

# 조건 단일화 기반 PATR II 문법을 이용한 우리말 분석

양승원, 황이규, 이기오, 이용석  
전북대학교 전자계산학과

## An Analysis for Korean with PATR II Based on Conditional Unification

Seung-won Yang, Yi-kyu Hwang, Gi-o Lee, Young-seok Lee  
Dept. of Computer Science, Chunbuk National University

### < 요약 >

본 논문은 조건 단일화 기반 문법을 사용한 우리말 분석에 관한 연구이다. 우리말은 문법이 매우 간단한 반면 생략현상이 빈번하고 어순이 자유로운 언어이다. 우리는 이와같은 특징을 잘 소화해내기 위한 방법으로 조건 단일화 기법을 제안한다. 본 논문에서 사용한 문법은 PATR II이며 실험을 위한 시스템은 Sparc Station I\* 위에서 Lisp를 사용해 구현하였다. 이를 바탕으로 우리말 파싱시 가장 어려운 문제들이라고 여겨져 온 생략, 도치, 보문화등을 조건 단일화 기법을 이용하여 분석할 수 있음을 보인다.

## I. 서론

컴퓨터를 사용하여 자연어를 처리하려는 시도는 컴퓨터의 탄생 이후 꾸준히 이어져 왔다. 자연어 처리의 어떤 분야에서이건 가장 근간이 되는 과정은 파싱이며 이를 위한 연구 또한 활발하게 진행되고 있다. 파싱을 효율적이고 정확하게 하기 위해 다수의 문법학자들과 전산학자들은 단일화라는 연산만을 갖는 단일화 기반 문법을 사용하고 있다[6].

우리말의 분석에서도 다수의 단일화 기반 분석 기법이 발표되어 있다[10, 11, 13, 15]. 이들 연구들은 각자 나름대로의 특성을 가지고 있으며 크게 문맥자유문법(CFG)과 의존문법으로 나눌 수 있다. 문맥자유문법에서는 각 문장성분의 결합관계에 중점을 두어 부분과 전체의 관계를 기술하며, 의존문법에서는 문장 성분간의 의존관계(수식-피수식)를 찾아내는 것을 분석 목표로 한다. 그렇지만 우리가 보는 관점은 의존문법과 문맥자유문법이 완전히 별개의 독립된 포맷리즘이기 보다는 언어라는 하나의 현상을 약간 다른 시각에서 접근하고 있다는 것이다. 따라서 우리는 명료함과 유한성을 제공하는 문맥자유문법에서 각 문장성분들 간의 의존관계를 찾는 과정을 분석이라고 생각한다.

우리말은 문법구조가 비교적 간단한 반면, 문장성분의 순서가 자유롭고 필수요소의 생략이 빈번해 문법 자체에서 생겨나는 모호성이 많다. 이를 처리하기 위한 방법은 문법 규칙을 필요한 만큼 기술하는 방법과 규칙은 최대한 간단하게 기술하되 이로 인해

서 발생하는 모호성을 처리해 나가는 방법이 있다. 전자는 문법이 매우 복잡해지고 출현 빈도수가 극히 적은 문장 형식에도 규칙이 필요하기 때문에 비효율적이다. 후자는 모호성은 심하게 발생하지만 이해하기 쉽고 대부분의 문법현상을 표현할 수 있다. 본 논문에서는 후자의 방법을 택한다. 후자의 대표적인 연구인 [13,14]에서는 이진 문법을 통해 우리말 문법을 체계화 했으며, 이때 발생하는 모호성은 규칙을 간단히 하지 않아도 우리말의 특성상 발생하는 것으로 보고 있다. 그러나 어떤 형태로든 이 모호성은 해결되어야 하며 우리는 이를 위해 조건 단일화 방법을 제시한다.

조건 단일화는 각 규칙에서 감축이 일어나기 위해 수행되는 단일화를 선택적으로 수행한다는 개념이다. 즉, 자질구조 안의 임의의 속성 값을 동적으로 검사한 후 조건에 맞는 단일화만을 수행한다는 전략이다. 우리는 이러한 특성을 표현하기 위해 단일화 기반의 PARTII [6]를 사용한다. 이 포말리즘은 간단하고 표현력이 뛰어날 뿐만 아니라 기존의 여러 단일화 기반 포말리즘 - CG, GPSG, LFG, FUG, HPSG - 의 이론도 표현 할 수 있다.

