확장된 문형 정보를 사용하는 조건 단일화 기반의 한국어 파싱

정회원 김 기 철*

Korean Parsing based on Conditional Unification Using Extended Sentence Patterns Information

Ki-cheol Kim* ***Regular Member***

요 약

영어는 다소 고정 어순의 언어인데 비해 한국어는 부분 자유 어순의 언어이다. 그리고 용언이 의미적 제약을 통해 문장을 지배한다. 또한, 대부분의 한국어 문장은 내포문을 가지는 복문으로 구성되어 있다. 따라서 한국어에 적합한 문법이나 구문제약을 기술하기가 어렵다. 그 결과로 한국어 파싱은 많은 구문 모호성이 발생한다. 따라서 본 논문에서는 기존의 단문 위주의 문형 정보를 복문 구조까지도 표현할 수 있도록 확장하고 이것을 구문 제약 조건으로 기술하기 쉬운 조건 단일화 기반의 CFG 문법을 사용한 한국어 파싱 방안을 제안한다. 제안한 방법으로 2,641개의 한국어 문장을 파싱 했을 때, 구문 모호성이 84.81% 감소됨을 보인다.

Key words : Conditional Unification based CFG Grammar, Sentence Patterns Information, Korean Parsing, Syntactic Ambiguity Resolution

ABSTRACT

English is a more or less fixed word order language, while Korean is a partially free word order language. And it controls sentences by limiting the meanings of the predicate. In addition, Most of Korean sentences are complex sentences which are consisted of embedded clause. Therefore it is difficult to describe appropriate grammar or syntactic constraint for the Korean. So, Korean parsing has a lot of syntactic ambiguity. In this paper, we extend a Sentence Patterns Information(SPI) to include structure of complex sentences. And, we propose Korean parsing method using conditional unification based Context Free Grammar(CFG) that is easy to describe extend SPI as syntactic constraint conditions. By empirical results of parsing 2,641 sentences, we found that our method decreases 84.81% of syntactic ambiguities.

Ⅰ. 서 론

파싱은 문장의 중심이 되는 용언과 나머지 단어와의 문법적 관계를 밝히는 작업이지만 구문 모호성을 해결하는 것이 보다 중요한 일이다. 구문 모호성을 야기하는 요인은 다양하지만 그 중에서도 형태소 분석 결과의 과생성, 품사의 다양성, 명사구들의 나열, 체언구나 부사구가 용언과의 결합에서 발생되는 구 부착의 문제, 내포문을 가지는 복합문의 구조에서 절의 범위에 의한 모호성이 대부분을 차지한다. 예로 다음의 관형절 문장을 살펴보자.

가) **[철수가** [*가끔* [영수와 [집에서 [**싸우는**]]]]]

※ 본 논문은 2008년도 원광보건대학 교내 연구비 지원에 의하여 수행되었습니다.
* 원광보건대학 유아교육과(kckim@wkhc.ac.kr)
논문번호 : 10046-1128, 투고일자 : 2010년 11월 28일

영희를 **본다**.

문형) 가다 : N이 N에 V, N이 N로 V, N이 V
보다 : N이 N을 V

문장 가)에서 체언구 "철수가"와 "영수와", "집에서", "영희를"은 모두 용언 "싸우는"이나 "본다"와 부착할 수가 있기 때문에 체언구 부착 모호성이 발생한다. 또한, 용언을 수식하는 기능을 가지는 부사구 "가끔"도 용언 "싸우는"이나 "보다" 모두를 수식할 수 있어서 부사구 부착 모호성이 발생한다. 이들 구 부착에 의한 모호성은 전체 구문 모호성의 26.2% 정도를 차지한다[1]. 아울러, 관형형 어미로 끝나는 용언 "싸우는"의 범위를 어떻게 보느냐에 따라서도 많은 구문 모호성이 발생한다[2]. 실제로 기존의 구문 분석 방법에 따라 가) 문장의 구문 분석을 수행하면 그림 1과 같이 5개의 구문(파스) 트리가 생성된다.

이러한 구문 모호성들은 구문 분석 단계의 시간/공간 복잡도를 증가시키며, 불필요한 분석 후보를 만들어 내어 의미 분석이나 다른 자연어 응용 시스템을 개발하는데 많은 영향을 미친다[2,3]. 따라서 이를 해결하기 위한 많은 연구가 진행되어 왔다. 그림 1에서 보는 바와 같이 체언구 부착의 모호성은 관형절의 수식 범위를 결정하면 해결할 수 있고 부사구도 가까운 용언에 부착하도록 관형절의 수식 범위를 확장하면 자연스럽게 모호성이 해결된다.

본 논문에서는 복문의 구조에서 발생되는 관형절의 수식 범위를 결정하기 위한 방안으로 단문 위주의 문형 정보를 복문 구조까지 표현할 수 있도록 확장한다. 그리고 확장된 문형 정보를 구문 제약으로 사용하기 쉬운 조건 단일화 기반의 CFG 문법을 이용한 한국어 파싱 방법을 제안한다. 이를 위해서는 다양한 유형의 한국어 문장을 분석해서 규칙을

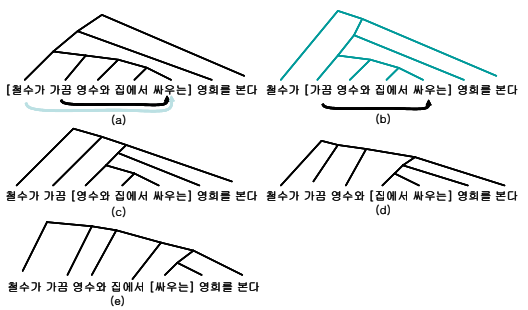

그림 1. 기존 방법에 의한 구문 모호성

찾아내고 이를 문형과 휴리스틱 정보로 정리한다. 또한 정리된 문형을 가지고 한국어 파싱에 적합한 구문 분석용 문법을 작성한다. 이를 토대로 한국어 파싱에서 많이 발생되는 구문 모호성을 감소시켜 정보 검색이나 기계 번역과 같은 응용 분야에 적용이 가능한 실용적인 한국어 파싱 시스템을 구현한다.

