위접속된 동사 어간을 보족어로 간주할 수 없다. 이것은 동사의 피동 형태가 어휘적으로 생성되었음을 나타내 준다. 또한 이것은 사동 형태에서도 마찬가지로 작용한다.

i. * 부지런 하 - ㄴ - 고 영리하 - ㄴ 철수




시제 형태소를 지닌 두개의 동사구가 동시에 현재 시제를 표시할 때는 서로 등위접속될 수 없음을 보여 준다.

이상의 예에서 (6)에서와 같은 핵심어 원리(HEAD PRINCIPLE)와 이전의 관할규칙 SRK1과 SRK2를 HPSG내에서 기술할 수 있다.

(6) 핵심어 원리

```
[DTRS]  ==>     SYN¦LOC¦HEAD          1
                DTRS¦HEAD-DTR¦SYN¦LOC¦HEAD      1
```

그러나 (5)h를 배제시키기 위해서 지배규칙 SRK1을 다음과 같이 수정해야만 한다.

(7) SRK1+

```
SYN¦LOC¦CONJ    SUBST (1,2)
DTRS - HEAD-DTR¦SYN¦LOC¦HEAD¦MAJ      X
                                CONJ    1

       CONN-DTR   PHON    2
                  SYN¦LOC¦HEAD¦CONJ   2

          where   2 = {WA, KO, NA,... } &
                  X   P
```

(5)h의 비문법성을 예측하기 위해선 사역 형태소와 -지, -이, -히 -기 -리 등의 피동 형태소는 자립형태소가 아니라 구속형태소이며 통사적으로 등위접속된 동사어간의 보족어를 끌어오지 않는다고 할 수있다. 나아가 아래의 (9)a, b와 같은 예문의 비문법성을 처리하기 위해서 핵심어 원리를 다음과 같이 교정해야 한다.

(8) 확대 핵심어 원리

```
[DTRS] ==>

SYN   LOC¦HEAD      1
      BIND¦SLASH    2

DTRS¦HEAD-DTR¦WYN    LOC¦HEAD      1
                     BIND¦SLASH    2
```

(9)a. * 영희-가 좋아하-나 미희-를 사랑하-는 철수

```
          --        --      --        --
          N         CONJ    A         TEMP
      ------------     ----------------
V[SC<>,SL<[CASE A]>] V[SC<CASE N]>,SL<>]

   ==> V[SC<CONFL>, SL<CONFL>]
```

b. * 영희-가 좋아하-고 미희-가 책-을 선물하-ㄴ 철수

```
          --        --       --    --       --
          N         CONJ     N     A        TEMP
   ---------------     ----------------------
   V[SC<>, SL<[CASE A]>]  V[SC<>,SL<[CASE D]>]

       ==> V[SL<[CASE CONFL]>]
```

그런데 지금까지의 논의와 제안이 여러 시제 형태소를 지닌 동사구의 등위 접속 구분의 분석에 적절하지 않다. 다음의 예들을 살펴 보자.

(10)a. 학생시절-에-도 부지런하-었-고

```
   ------------------------- --
   V[PRES, SL<[CASE N]>]     CONJ
```

지금-도 부지런하-ㄴ 철수

```
   -----------------
   V[PAST, SL<[CASE N]>]
```

b. * 부지런하-ㄴ-고 영리하-ㄴ 철수

```
                --
              CONJ
   ----------       --------
   V[PAST, SL<[CASE N]>]  V[PAST, SL<[CASE N]>]
```

c. 학생-때 부지런하-었-고 영리하-었-던 철수

```
                    -- --       --
            TEMP[PAST] CONJ  TEMP[PAST]
   ------------------    -------------
   V[PAST, SL<[CASE N]>]  V[PAST, SL<[CASE N]>]
```

d. * 철수-가 그 책-을 펴보-았-고 사-ㅆ-다

```
                                  -- --    --
                                TEMP CONJ TEMP
                               -------    -----
                              V[PAST, V0] V[PAST, V0]
```

(10)a의 예에서 문제가 되는 것은 핵심어 원리의 위반과 관계가 있다. 왜냐하면 [TEMP PAST]와 [TEMP PRES]의 두 자질쌍이 충돌하기 때문이다.(10)b의 예의 비문법성은 현재시제 형태소를 가진 두개의 동사구가 서로 접속할 수 없기 때문인데, 이 사실은 과거시제 형태소를 가진 동사구들 사이에서는 적용되지 않는다. (10)c와 (10)d에서 그 차이를 알 수 있다. 그러나 이러한 문제는 아주 지협적이고 이에 대한 해결은 시도하지 않겠다.