본 논문에서는 PARTII를 확장하여 조건 단일화를 수행함으로써 우리말의 분석에 대하여 기술한다. 제 II 장에서는 연구에 도입된 개념에 대해 기술하고, 제 III 장에서는 우리말 분석기를 개략적으로 설명하며, 제 IV 장에서는 우리말 분석시 가장 어려운 문제중의 하나인 생략현상 처리에 관하여 논한다. 마지막으로 제 V 장에서 결론 및 향후 연구방향을 토의한다.

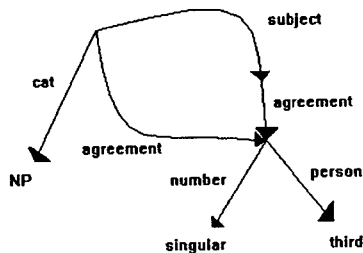
## II. 연구에 도입된 개념

### 2.1 PARTII

[6]에 의해 제안된 PARTII는 이론적으로 잘 정리된 간단한 단일화 기반 문법 포말리즘이다. PARTII는 기존의 단일화 기반 문법들의 특성을 표현할 수 있도록 설계되었다. 이 소절에서는 PARTII의 간단한 특징만을 기술하며 이 문법 포말리즘에 관한 보다 깊은 부분은 [6]를 참조하기 바란다.

#### 2.1.1 자질구조

PARTII 포말리즘에서 구문 규칙과 어휘 규칙은 정보(Information)를 표현하기 위해 공통적으로 속성-값 쌍의 집합으로 기술한다. 이 속성-값 쌍을 자질구조라 하는데 이 자질구조를 효율적으로 표현하기 위해 DAG(Directed Acyclic Graph)를 사용한다. <그림 1>은 영어 문장의 일치 현상을 위한 표현구조를 보여준다.

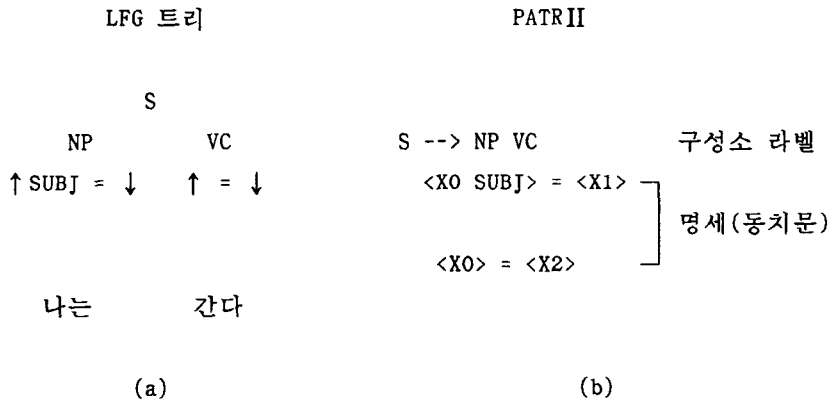


<그림 1> DAG

### 2.1.2 조합 규칙

규칙은 구성소 라벨(Constituent label)과 명세(Specification)로 이루어진다 [3, 6]. <그림 2(b)>에서 볼 수 있듯이 구성소 라벨은 문맥자유문법의 형식으로 표현하며 작은 구성소들이 큰 구성소로 모이는 방법을 나타낸다. 명세는 동치문(Identity statement)으로 표현하며 모아지는 구성소들의 문법적 관계를 표현하기 위한 제약들이다. PATRII에서 이 동치문은 부구성소들을 그들의 단일화로 대치시키라는 명령으로 해석되며, 구성소 라벨의 오른쪽에 있는 부구성소들이 왼쪽의 보다 큰 구성소로 감축(Reduction)될 때 동치문에서 기술된 순서에 따라 단일화가 일어난다. 이렇듯 PATRII에서는 동적인 단일화에 정적인 동치문을 기술하여 절차적 사고를 가능케 한다.