Ⅱ. 관련연구

기존의 파싱 방법론은 서양언어의 분석 틀을 이용하여 한국어를 분석하려고 하였다. 의존문법[4]은 생략이 빈번하고 어순이 자유로운 특성을 반영하기 위해 개발되었다. 그러나 구구조 정보에 의해 간단히 제거될 수 있는 파스들도 분석의 대상이 되기 때문에 너무 많은 파스트리가 생성된다. 또한, 단일화 기반의 문법이론[5]은 비문(ungrammatical sentence)을 가려내기 위하여 각종 제약 조건들을 설정하여 분석하는 방법이다. 그러나 어순이 자유롭고 의미가 중요한 역할을 하는 한국어 분석에 직접 적용하기는 어렵다.

따라서 최근에는 다양한 지식을 이용하여 모호성을 해결하는 연구가 주류를 이루고 있다. 다양한 지식으로는 확률 정보[6]나 중심어간의 공기정보[7], 부분적인 어절 결합[8], 최장 묶음[3,9], 구간 분할[10], 문형 정보[2,11] 등의 정보를 이용하여 부분적으로나마 구문 모호성을 해결하고자 하였다. 그러나 현재까지도 한국어 파싱의 표준이 될 수 있는 방법론이 대두되고 있지 않은 실정이다.

확률 정보나 중심어간의 공기 정보와 하위 범주화 정보를 이용하여 구문 모호성을 해결하는 연구는 좋은 방법이지만 대상으로 하는 원시 말뭉치나 태깅된 학습 결과가 필요하며 이를 구축하기가 어려울 뿐만 아니라 심각한 자료 부족 문제를 야기한다. 또한 최장 묶음이나 부분적인 어절 결합은 기계 번역 시스템에서 사용하기 위해 구 단위로 묶어 처리하는 방법으로 구문 분석이라기보다는 형태소 분석 후처리나 구문 분석의 전처리로 보아야 한다.

본 연구에서는 한국어의 언어적 특성상 분석 중간에 문장 성분 간의 관계를 검사하는 동적인 단일화가 선택적으로 이루어지기 때문에 문법 기술 수준에서 이러한 선택적 단일화를 지원하는 조건 단일화를 사용한다. 조건 단일화는 규칙의 명세를 'IF-THEN-ELSE' 형태로 기술하여 자질 구조 안의 특정 속성 값을 필요한 시점에서 검사함으로써 각 문장 성분이 가지는 문법적 관계를 검사하여 조건

에 맞는 단일화만을 수행한다. 이 방법은 의존 문법에서 표현되는 의존 관계를 동적으로 제어할 수 있을 뿐만 아니라 도메인이 변경되어도 문법의 수정만으로 쉽게 적응할 수 있기 때문이다. 따라서 본 논문에서는 기존의 PATRII를 확장한 조건 단일화 기반[5]의 문법으로 한국어 문장의 구구조를 기술하고 확장된 문형과 휴리스틱 정보는 단일화 식을 이용하여 기술한다.

Ⅲ. 구문 모호성 해결

3.1. 휴리스틱 정보의 활용

한국어의 구문 모호성을 야기하는 대표적인 형태가 복합 명사구이다[1,5,8]. 이 복합 명사구는 발생 빈도가 매우 높기 때문에 처리하지 않으면 분석을 원활하게 진행할 수 없다. 따라서 (1)과 (2)의 유형은 조사가 붙는 단어를 만날 때까지 최대한으로 묶어 하나의 명사구로 처리한다. 그러나 (3)의 유형은 정확한 의미 정보의 개입 없이는 해결하기가 어려운 경우가 많기 때문에 수식 관계로만 설정하고 분석을 한다.

(1) 명사가 나열된 형태
예: 정보 검색 시스템
(2) 조사가 생략됨으로써 발생되는 명사구
예: 철수 얼굴 <== 철수(의) 얼굴
(3) 접속조사 “와/과/또는” 등의 문장소에 의해 접속된 명사구
예: ((LG 냉장고)와 (삼성 냉장고))

구문 모호성을 야기하는 또 다른 유형은 관형어의 수식 범위이다. 관형사는 명사나 명사구를 수식하는데 수식을 받는 명사구 안에 다른 수식구가 존재하는 문장에서 모호성이 발생한다. (4)는 가장 가까운 명사를 수식하도록 하며 (5)는 뒤에서부터 수식하도록 하고 (6)의 유형은 의존 명사는 관형어의 수식을 받지 않은 상태에서 다른 명사를 수식할 수 없는 특징[12]를 제약 조건으로 사용하여 해결한다.

(4) 관형사: ((그 지진)의 강도)
(5) 명사 + 관형격 조사: (궁전의 (공주의 방))
(6) 의존명사 + 관형격 조사: ((색다른 곳)의 느낌)

(1)에서 (6)까지의 방법은 항상 정확한 결과를 가진다고 보장할 수 없다. 단지 본 논문에서는 문형 정보를 구문 제약으로 하는 조건단일화 기반으로 한국어를 파싱하는데 초점을 맞추었기 때문에 실험 환경을 위해 제약 조건으로만 사용한다. 이외에 한국어를 파싱하기 위한 휴리스틱 정보는 [2]를 사용하며 일부를 보이면 다음 표 1과 같다.