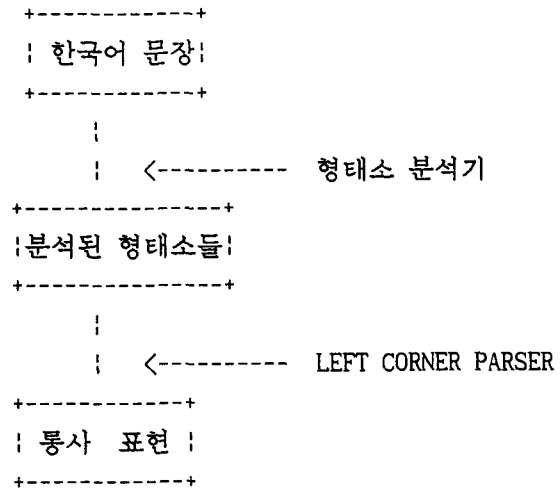
## 2. 한국어의 형태소 분석기와 LEFT-CORNER PARSER

한국어 형태소 분석기를 포함한 한국어 파서를 논의하겠다. 실행언어로 CLOCKSIN/MELLISH(1981)의 제 1세대 프롤로그(PROLOG)가 사용되었다. 프로그램은 IBM 호환 PC에서 수행된다.

앞 절에서 몇몇의 속박형태소가 한국어에서 통사적으로 독자적인 문법기능을처리할 수 있다고 말했다. 따라서 어간으로부터 속박형태소를 떼어놓는 형태소 분석이 필수적으로 이루어져야 한다.

한국어의 분석은 대략 다음과 같다.

(11)




이제 분석 예를 살펴보자

(12) 예문: 철수가 영희를 좋아한다

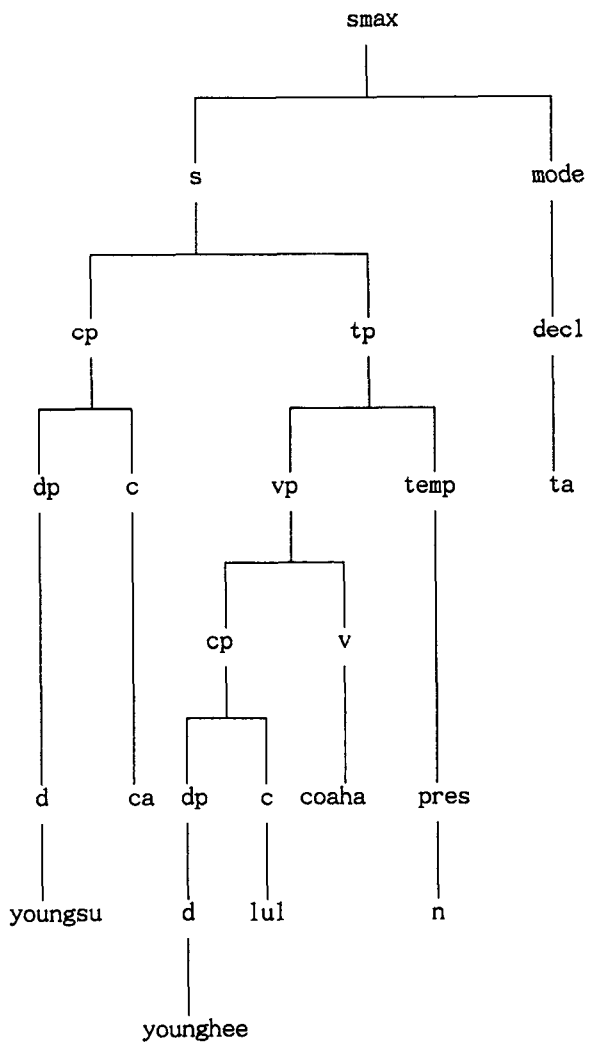
/* test: [cheolsuka, youngheelul, coahanta] */

a. THE MORPHOLOGICAL ANALYSIS:
[cheolsu, ka, younghee, lul, coaha, n, ta]

b. PARSE:

smax(s(cp(dp(d(youngsu)),c(ka),tp(vp(cp(dp(d(younghee)),
c(lul)), v(coaha)),tense(pres(n)))), mood(decl(ta)))

c.