또한 PATRII는 조합규칙을 적절히 기술함으로써 기존의 단일화 기반 문법들의 특성들도 표현할 수 있도록 설계되었다. 예를 들면 “나는 간다”라는 간단한 우리말 문장이 LFG로 표현되는 형식은 <그림 2(a)>와 같은데, 이것을 PATRII로는 <그림 2(b)>와 같이 표현한다.



<그림 2> LFG 와 PATRII

<그림 2(b)>의 명세부분에서 X0는 구성소 라벨 좌변의 S와 대응되고 X1, X2는 각각 우변의 왼쪽부터 차례로 대응되는 변수이다. 즉, X1은 NP를, X2는 VC를 대신한다.

### 2.2 조건 단일화를 위한 포말리즘의 확장

PATRII에서 규칙의 감축조건 검사는 단일화에 의해 실행된다. 즉 모든 구성소의 문법적 관계는 단일화에 의해 검사된다. 그러나 이러한 순수 단일화만으로는 정보를 완벽하게 전달하기가 어렵다[2]. 특히 우리말에서 생략과 도치등은 순수한 구구조문법으로 표현하기에는 복잡한 현상이 빈번하게 발생되는데, 이러한 경우에 순수한 단일화만을 이용하여 이 현상들을 해결해 내기는 상당히 어렵다[10]. 이에 따라 우리는 규칙의 명세부분에 기술된 단일화가 선택적으로 수행될 수 있도록 하는 방법을 택하였다. 이를 조건단일화(Conditional Unification)라 하며, 이것의 실현을 위해 단일화 연산의 조건을 검사할 수 있는 장치를 도입하여 PATRII를 확장한다. 우리의 시스템에 도입된 장치는 다음과 같다. 첫째는 디스정션(Disjunction) 처리기능이다. 이 기능은 규칙을

간단히 기술하는 데 따르는 문제점을 효율적으로 제거하고 사전에서 중복을 감소시키기 위해 도입하였다. 우리의 분석기에서는 디스정선의 기호를 \*OR\*로 쓴다. 어떤 규칙이 여러 범주의 문장을 분석해 내려면 여러 종류의 처리방법이 필요하다. 따라서 하나의 규칙 아래에 여러가지 방법을 기술해 놓고 이들중 조건에 맞는 단일화 연산을 수행해야 한다. 예를 들면 <그림 3(a)>에서 처럼 규칙 S --> XP VC에서 X2(VC)의 문법범주가 자동사인지 타동사인지에 따라서 X1(XP)이라는 미결정된 명사구가 자신의 문법적 역할을 부여받게 되므로 술어인 X2의 종류에 따라 처리 방법의 기술(Description)이 달라진다. 또한 현대의 문법체계에서는 분석을 위한 대부분의 정보가 사전에 기술되어 있어 사전관리가 중대한 문제가 되는데 단어의 중의성 때문에 사전에 쓸데없이 중복되어 기술되어지는 정보가 많다. <그림 3(b)>는 명사 '배'의 여러 의미를 우리 시스템내의 사전에서 디스정선 형식으로 기술한 예를 보여준다.

S -> (· XP VC )

(( \*OR\*

((X2 CAT) = C Vi)

(자동사 처리부분

... ))

((X2 CAT) = C Vt)

(타동사 처리부분

))

·  
·  
·

N -> ( 배 )

(( (XO CAT) = N )

(\*OR\*

(( (XO SEM KEY) = BOAT )

(( (XO SEM EAT) = - )

... )

(( (XO SEM KEY) = FLUIT )

(( (XO SEM EAT) = + )

... )