표 1. 휴리스틱 정보

휴리스틱	설 명
h1	문두의 부사나 문장 부사는 문장 전체를 수식
h2	문장 속에 있는 부사는 가까운 거리에 있는 용언을 수식
h3	가까운 거리에 있는 단어끼리 묶인다.
h4	관형절에서는 하나의 필수 성분이 빠지므로 뒤의 체언구를 반드시 포함
h5	이중주어 문장은 무조건 보조사가 주어이나 조사의 형태가 같으면 앞에 것이 주어
h6	문두의 ’은/는/ㄴ’은 본용언과 호응관계를 가짐
h7	부사절에서는 주어가 생략
h8	인용문이 명령형과 청유형이면 주어가 생략
h9	문형 중 N 이 S 기 V 에서는 S 의 주어가 생략
h10	관형사와 관형격 조사 ’의’가 같이 쓰이면 관형사가 우선 예) 그 지진의 강도 ==> [그 지진의] 강도
h11	관형격 조사가 여러 개 사용된 경우는 뒤의 관형격 조사가 우선 예) 궁전의 인어공주의 방 ==> 궁전의 [인어공주의 방]
h12	관형격 조사와 용언의 관형형이 연속되면 관형격 조사가 우선 예) 잠자는 공주의 아름다운 모습 ==> [잠자는 공주의] 아름다운 모습
h13	의존 명사는 관형어의 수식을 받지 않은 상태에서 다른 명사를 수식 불가 예) 색다른 곳의 느낌 ==> [색다른 곳의] 느낌

3.2 한국어 문형

한국어는 생략이 자주 발생하고 자유 어순을 갖는 비구조적 언어이다. 또한 용언에 따라 다양한 격조사를 요구한다. 따라서 문장의 구조를 파악하기 위해서는 정형화된 구문 정보만을 이용할 수 없다. 예를 들어 다음의 문장을 살펴보자.

1) 철수가 순이를 대표로 뽑다.
2) 철수가 순이를 *대표에게* 뽑다.*

1)과 2)에서 '뽑다'는 "~로"라는 조사가 올 수 있지만 "~에게"라는 격조사는 타당하지 않다. 이러한 현상은 '뽑다'라는 용언에 한정된 것이 아니다. 이와 같이 한국어는 특별한 격을 수반하는 용언이 많이 존재한다. 이러한 용언의 경우 나머지 격을 보조적인 의미로 파악하기 때문에 문장의 올바른 의미를 파악하기 어렵거나 모호성 발생의 원인이 된다. 따라서 이러한 용언들의 구조적 유형을 어떤 틀로 제약할 필요가 있다. 대표적인 예로는 격틀과 문형이 있는데 격틀은 용언에 대한 정확한 의미지식을 요구하기 때문에 정확한 구문 분석은 가능하지만 구축이 어려운 실정이다. 반면에 문형은 순수한 구문적 정보만을 이용하며 약간의 의미적 제약을 가할 수 있기 때문에 본 논문에서는 문형을 이용하여 구문 분석을 수행한다. 한국어의 경우, 구문 분석에서 이러한 문형 정보의 이용은 필수적이라고 여겨진다[2,11].

3.3 조건 단일화 기반 문법

구문 모호성을 해결하기 위한 파싱 문법은 조건 단일화 기반의 CFG 문법을 이용한다. 한국어에서 발생하는 문맥 자유적 특징인 생략과 도치 현상, 부분 자유 어순 등의 특성을 표현하기 위해서는 구구조 규칙을 사용한다. 용언의 하위 범주화 정보에 따라 요구되어지는 필수격과 체언구가 결정되는 문맥 의존적 특징을 표현하기 위해서는 조건 단일화식을 이용한다. 이와 같은 두 가지 방법을 문법이라는 하나의 틀로 표현하기 위해서는 표 2와 같은 정형화된 표현 기법을 사용한다.

표 2. 문법 규칙 기술 형태

```
(
    S -> XP S
      (단일화식)
      (단일화식)
      …
       (단일화식)
)
```

위의 표 2에서 문법 규칙의 첫 번째 요소(S)는 LHS(Left-Hand Side)이고 두 번째 요소(->)는 문법 이동의 표기, 세 번째 요소(XP S)는 RHS(Right-Hand Side), 마지막 요소는 단일화 식을 의미한다. 즉, 'S -> XP S'가 언어의 문맥 자유적 특징을 표현하고 구와 구의 결합 규칙이 기술된다. 단일화 식들의 집합은 문맥 의존적 특징을 표현하며 구와 구가 결합될 때 적용되는 제약 조건들을 기술한다.

단일화 식의 내용을 살펴보면 아래 표 3에서 보는 바와 같이 경로 대 경로, 경로와 값, 동일성 제약 등으로 세분화해서 볼 수 있다.

동일성 제약은 경로와 값을 단일화하는 것이 아니다. 즉, 위의 예에서 "(x0 subcat) =c pvgt"의 경우 (x0 subcat)에 'pvgt'라는 값이 반드시 들어 있어야만 단일화에 성공하게 된다. 이 연산은 LFG 등에서 언어정보의 제약을 위해 사용되고 있다. 한국어의 언어적 특성상 분석 중간에 동적인 단일화가 선택적으로 이루어지기 때문에 문법기술 수준에서 이러한 선택적 단일화를 지원할 필요가 있다. 이러한 선택적 단일화를 조건 단일화라 부른다[5].