(12)a에서 c의 예는 '토니오가 영희를 좋아한다'라는 문장의 분석과정을 보여준다. 맨처음 프로그램은 문장을 리스트로 받아들여 [cheolsuka,youngheelul, coahanta]로 변환한다. 다음에 형태소 분석기가 사전을 접근하여 형태소 분석을 실행하고 (12)a 처럼 분석 결과를 제시해 준다. 그리고 나서, LEFT-CORNER PARSER가 (12)a를 입력으로 취하여 술어-논항-구조((12)b)와 수형도((12)c)를 통사분석 결과로 제공한다.

여기서 형태소 분석기의 세부사항에 대해 살펴보면, 형태소 분석기는 귀환적으로 호출되는 술어 morphsyn에 기인한다. 이 술어는 리스트를 입력으로 취하고 술어 morphanal이 리스트의 첫째 원소를 분석하여 리스트의 마지막 원소까지 반복하여 분석한다. 이때 미리 정의된 술어 append가 분석된 결과를 모으게 된다.

```
(13) ?- morphsyn([youngsuka,youngheelul,coahanta],C).
          C = [youngsu,ka,younghee,lul,coaha,n,ta]

(14) ?- morphsyn([youngsuka,youngheelul,coahanta],C).
          C = [youngsu,ka,younghee,lul,coaha,n,ta]
       --> ?- morphanal(youngsuka,A).
              A = [youngsu,ka]
       --> ?- morphsyn([youngheelul,coahanta],B)
                    B = [younghee,lul,coaha,n,ta]
            --> ?- morphanal(youngheelul,B1).
                       B1 = [younghee,lul]
            --> ?- morphsyn([coahanta],B2).
                          B2 = [coaha,n,ta]
                  -- ?- morphanal(coahanta,B2).
                          B2 = [coaha,n,ta]
            --> ?- append(B1,B2,B).
                    B = [younghee,lul,coaha,n,ta].
     --> append(A,B,C).
          C = [youngsu,ka,younghee,lul,coaha,n,ta]
```

술어 morphanal은 복잡하게 구성된 단어를 사전을 이용하여 통사상 상응하는 단위로 최대 분석해 내어야 하고 그 결과를 리스트 형태로 제공하고 있다. (14)에서 술어 morphanal이 3번 이용되는 것을 볼 수 있다.

```
(15) a. ?- morphanal(youngsuka, A).
           A = [youngsu, ka]
     b. ? - morphanal(youngheelul, B1).
           B1 = [younghee, lul]
     c. ? - morphanal(coahanta, B2).
           B2 = [coaha, n, ta]
```

(15)a-c의 분석은 (16)a-g까지의 예에 나타난 것과 같은 사전 어휘 항목을 근거로 가능하다.

```
(16)a. word(youngsu, npr(d(youngsu))).
    b. word(younghee, npr(d(younghee))).
    c. word(coanh, v2(v(coaha))).
    d. word(ka, c(c(ka), nom)).
    e. word(lul, c(c(lul), acc)).
    f. word(n, t(pres(n), pres)).
    g. word(ta, mood(decl(ta))).
```

통사 분석의 예를 살펴보자. 실행시에 등위구조와 명사화 구조를 기술할 수 있는 통사분석기에 중점을 두었다. 따라서 한국어 특히 등위구문에 나타나는 LEFT-RECURSION을 피하기 위해서 LEFT-CORNER-PARSER를 채택하였다. 문법의 형식화를 위해 한국어가 형상적 언어임을 가정하고 문법기능 사이에 'NOM <DAT <ACC'의 위계가 존재함을 가정한다. 또한 통사분석기는 기본 어순을 가지는 문장을 입력으로 받아서 문법성을 판단하고 이에 대한 통사표현을 제공한다.

문법의 특성은 다음과 같다.