·  
·  
·

(a) 규칙내의 명세 기술

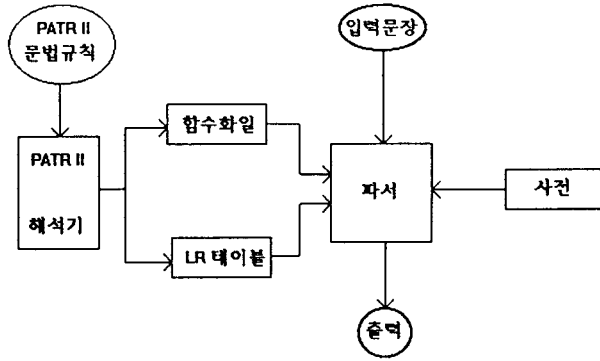
(b) 사전에서의 중복 제거

<그림 3> 디스정선 사용 예

둘째로 LFG[1]에서 도입한 동일성 제약기능 '=C'를 바탕으로, 우리 시스템에서도 이 제약의 표현을 위해 '=C' 기호를 사용한다. 동일성 제약 '=C'는 '='과는 달리 좌우변 자질구조간의 단일화는 일어나지 않으며 단지 좌우항의 동일성만 검사해서 같으면 참, 다르면 거짓이 된다. 이 동일성 제약문이 거짓이면 그 이후 동치문에 의한 단일화는 수행되지 않는다. 셋째는 존재여부 검사기능이다. 이 기능은 우리말 문장안에서 각 구성소의 문법적 역할이 다른 구성소들과의 관계를 비교해 본 후에 비로소 결정되므로 특정 자질구조안의 임의의 속성값을 동적으로 검사하기 위하여 도입되었다. 이 기호에는 '\*EXIST\*'와 이것의 부정인 '\*NONEXIST\*'를 함께 사용한다.

### III. 우리말 분석기

우리의 분석 시스템은 <그림 4>와 같이 구성되어 있다.



<그림 4> 우리말 분석기

여기서 문법 규칙은 앞에서 언급한 형식의 확장된 PATRII 문법을 사용하여 기술한다. PATRII 해석기는 이 문법을 해석하여 함수 화일과 LR 테이블을 생성해 낸다. 함수 화일은 기술된 문법의 syntax와 semantic을 LISP 함수 형태로 가지고 있어 파서가 입력 문장을 파싱할 때 참조된다. LR 테이블은 파싱시 파서의 동작을 결정하는 역할을 수행한다. 파서는 입력 스트링을 받아들여 분석된 문장의 자질구조를 출력한다. 우리가 사용하는 파싱 알고리즘은 현재 발표된 것중 가장 빠르다고 인정받고 있는 Tomita의 방법[7,8]을 사용하였다. 사전은 구문적 정보를 포함하고 있는 엔트리들의 모임이다. 단일화 기반 문법에서는 이전의 구구조 문법과 달리 사전에 구문적 정보를 표현하므로 사전의 관리가 매우 중요하다. 우리의 시스템에서는 사전의 효율적 관리를 위하여 문법 범주의 간소화와 디스정선 표기법을 도입하였다.

실제로 우리의 시스템에서 사전요소과 문법을 기술하는 예와 이것들을 바탕으로 간단한 예문을 분석하는 과정을 살펴보자.

```

vc <-- ( 본다 )
      (((x0 root) = 본다)
      ((x0 cat) = vt))
p <-- ( 가 )
      (((x0 form) = 가)
      ((x0 cform) = 주격 ))
n <-- ( 영화 )
      (((x0 form) = 영화))
n <-- ( 책 )
      (((x0 form) = 책))
  
```

<그림 5-1> 사전 요소 기술

규칙 1 : s <=> ( xp s )  
 (((x1 case) = \*EXIST\* )  
 (\*OR\*  
 (((x1 cform) =C 주격) ----- ㉠  
 (\*OR\* (((x2 cat) =C vi)  
 (x0 = x2)  
 ((x0 subj) = x1))  
 (((x2 cat) =C vt)  
 (x0 = x2)  
 ((x0 subj) = x1))))  
 (((x1 cform) =C 목적격) ----- ㉡  
 (\*OR\* (((x2 cat) =C vi)  
 (x0 = x2)  
 ((x0 subj) = x1))  
 (((x2 cat) =C vt)  
 (x0 = x2)  
 ((x0 subj) = x1))))))