조건 단일화는 각 규칙에서 감축(reduce)이 일어나기 위해 수행되는 단일화를 선택적으로 수행한다는 개념이다. 즉, 자질 구조 안의 특정 속성 값을 동적으로 검사한 후 각 문장 성분이 가지는 문법적 관계를 검사하여 조건에 맞는 단일화만을 수행한다는 전략이다. 따라서 조건 단일화는 규칙의 명세를 'IF-THEN-ELSE' 형태로 기술하여 자질 구조 안의 특정 속성 값을 필요한 시점에서 검사함으로써 각 문장 성분이 가지는 문법적 관계를 검사하여 조건에 맞는 단일화만을 수행한다. 이러한 방법은 의존문법에서 사용하는 단어간의 의존관계를 동적으로 제어할 수 있는 특징을 가진다.

표 3. 단일화 식의 예

단일화의 예	실제 예
PATH = PATH	(x0 subj) = x1 (x0 co-subj) = (x0 subj)
PATH = ATOMS	(x0 mood) = dec
PATH =c ATOMS	(x0 subcat) =c pvgt

Ⅳ. 조건 단일화 기반의 한국어 파싱

4.1. 문형의 확장

단문 구조를 기반으로 설정된 문형은 "철수가 밥을 먹자고 제안하다"의 문장이나 "철수가 밥을 먹기 시작하다" 등의 구문 분석에서는 모호성이 발생하게 된다. 따라서 이를 수용할 수 있는 문형 정보

를 구축해야만 한다.

1) 철수가 밥을 먹자고 제안하다.
기존 문형) 먹다: N이 N을 V, 제안하다: N이 N을 V
제안하는 문형) 기존 문형 + 제안하다: **N이 S자고 V**

2) 철수가 밥을 먹기 시작하다.
기존 문형) 먹다: N이 N을 V, 시작하다: N이 N을 V
제안하는 문형) 기존 문형 + 시작하다: **N이 S기 V**

따라서 기존의 문형에 대한 연구를 정리하기 위해 [13]과 [14]를 분석하고 [15]의 격틀 정보를 참고한 결과 기존에 사용되던 단문 위주의 문형 외에 새로운 문형이 필요하게 되었다. 즉 기존의 문형은 한국어 문장의 대부분을 차지하는 내포문에 대한 고려가 없었으며 이중 주어 및 이중 목적어에 대한 고려가 미흡하였다. 따라서 본 연구에서는 이들에 대한 고려를 하였으며 기본적으로 3번 이상 출현 빈도를 보인 문장의 구조를 다음 표 4와 같이 새로운 문형으로 만들어 사용한다.

본 논문에서 사용하는 문형은 동사가 31개, 형용사가 8개, 서술격 동사가 5개로 총 44개이다. 실제 문형 사전에는 동사와 형용사에 대해서만 문형을 기술하고 서술격 동사는 기술하지 않았다. 그 이유는 서술격 동사가 "명사+이다"의 형태로 모든 명사에 문형을 수록하는 것은 타당하지 않기 때문이다. 본 논문에서는 조사가 생략된 문장뿐만 아니라 이중 주어 문장 및 이중 목적어 문장과 내포문을 가지는 문장의 구조를 처리할 수 있도록 문형을 설정하였다.

문형을 분류하기 위한 기본 문법 형태소로 기존의 연구에서는 보통 11개를 사용하였지만 본 논문에서는 조사가 생략된 경우와 이중 주어, 이중 목적어인 경우도 고려하였으며 내포문을 나타내는 조사 상당어구인 "기/음"과, "기로", "지/냐/는가/고"등을 추가하여 표 5와 같은 문법 형태소를 사용한다.

문형 정보는 [15]에 수록되어 있는 엔트리를 대상으로 중첩을 포함하여 8,762개의 엔트리(형용사:2,337개, 동사:6,425개)에 대해 문형 사전을 구축하였다. 하나의 엔트리당 평균 문형수는 1,41개(형용사;1.14, 동사:1.68)를 가진다. 가장 많은 문형을 가지는 용언은 "제시하다"로 8개의 문형(V1, V2, V3, V11, V12, V13, V26, V27)을 가지며 형용사의 경우에는 "가깝다", "관계없다", "떳떳하다" 등 16개의 엔트리가 3개씩의 문형을 가진다. 이를 정리하면 표 6과 같다.

문형의 총 갯수는 동사가 10,771개이며 형용사는 2,664개이다. 빈도를 [문형의 수 / 문형의 총 갯수]로 표현할 때 상위 빈도를 가지는 문형의 일부는 뒤의 표 7과 같다.

표 4. 기존 연구에 새롭게 추가된 문형

구분	문형 표기		용언의 예
자동사	V30	N이 N와 N에 V	동승하다, 조인하다
	V31	N이 N에게 N에서 V	지다, 참패하다
타동사	V20	N이 N1을 N2를 V	삼다, 만들다
	V27	N이 N에/에게 N를 N로 V	기재하다, 내려오다
조사 생략	V29	N이 N V	들다, 가다
내포문	V21	N이 S라고 V	생각하다, 추측하다
	V22	N이 S(기/음) V	바라다, 인정하다
	V23	N이 S기로 V	결정하다, 하다
	V24	N이 S기로 N에게/와 V	약속하다, 부탁하다
	V25	N이 S는지/는가 N에게 V	묻다, 질문하다

표 5. 문형 분류를 위해 사용된 문법 형태소

구 분	문 법 형 태 소
조 사	이(가), 을(를), 에(에게), 에서, (으)로, 와(과), 보다
조사 상당어구	를 위해, 에 의해, 에 대해, 라고
내포문 상당어구	기/음, 기로, 지/냐/는가/고