첫째, 통사분석기는 MOOD, TENSE, CASE를 독자적인 범주로 파악하여 표시한다.

```
(17)a. smax(smas(S, mood(M))) --> [s(S), mood(M)].
    b. s(s(CP, T)) --->
                [cp(CP, nom), tp(T, TNS)].
    C. tp(tp(VP, tense(T)), TNS) ---> [vp(VP), t(T, TNS)]
    d. cp(cp(DP, P), CASE) --->
                [dp(DP, c(P, CASE)].
    e. word(ta, mood(decl(TA))).
    f. word(ss, t(past(ss), past)).
    g. word(i, c(c(i), nom)).
```

둘째로, 소위 전통적인 명사구를 관사구(DETERMINER PHRASE)로 이해하고 고유명사를 0-항 관사로, 관사를 1-항 관사로 보겠다.

```
(18) a. dp(dp(D)) ---> [npr(D)].
     b. dp(dp(Det, N1)) --->
              [det(DET), n_bar(N1)].
    C. dp(dp(N)) ---> [cn(N)].
    d. word(cheolsu, npr(d(cheolsu))).
    e. word(ku, det(demon(ku))).
    f. word(chayk, cn(n(chayk))).
```

통사 분석기는 술어 parse와 관련이 있다. 이 술어는 리스트를 입력으로 받아들여 술어-논항 구조를 출력한다. 또한 술어 print_cstr를 통해 (13)c처럼 수형도로 변형될 수 있다.

통사분석의 예를 살펴보자.

```
(19) ?- parse([youngsu,ka,younghee,lul,coaha,n,ta],[],L1),L1=..[_,L].

        L1 = smax(

             smax(s(cp(dp(d(youngsu)), c(ka)),
               tp(vp(cp(dp(d(younghee)), c(lul)), v(coaha)),
                 tense(pres(n)))), mood(decl(ta)))


        L = smax(s(cp(dp(d(youngsu)), c(ka)),
               tp(vp(cp(dp(d(younghee)), c(lul)), v(coaha)),
                 tense(pres(n)))), mood(decl(ta)))
```

문장 '철수가 영희를 좋아한다'의 통사구조는 (19)와 같은 술어-논항 구조이다. 이것은 술어 print_cstr에 의해 (13) c처럼 수형도로 표시될 수 있다.

## III. 맺음말

국어 형태소 분석기의 구현이 전산학자들에 의해 많이 시도되었거나 시도되고 있다. 그러나 국어의 경우 형태소 분석에 대한 언어학적인 동기가 무엇인지에 대한 논의가 드물었다. 이 논문에서는 등위접속 구문의 통사분석을 통해 교착어로서의 국어의 특성이 바로 형태소 분석에 대한 동기를 부여함을 논의했다. 또한 형태소 분석이 국어의 통사구조 분석에 어떻게 기여하는지를 보였다.

## 참고문헌


[1] Choi, J. M., M.S. Song, "A Prolog-based Korean-English Machine Trnaslation System and its efficient Method of Dictionary Management," Logic Programming '85 proceedings of the 4th Conference, 236-245, 1985

[2] Clocksin, W.F./C. Mellish, Programming in Prolog, 1981

[3] Colmeraur, A. Theoretical Model of Prolog II, Logic Programming and its Applications, Norwood 3-31,1986

[4] Colmeraur, A. Intorduction to Prolog III, ms,1990

[5] Gazdar, G./C. Mellish, Natural Language Processing in PROLOG, Workingham, 1989

[6] Gross, S. "Adjektivkomposita u.ihre Theta-Eigenshcaften: Linguistische Analyse und Implementierung" in If/Prolog. Universitaet Muenchen, 1987

[7] Kartunnen L.,Kaplan R., Zaenen A.," Two-level Morphology with Composition," COLING 92, 1992

[8]Koeskennimi K.,"Two-level Morphology, A general Computational Model for Word-Form Recognition and Production," Publications,No.11. 160pp, 1983

[9] Lee, M. Kontrastive Syntax und maschinelle Sprachanalyse im Rahmen einer Unifikation, Frankfurt/m. Grammatik, 1992

[10] Lerner, Ch. Prolog and Lingusitics. 1990

[11] Pereira,/Shieber Prolog and Natural Language Analysis. CSLI, 1987.

[12] Pollard/Sag Information-Based Syntax and Semantics, vol I. CSLI ,1987

[13] Stechow,A.v."Morphologie und Syntax," Arbeitspapier Nr16, Uni. Konstanz, 1990

[14] 강승식, 김영택, "한국어 형태소 분석기에서 불규칙 용언의 분석 모형," 정보과학회논문지, VOL.19, No2, 1992.

[15] 고영근, 남기심, 표준국어문법론, 1985

[16] 김성묵, "술어적 보족어에 대한 통사, 의미론적 연구: 기계 구문분석을 위한 시도," 서울대학교 석사학위논문, 1988

[17] 신수송, 통합문법이론의 이해, 한국언어학회, 1991