규칙 2 : xp <=> ( np p )  
 ((x0 = x1)  
 ((x0 case) = (x2 form))  
 ((x0 type) = (x2 type))  
 ((x0 cform) = (x2 cform)))

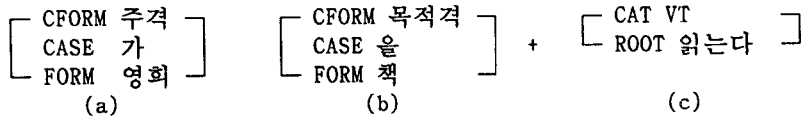
규칙 3 : s <=> ( vc )  
 ((x0 = x1))

<그림 5-2> 규칙 기술

<그림 5-1>과 <그림 5-2>처럼 기술한 사전과 규칙을 기반으로

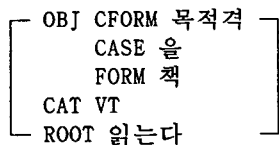
① 영희가 책을 읽는다.

라는 문장을 우리의 분석기가 처리하는 단계를 보면, <그림 6>의 (a), (b)는 규칙 2에 의해, (c)는 규칙 3에 의해 사전으로부터 규칙에 적용된 결과이다.



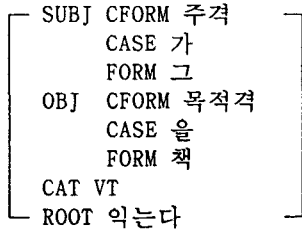
<그림 6> 자질구조 1

<그림 7>는 규칙 1의 ㉡부분에 의해 <그림 6(b)>와 <그림 6(c)>가 단일화된 결과이다.



<그림 7> 자질구조 2

<그림 8>은 최종적으로 규칙 1의 ㉔부분에 의해 <그림 6(a)>와 <그림 7>이 단일화된 결과이다.



<그림 8> 자질구조 3

## IV. 생략 요소가 있는 문장의 분석에

### 4.1 격 조사가 생략된 문장 분석

문장은 기본적으로 주어, 목적어, 술어의 세가지 뭉치로 구성된다. (물론 술어가 자동사인 경우는 주어와 술어로 한 문장이 구성된다). 우리말에서 이 세가지 뭉치는 격 조사에 의해 결정된다. 그런데 이렇게 중요한 역할을 하는 격조사가 여러가지 이유로 생략될 수 있으며[16] 이로 인해 결정적 파싱이 어려워진다.

격조사가 생략되는 예를 들어 보면 아래와 같다.

- ② 영희가 책 읽는다.
- ③ 영희는 철수만 좋아한다.
- ④ 영희는 책만 좋아한다.

①의 경우 주격 조사인 "가"와 목적격 조사인 "을"이 모두 쓰였다. 즉 완전한 우리말 문장이다. 그러나 ②의 경우는 "책"의 문법적 역할을 결정하는 목적격 조사 "을"이 생략되었다. ③의 경우는 보조사 "는"과 "만"이 주격조사, 목적격 조사 대신 사용되었다. 이렇듯 우리말에서는 보조사가 오면 격조사는 탈락되고 보조사가 격조사의 위치에 사용된다.

조사가 생략된 경우의 문법적 관계를 찾아내는 방법을 생각해 보자. ②에서 처럼 하나의 격조사가 생략된 경우에는 생략되지 않은 격조사로부터 생략된 격조사를 유추해 낼 수 있다. 이러한 유추는, 문법적 관계가 결정되지 않는 요소는 술어에 대해 주어나 목적어의 뭉치라는 기본적인 원리로부터 이끌어 올 수 있다. 즉, "읽는다"가 하나의 목적어를 갖는 타동사 이므로 "영희가"가 주어라는 사실로부터 "책"에 목적격 조사가 생략되었음을 알 수 있다. 그런데, 격조사가 모두 탈락된 경우 모호성이 발생한다. ③의 경우에서는 두개의 뭉치 모두 문법적 관계의 명확한 설정이 불가능하므로 다음과 같은 2가지 가능성이 존재한다.