표 6. 구축된 문형에 대한 기본 정보

구분	동사	형용사
구축된 엔트리 수(8,762)	6,425개	2,337개
엔트리당 평균 문형 수(1.41)	1.68개	1.14개
엔트리당 최대 문형 수	8개	3개

표 7. 상위 빈도의 문형정보

동사	빈도	갯수	형용사	빈도	갯수
V1	31.08%	3,348	A1	83.1%	2,213
V11	35.49%	3,823	A5	10.1%	270
V2	12.54%	1,351	A2	4.5%	119
V12	6.0%	647			

4.2 조건 단일화 기반 CFG

조건 단일화는 규칙의 명세를 'IF-Then -Else' 형태로 기술하여 자질 구조 안의 특성 속성 값을 필요한 시점에서 검사함으로써 각 문장 성분이 가지는 문법적 관계를 검사하여 조건에 맞는 단일화만을 수행한다는 전략이다. 이러한 방법은 의

존 문법에서 사용하는 단어 간의 의존 관계를 동적으로 제어하는 특징을 가진다.

본 논문에서는 구문 분석을 위한 기본 틀로 문형을 제약 조건으로 사용할 수 있는 조건 단일화 기반의 PATRII를 이용하였다[5]. 이는 단어들이 결합하여 문장을 이루는 경로(path)로 구구조 규칙을 설정하고, 개개의 구구조들이 결합할 때 조건 단일화 기반의 제약에 의하여 구구조의 결합을 제약하는 방법이다. 이는 구구조 규칙의 간결함과 구구조 의존적 언어의 특성을 조건 단일화 제약을 통해 문장을 분석하는 것이다.

위의 그림 2는 구구조 규칙 "SV -> NP SV"가 적용되기 위해 필요한 제약들을 문형 정보 및 의미 지식을 이용하여 기술한 예를 보여주고 있다. NP는 체언구를 의미하여 SV는 용언이 포함된 문장을 의미한다. "철수가 밥을 먹었다"라는 문장을 분석하기 위해서는 우측의 <NP>에 "철수가"가 할당되며 x1의 값을 가지고 "밥을 먹었다"가 우측의 <SV>가 되며 x2의 값을 가지며 이 값을 이용하여 좌측의 <SV>로 단일화하게 된다. 먼저 x2를 x0로 단일화한 후에 x1(철수가)의 격조사가 주격((x1 jform) =c jcs)이면 아래의 문장으로 진행하고 주격이 아니면 '*or*'의 역할에 의해 '((x1 jform) =c jco)'의 문장

```
(<SV> -> (<NP> <SV>)   ;;; CFG Rule
 ((x0 = x2)            ;;; 단일화 식
   (*or*
    (((x0 subcat) =c pvgi)    ;; 자동사이면
      (*or*
          (((x1 jform) =c jcs)  ;; 주격이면
             (*or*
                (((x0 subj) = *defined*)  ;;이미
주어가 있으면
              ((x0 sp-info) =c (*or* v6 v10)) ;;문형이
v6이나 v10이면
               (*or*
                 (((x0 subj jform) =c jxc)
                  ((x0 comp) = x1))
                    :
          (((x1 jform) =c jco)  ;;; 목적격이면
             (*or*
                (((x0 sp-info) =c v2)   ;;v2 문형이면
                 ((x0 dest) = *undefined*)
                 ((x0 dest) = x1))
                (((x0 sp-info) =c (*or* v26 v27))
;;v26, v27문형이면
                 ((x0 about) = *undefined*)
                 ((x0 about) = x1))
```

그림 2. 문형을 제약조건으로 하는 조건 단일화 기반 CFG의 예

으로 스킵(skip)하여 단일화를 진행한다. '((x0 subj) = *defined*)'의 문장은 "x0(밥을 먹었다)"의 주격(subj)이 이미 정의되어 있는지 체크하여 정의가 되어 있으면 x0의 문형 정보가 v6이나 v10인지를 검사((x0 sp-info) =c (*or* v6 v10))하는 과정을 진행한다. '*or*'의 역할은 격조사가 주격(jcs)일 수도 있고 목적격(jco)일 수도 있는 것처럼 여러 항목의 조건을 검사하기 위해 사용된다.

한국어를 구문 분석하기 위해 작성된 문법 규칙은 112개이나 하위 정보에 의해 분류되어 중복되는 규칙을 제거하면 총 89개이다. 중복이 발생하는 규칙은 대부분 체언구(NP)가 동사구(SV), 형용사구(SA), 명사술어구(SN)와 단일화하여 동사구가 되는 경우와 체언구가 동사관형구(VNP), 형용사관형구(VNP), 명사술어관형구(NNP)와 단일화하여 각각의 관형구를 생성할 때 즉, "SV -> NP SV"나 "ANP -> NP ANP"와 같이 "NP"의 격조사에 따라 다양한 처리가 이루어져야 하는 구구조 규칙에서 발생하는데 하나의 문법으로 기술하면 문법의 크기가 커질 뿐만 아니라 기술하기도 어렵고 문법을 이해하기도 어려워서 격조사 별로 문법 규칙을 작성하였다. 체언구에 대한 문법 규칙의 수가 가장 많은 14개였다.

4.3 PATRII의 확장

본 논문에서는 PATRII를 이용하여 위의 그림 2와 같은 CFG 기반의 문법을 작성하고 이를 구문 분석을 위한 LR 파싱 테이블과 조건 제약을 위한 함수로 번역하여 LR 파서 기반의 구문 분석기를 이용하고 있다. 본 연구에서는 문형을 이용하여 문맥 의존적인 특징을 반영하고 이러한 검사를 조건 단일화라는 제약 조건을 통해서 행할 수 있도록 단일화 식을 작성한다. 따라서 기존의 PARTII 문법에서는 무조건적인 순수 단일화 이론만을 택하고 있기 때문에 [5]가 제안한 확장된 PARTII를 사용하여 문법을 기술한다.