- |         |       |       |         |       |       |
|---------|-------|-------|---------|-------|-------|
| (1) 영희는 | 철수만   | 좋아한다. | (2) 영희는 | 철수만   | 좋아한다. |
| SUBJ    | OBJ   | PRED  | OBJ     | SUBJ  | PRED  |
| (영희가)   | (철수를) |       | (영희를)   | (철수가) |       |

이 2가지 가능성은 모두 옳은 구문구조와 의미구조를 가진다. 이러한 경우에는 오직 발화상황하의 분석에 의해서만 모호성이 제거될 수 있다[17]. 따라서 이 경우에 우리의 분석기는 이 경우 2가지 파스를 모두 생성해 낸다. ④와 같은 문장의 경우에도 역시 아래와 같은 2가지 가능성이 존재하는데,

(1) 영화는	책만	좋아한다.	(2) 영화는	책만	좋아한다.
SUBJ	OBJ	PRED	OBJ	SUBJ	PRED
(영화가)	(책을)		(영화를)	(책이)	

이들중 (2)에서는 "책"이 "좋아한다"의 주어가 될 수 없으므로(의미구조로 볼 때 옳지 않은 문장이므로) (1)만이 남게 된다. 이와 같이 분석을 하기 위해 의미표지(SEMANTIC MARKER)를 사용하는 방법이 제시된 바 있다[4]. 우리도 역시 이 의미표지를 사용해서 이러한 모호성을 해결한다. 그렇지만 의미표지를 무분별하게 설정하여 사용하면 자칫 분석의 일관성을 잃기 쉬우므로 우리는 가능한한 최소한의 의미표지를 사용해서 분석하려고 노력하였다. <그림 9>는 단문 분석을 위한 규칙이다. 이 규칙의 명세 부분은 실제로는 엄격한 LISP 구조이나 여기에서는 읽기 편하도록 알고리즘으로 기술하였다.

```

S -> XP S

if XP에 격 정보가 없으면
  case: S cat = 자동사
      XP => 주어
  case: S cat = 타동사
      (XP => 주어) or (XP => 목적어)
if XP에 격 정보가 있으면
  if XP 격 정보 = 주격
    case: S = 자동사
        XP => 주어
    case: S = 타동사
        S에 주어가 없어야 함
        XP => 주어
  if XP 격 정보 = 목적격
    S cat =C 타동사
    S에 목적어가 없어야 함
    XP => 목적어
  if XP 격 정보 = 보조사
    (XP => 주어) or (XP => 목적어)

```

<그림 9> 단문 처리 규칙

#### 4.2 관형절의 분석

관형절의 분석을 살펴 보면 아래와 같다.

- ⑤ 내가 책 읽는 그를 안다.
- ⑥ 나는 그가 책을 읽는 모습을 보았다.



⑤와 ⑥의 문장은 관형절의 예이며 ⑤는 관계관형화, ⑥은 명사구 보문화 현상이 나타나는 문장이다. 여기서 관계관형화란 관형절이 수식하는 명사구는 자신의 절안에서 생략된 뭉치인 경우이며, 명사구 보문화란 관형절이 수식하는 명사구가 관형절안의 문법적 관계와는 상관없이 없는 경우를 말한다[9]. 즉, ⑤에서는 관형절 '책을 읽는'의 주어는 '그'이며 이것이 생략되어 수식어로 나타났지만 ⑥에서는 관형절 '그가 책을 읽는'이 모든 문장 성분을 가지고 있으므로 이것이 수식하는 명사 '모습'은 관형절안의 문법적 관계에 영향을 미치지 못한다. 따라서 ⑤의 분석을 위해서는 관형형 술어 '읽는'의 앞의 상태와 뒤의 상태를 살펴 본 후 문장의 문법관계를 결정하도록 해야한다.

우리의 시스템에서 ⑤을 분석하는 단계를 보면, "책을 읽는"에서 "읽는"을 가지고 서술어 앞의 문장 성분을 목적격이 생략된 목적어로 판정한 다음, 관형절 "책을 읽는"이 수식하는 '그'와의 관계를 결정한다. 이때, 관형절의 서술어가 타동사이므로 주어와 목적어가 필요한데, 주어가 생략되었으므로 "그"를 관형절의 주어로 분석한다.