그러나 기존의 확장된 PATRII 문법에 사용된 연산자만으로는 한국어의 다양성을 모두 적용하는데 한계가 있어 조건 체크 명령어를 추가하여 사용한다.

4.3.1 멤버 여부 검사 기능

이 기능은 특정 속성의 값이 멤버로 속하는지의 여부를 동적으로 검사하기 위해 사용된다. 이 기능을 위한 기호로는'*MEMBER-NOT*'을 사용한다. 이 연산은 용언에 따라 다양한 문형 정보를 가지며

하나의 용언이 여러 개의 문형 정보를 가지는 한국어의 문장 분석에 특히 필요하다. 예를 들면 '((X1 SP-INFO) =C (*MEMBER-NOT* V4 V10))'이라는 단일화 문은 'X1'의 문형 정보를 나타내는 V4, V10형이 아니면 단일화를 수행하라는 의미이다. 만일 문형 정보가 V4인 경우도 조건식이 거짓이 되며 문형 정보에 V4와 V10이 모두 없을 경우에만 참이 된다.

(S -> (XP S)
((*OR*
(((X2 SP-INFO) =C (***MEMBER-NOT*** V4 V10))
(서술어가 V4, V10 문형을 가지지 않을 때의 단일화 집합))
(((X2 SP-INFO) =C (*OR* V4 V10))
(서술어가 V4, V10 문형을 가질 때 처리하는 단일화 집합))
)))

4.3.2 명세의 선택적 이접 기술(disjunctive description) 기능

선택적 이접 기술의 기호로 '*EOR*'를 쓴다. '*EOR*'로 시작하는 명세에는 대등한 위치를 점하는 명령문을 나열하여 쓸 수 있다. 이 기능은 기존의 *OR*가 중첩 IF문의 구실을 하며 여러 명령문을 수행하도록 기술되어 있던 것에서 조건을 만족하는 하나의 명령문만을 먼저 수행하고 나머지는 건너뛰는 기능을 수행한다. 예를 들어 다음과 같이 명세를 기술하면 S(X2)에 속해 있는 서술어의 문법범주 속성 CAT의 값이 'PVGI'이면 자동사에 관련된 처리를 수행하고 끝내며 CAT의 값이 'PVGT'이면 타동사에 관련된 단일화를 수행하고 끝낸다는 의미이다. 그러나 이것을 *OR*로 기술하면 자/타동사인 경우에는 자동사의 처리와 타동사의 처리를 모두 수행하게 된다.

(S -> (XP S)
((***EOR***
(((X2 CAT) =C PVGI)
(서술어가 **자동사**일 때 처리하는 단일화 집합))
(((X2 CAT) =C PVGT)
(서술어가 **타동사**일 때 처리하는 단일화 집합))
)))

Ⅴ. 문형정보를 이용한 문장 파싱

영어의 경우, 기본적 문형 구조는 어순이 존재하기 때문에 NP의 역할이 위치에 따라 결정된다. 그러나 한국어의 경우는 어순이 비교적 자유롭고 생략이 빈번하다. 또한 격조사와 부사구가 발달되어 있다. 따라서 구문 분석이 어려울 뿐만 아니라 구문 모호성이 많이 발생한다.

본 논문에서는 이러한 구문 모호성을 해결하기 위하여 확장된 문형 정보와 휴리스틱 정보 를 이용하며 문형으로 해결할 수 없는 경우는 의미지표를 사용한다. 문형은 용언에 따른 NP의 문법 형태소 정보를 파악하기 위한 제약으로 사용한다. 의미지표는 문형에 대한 제약으로 사용한다. 따라서 본 논문에서는 한국어를 파싱하기 위하여 조건 단일화 기반의 CFG 문법을 기본 틀로 간주하고 문형과 의미지표를 제약조건으로 사용하여 구문 모호성을 해결할 수 있음을 보인다.

NP ADV NP NP PVM NP VP

나) 철수가 (**자주** 영수와 *학교에서* 싸우는 영희를) 보다.

문형) 싸우다 : N이 N와 V

보다 : N이 N을 V, **N이 N을 N로 V**

기존의 구문 분석 방법은 좌에서 우로 분석을 시도하며 공동격 조사 "와"에 대한 처리를 고려하면 예문 나)는 다음과 같이 7가지의 분석 결과를 가질 수가 있다. 본 논문에서 제안한 방법을 사용하면 구문 모호성이 발생하지 않을 뿐만 아니라 공동격 조사 "와"에 의한 모호성도 해결됨을 알 수 있다. 위의 예문 나)만을 구문 분석하기 위해 본 논문에서 사용하는 문법 규칙은 그림 3과 같다.

또한, 문형은 부사구 부착 문제를 해결할 수 있으며 공동격 조사 '와'에 의한 모호성의 일부도 해결할 수가 있다. 예를 들어 예문 3)은 부사구 부착과 관련된 모호성이 야기된다. 부사구 "학교에"는 "가는"과 "보았다"라는 용언을 수식할 수가 있다. 그러나 용언 "가다"는 "N이 N에 V"라는 문형이고 "보다"는 "N이 N을 V" 문형이다. 따라서 "학교에"는 "N이 N에 V" 문형을 가지는 '가다'와 결합한다는 것을 알 수가 있다.