다음에는 ⑥의 문장을 통해 관형절에서의 보문화 현상이 어떻게 분석되는지를 보자. ⑤에서와 같은 방법으로 분석하여 보면 관형절 "그가 책을 읽는"이 완전한 문장 형태를 가지고 있으므로 단지 "모습"을 꾸미는 역할만을 한다. 이후, "그가 책을 읽는 모습"이 "보았다"의 목적어 역할을 하므로 문장상에 남아있는 "나는"은 당연히 "보았다"의 주어로 분석된다. 이 분석을 위한 규칙은 <그림 10>과 같다.

NP -> S NP

S type =C 관계절

case: S cat = 타동사

if S에 주어가 정의되어 있으면

if S에 목적어가 정의되어 있으면

S => 수식어(보문화 현상)

if S에 목적어가 정의되어 있지 않으면

NP => 목적어

if S에 주어가 정의되어 있지 않으면

NP => 주어

case: S cat = 자동사

if S에 주어가 정의되어 있으면

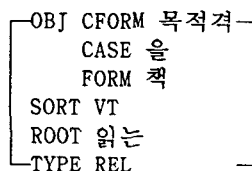
S => 수식어(보문화 현상)

if S에 주어가 정의되어 있지 않으면

NP => 주어

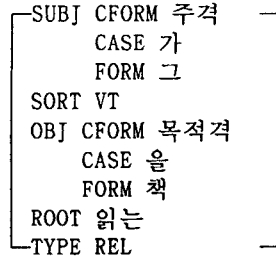
<그림 10> 관형절 처리 규칙

이 규칙을 가지고 문장 ⑥을 분석해 보면, "책을 읽는"은 아래와 같은 자질구조를 가지고,



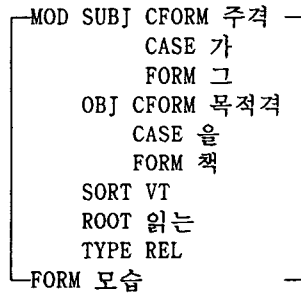
<그림 11> 자질구조 4

이것이 "그가"와 결합하여 "책을 읽는"은



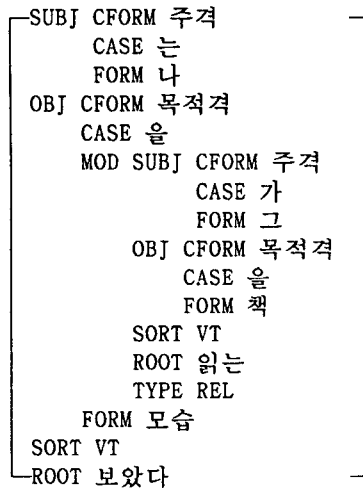
<그림 12> 자질구조 5

이다. <그림 12>과 "모습"이 결합하여 <그림 13>인 "그가 책을 읽는 모습"이 된다.



<그림 13> 자질구조 6

<그림 14>는 목적격 조사 "을"과 결합하여 전체 문장의 서술어 "보았다"의 목적어가 되며, 따라서 "나는"은 주어로 판정되어 최종적으로 "나는 그가 책을 읽는 모습을 보았다"가 되어 옳은 문장으로 분석해 낸다.



<그림 14> 자질구조 7

## V. 결론 및 향후 연구방향

본 논문에서는 문맥자유문법하에서 각 문장소들간의 의존관계를 찾아내기 위해 조건 단일화 기능을 사용하여 우리말 분석을 시도하였다. 이를 위해 디스정션, 동일성제약, 존재여부 검사기능 등을 도입해 PATRII를 확장하였다. 분석 결과 기본적인 문장의 완벽한 분석과 더불어 생략 요소가 있는 문장, 도치문, 관형절등 한국어의 언어적 특성상 난제들이라고 여겨지는 문제들을 포함한 대부분의 우리말 문어체 문장을 분석해 낼 수 있었다. 물론 문법적인 정보만으로 모든 문장을 파악할 수 없었으며 이의 대안으로 의미표지를 도입하여 대부분의 모호성은 제거하였다.