예문 4)와 5)는 공동격 조사에 의해 모호성이 발생한다. 공동격 조사는 "A와 B"의 구조를 갖는다. 따라서 이 정보만을 활용하면 예문4)는 'B'에 해당하는 명사가 없으므로 구문분석에 실패하고 5)는 구문 분석이 성공적으로 일어난다. 그러나 예문 4)는 올바른 문장이다. 따라서 본 논문에서는 이런 문제를 해결하기 위해서 문형 정보를 이용한다. 즉, "싸우다"의 문형인 "N이 N와 V"를 이용하면 예문 4)는 올바르게 파싱이 된다.

4) 철수가 영수와 싸웠다.
5) 철수가 영수와 빵을 먹었다

```
■ MA <- ADV          // 부사  : 가끔
■ VNP <- PVM NP      // 관형형 체언구  : 싸우는 영희를
■ VNP <- NP VNP      // 체언구 관형구
■ VNP <- MA VNP      //  부사  관형구
■ NP <- VNP          //  관형구     단문 분할
■ SV <- NP VP        //  체언구와  동사
■ SV <- NP SV        //  체언구와  동사구
■ S <- SV            // " 철수가 []를  본다"를  문장으로
```

그림 3. 예문 나)를 구문 분석하기 위한 문법 규칙

Ⅵ. 실험 및 평가

6.1 실험 및 분석

실험을 하기 위한 시스템 구성은 SUN SPARC Station1+ 에서 COMMON LISP를 이용하여 구현하였다. 문법 번역기는 확장된 PATRII 문법이 기술된(*.gra) 파일을 입력받아 단일화식들을 번역한 함수를 가지는 파일(*.fun)과 번역된 함수들과 문맥자유 문법을 연결하는 정보를 가지는 파일(*.info)을 생성해 낸다. 구문 분석 엔진으로는 다중 경로 기반의 GLR 구문 분석기를 사용하였다. 따라서 LR 테이블 생성기가 문법 파일(*.gra)안의 각 규칙들 중에서 구문 분석기가 참조할 LR 테이블을 .LR 파일을 생성해 낸다. 또한, 구문 분석기의 입력이 되는 형태소 분석기는 구문 형태소 단위의 [4]를 사용하였다.

실험은 내포문 분할이 가능한 10 어절 이내의 2,641 문장을 가지고 수행하였다. 이를 위해 국어정보베이스[13]에서는 2,153 문장을 추출하고 초등학교 사회 교과서[14]에서는 488 문장을 추출하였다. 추출된 문장은 평균 3.17개의 용언(보조 용언은 제외)을 포함하는 복문의 구조를 가졌다. 구문 모호성을 제약하기 위해서 다음과 같은 2가지 방법으로 실험을 하였으며, 각각의 실험 결과에 따른 평균 구문 트리의 수(구문 모호성의 수)는 표 8과 같았다. 실험1에 비해 본 논문에서 제안한 방법은 구문 모호성의 수가 84.81%로 감소하였다.

실험1 : 일반적인 구문 분석 방법[5]를 사용
실험2 : 확장된 문형과 휴리스틱 정보 사용

표 8. 실험에 따른 평균 구문 모호성의 수

실험 문장	평균 용언 수	실험1	실험2
KIBS(2,153)	3.69	65.21	6.82
사회교과(488)	2.65	52.35	7.04
평 균	3.17	58.78	6.93

6.2 실험 및 평가

실험에 따른 결과를 분석하면 평균적으로 사용된 용언 수가 적은 사회교과서가 KIBS보다 구문 모호성이 더 많이 발생하였다. 그 이유는 KIBS에 비해 사회교과서는 명사들의 나열이나 보조사가 많이 사용되었으며 이로 인해서 명사구의 범위에 의한 모호성과 보조사에 의한 불확실한 격 정보 때문에 많은 모호성이 발생하게 되었다. 예를 들어 "조선의 궁궐과 가옥", "조선의 역사와 현재"는 같은 구조로 되어 있지만 "조선의 [궁궐과 가옥]"과 "[조선의 역사와] 현재"와 같이 분석 결과가 다른 구조로 되는 경우가 많아 최상위 결과로 정확하게 분석하는데 어려움이 있었다. 또한, KIBS에서도 신문이나 교과서보다는 소설에서 발췌한 문장에서 오류가 많았다. 이는 소설의 경우 어순의 도치와 생략이 심하고 "철수의 손잡이가 달린 가방"과 같이 의미 정보를 요하는 문장이 많았기 때문이다.

그러나 본 논문에서 제안한 방법으로 구문 분석을 수행하면 구문 모호성의 수가 평균 58.78개에서 6.93개로 84.81%나 줄어들었다. 이는 테스트 문장이 대체로 10어절 내외의 문장임에도 불구하고 문형을 제약조건으로 한 내포문 분할 방법이 한국어의 구문 분석에 매우 효율적임을 보여준다. 실험에 소요된 시간은 실험1이 평균 2.05초인데 반해서 실험2는 1.06초 소요되었다. 따라서 기존의 구문 분석 결과보다는 본 논문에서 제안한 시스템에 의한 구문 분석 결과가 정보 검색이나 기계 번역, 자연어 이해와 같은 응용 시스템에 보다 효율적으로 적용할 수 있다.

표 9. 실험2에 의해 생성된 최상위 구문 트리 결과의 정확도

실험 문장	평균 구문 트리 수	오분석 수	정확도
KIBS(2,153)	6.82	38(4)	98.24%
사회(488)	7.04	9(1)	98.16%
평 균	6.93	47(5)	98.20%

또한 실험 결과 중에서 47개의 분석에서 오분석이 발생했다. 이 중에서 41개는 최상위 결과로 분석되지 않은 경우이며 실제로 올바른 후보가 생성되지 않은 경우는 5문장이었다.

오분석된 경우를 예로 들면 다음과 같다. “아름답다”의 문형 정보만을 이용하면 아래의 2)문장은 1)문장과 같이 a)로 분석된다. 그러나 의미적으로는 b)가 타당하다. 이 문제를 해결하기 위해서는 “N이 N이 아름답다”라는 새로운 문형 정보를 추가하는 것을 고려해 볼 수가 있다. 그러나 이 경우에는 1) 문장이 b)처럼 해석되는 구문 모호성을 초래한다. 이와 같이 문형을 확대하면 오분석 되는 경우는 줄어들겠지만 구문 모호성의 수는 증가하게 된다. 따라서 이를 해결하기 위한 새로운 제약이 필요하다. 본 연구에서는 공기 정보를 이용하여 이를 해결하지만 차후에는 의미지식을 이용한 의미처리를 도입하고자 한다.

1) 철수가 [아름다운 공원을] 보았다.
2) 꽃이 아름다운 공원을 보았다.
  a) 꽃이 [아름다운 공원을] 보았다.
  b) [꽃이 아름다운 공원을] 보았다.

문형) 아름답다 <-> N1이 아름답다
보다 <-> N1이 N2를 보다

Ⅶ. 결 론

문장의 구조와 문형 정보는 구구조 규칙으로 기술하고 문형 정보나 휴리스틱에 대한 제약은 단일화 식으로 기술하기 위해 조건 단일화 기반의 CFG 문법을 작성하여 구문 분석을 수행하였다. 문형 정보를 단일화 식에서 평가하기 위해 기존의 PATRII에 ‘*MEMBER-NOT*’과 ‘*EOR*’연산자를 추가하였다. 이 방법은 구문 분석기 내부에서 분석을 제어하는 것이 아니라 문법 기술 수준에서 분석을 제어하는 것이다. 따라서 도메인(Domain)이 변경되면 구문 분석기 자체를 수정하는 것이 아니라 구문 분석 문법만을 수정해서 여러 분야에 활용이 가능한 구문 분석기를 쉽게 구현할 수 있는 장점을 가진다.

제안한 방법으로 10어절 이내의 2,153 문장을 구문 분석한 결과 평균 84.81%의 구문 모호성이 감소되었으며 98.20%의 문장이 정확한 결과를 포함하였다. 이는 일본어와 같이 문법을 기술하기 어려운 언어라도 문형만 파악된다면 효율적인 구문 분석이 가능함을 의미한다.

향후 연구 과제로는 지금까지 개발된 문형 정보가 문어체와 구어체 모두를 포함하고 있지만 주로 정문 중심의 문어체 위주로 구축되었다. 따라서 생략과 도치가 더 심한 구어체 위주의 문형에 대한 연구가 필요하다.

참 고 문 헌

〔1〕 김재훈, 서정연, 김길창(1992), “구문 그래프를 이용한 구문적 애매성 분석,” 제4회 한글 및 한국어 정보처리 학술대회 논문집, pp. 159-167.

〔2〕 이현영, 이용석(2008), “내포문의 단문 분할을 이용한 한국어 구문 분석,” 한국정보과학회 논문지, Vol. 27, No. 7, pp. 50-58.

〔3〕 황이규(2001), 구문 형태소를 이용한 형태소 및 구문 모호성 축소, 전북대학교 박사학위 논문.

〔4〕 윤덕호, 김영택(1992), “다단계 여과 및 탐색을 이용한 의존문법에 기반을 둔 한국어 분석 알고리즘”, 한국정보과학회 논문지, Vol. 19, No. 6, pp. 614-624.

〔5〕 양승원, 박영진, 이용석(1995), “조건 단일화 기반 PATRII를 이용한 한국어 구문 분석”, 한국정보과학회 논문지 Vol. 22, No. 4, pp. 653-662.

〔6〕 박소영, 김수홍, 임해창(2004), “문장성분의 다양한 자질을 이용한 한국어 구문분석 모델,” 한국정보처리학회논문지, 제11-B권 제6호, pp.743-748.

〔7〕 이공주, 김재훈(2002), “중심어 간의 공기정보를 이용한 한국어 확률 구문분석 모델,”한국정보처리학회논문지, 제9-B권, 제6호, pp.809-815.

〔8〕 김창제, 정천영, 김영훈, 서영훈(1995), “부분적인 어절 결합을 이용한 효율적인 한국어 구문 분석기”, 정보과학회 가을 학술 발표논문집, pp.597-600.

〔9〕 박상규, 정창민, 조준모, 이상조(1995), “최장 묶음을 이용한 효과적인 한국어 구문 분석기,” 정보과학회 봄 학술 발표논문집, pp.961-964.

〔10〕 김광백, 박의규, 나동렬, 윤준태(2002), “구간 분

할 기반 한국어 구문 분석," 제 14회 한글 및 한국어 정보처리 학술대회, pp.163-168.
〔11〕 강은국(1996), 조선어 문형 연구, 박이정출판사.
〔12〕 장석진(1993), 정보기반 한국어 문법, 도서출판 언어와 정보.
〔13〕 KIBS : Korean Information Base System, http://kibs.kaist.ac.kr/kibs
〔14〕 교육부(1995), 사회 5-1, 국정교과서주식회사.
〔15〕 연세대학교 언어정보개발원(1999), 연세한국어 사전, 두산동아.

김 기 철 (Gi-cheol Kim) 정회원
한국통신학회 논문지 제 28권 2T호 참조