문맥자유문법을 사용하면 문법을 체계적으로 명료하게 기술할 수 있다. 그러나 우리말에서 빈번하게 발생하는 생략, 도치 등의 문장에서 규칙이 복잡해진다는 사실에 대해 가장 큰 비난을 받아왔다. 그렇지만 우리의 시스템에서는 우리말의 특성에 가깝게 문법을 최대한으로 줄이는 방식을 택하였기 때문에 단일화 조건을 잘 조절하여 기술하기만 하면 규칙이 충분히 단순해질 수 있다는 것을 알 수 있었다. 또한 우리의 분석기는 문장의 문장상 문법적 역할을 나머지 성분들과의 의존관계 분석으로 찾아내므로 문장이 복잡해 질수록 잘 분석할 수 있는 특성이 있다. 이상과 같이 확장된 PATRII를 사용하여 우리말 분석을 시도함으로써 기존에 발표된 우리말 분석연구의 통합적인 연구로써 가치가 있다고 생각한다.

앞으로의 연구는 디스정션처리시 현저히 감소하는 시스템 효율을 올리는 문제와 약간의 포말리즘 수정으로 문장상에서 말뭉치가 생략된 경우에도 분석을 해낼 수 있도록 하는 연구가 필요하리라 본다.

## VI 참고 문헌

- [1] Bresnan, J. and R. Kaplan, "Lexical Functional Grammar: A Formal System for Grammatical Representation", The mental Representation of Grammatical Relations, ed. J. Bresnan, MIT Press, 1982.
- [2] Koiti Hasida., "Conditioned Unification for NLP", Coling 86, 85-87
- [3] Lauri Karttunen, D-PATR: A Development Environment for Unification-Based Grammars, CSLI, 1986
- [4] Ralph M. Weischedel and Lance A. Ramshaw, "Reflections on the Knowledge needed to Process Ill-formed Language", Machine Translation, ed. S. Nirenburg, Cambridge University Press, 1987.
- [5] Robert T. Kasper, "Conditional Descriptions in Functional Unification Grammar", ACL 26th, pp233-240, 1988.
- [6] Shieber, S. M., An Introduction to Unification-Based Approaches to Grammar, CSLI Lecture Notes, 1986.
- [7] Tomita M., Efficient Parsing for Natural Language, Kluwer Academic Publishers, 1986.
- [8] Tomita M., Mitamura T., Kee M., The Generalized LR Parser/Compiler: User's Guide, Technical Report, Center for Machine Translation, Carnegie-Mellon University, 1988
- [9] 권재일, 국어의 복합문 연구, 집문당, pp20-25, 1985.

- [10] 권혁철, 최준영, '단일화 기반 의존문법을 이용한 한국어 분석기', 한국정보과학회 논문지, Vol. 19, No. 5, pp467-476, 1992.
- [11] 서영훈, 의미정보를 이용하는 중심어 주도의 한국어 파싱, 서울대 컴퓨터공학과 공학박사 학위논문, 1991.
- [12] 윤덕호, 김영택, '다단계 여과 및 탐색을 이용한 의존문법에 기반을 둔 한국어 분석 알고리즘', 한국정보과학회 논문지, Vol. 19, No. 6, pp614-623, 1992.
- [13] 윤덕호, 김영택, '미지문법관계 속성을 이용한 LFG에서의 한국어 문장분석 연구', 한국정보과학회 논문지, Vol. 16, No. 5, 1988.
- [14] 윤덕호, '성분 결속 도표와 명사 동사 관계표에서의 관계항 삭제를 이용한 한국어 분석 알고리즘', 한남대학교 논문집(자연), Vol 21, pp303-314, 1991.
- [15] 정희성, '한글 구구조문법', 제1회 자연언어처리 워크샵 발표논문집, pp3-37, 1987.
- [16] 조규빈, 하일라이트 고교문법, 지학사, pp54-60, 1986.
- [17] 홍사만, 국어 특수조사론, 학문사, pp11-15, 1983.