

한국어의 장거리 의존 관계를 구현하기 위한 최소주의 문법 모형*

윤지원 **

- 차 례 -

1. 서론
2. 최소주의 전산문법
3. 기본 구문 분석
 - 3.1. 논항
 - 3.2. 술어
4. 장거리 의존 관계 분석: 한국어의 관계절
 - 4.1. 부가 분석
 - 4.2. 인상 분석
5. 활용 방안
6. 결론

1. 서론

본 논문은 한국어 구문 분석(parsing)을 위한 전산문법(computational grammar)로 최소주의 문법(Minimalist Grammar; Stabler 1997, 2011)을 소개하고, 이를 통해 한국어의 기본적인 통사 구조, 특히 장거리 의존 관계(long-distance dependency)를 전산적으로 구현하는 방법을 제시하고자 한다.

언어학 제 87 호 (2020. 8. 30: 63-86), 사단법인 한국언어학회

* 논문을 자세히 읽고 통사이론의 적용에 관해 많은 지적과 의견을 주신 허세문 선생님과, 전산언어학과 통사론 두 측면 모두 유익한 비평과 제안을 해주신 익명의 심사위원 선생님들께 감사드립니다.

** 스토니브룩 대학교 언어학과 조교수 (jiwon.yun@stonybrook.edu)

Chomsky (1995)가 발표한 최소주의 프로그램(Minimalist Program)은 현대 언어학의 주류를 이루는 언어 분석 방법론 중의 하나이며 이에 기반한 통사 이론을 통해 많은 연구 성과가 이루어져 오고 있다(고희정 2018 참조). 그럼에도 불구하고 전산 언어 처리에서의 구문 분석에 이용되는 문법으로서는 최소주의 통사론이 그다지 널리 쓰이지 못하였다. 이론언어학의 발전으로부터 얻을 수 있는 많은 성과와 언어 구조에 대한 통찰이 전산언어학 분야에서 활용되지 못한다면 큰 손실이 아닐 수 없다.

전산언어학에서 보다 널리 쓰여온 통사 이론으로는 HPSG (Head-driven Phrase Structure Grammar)로 대표되는 제약 기반 이론(Constraint-based grammar)을 들 수 있다(Kim et al. 2011 참조). 제약 기반 이론과 최소주의 이론은 언어의 생성성을 가정한다는 점에서는 공통점이 있으나, 최소주의 이론과는 달리 제약 기반 이론에는 언어 구조의 변형, 즉 통사적 이동을 나타내는 규칙이 존재하지 않는다. 이는 구조의 변형을 전산적으로 구현할 수 있는 언어처리 모형이나 분석기를 구축하는 것이 어렵다는 시각과 이어져있다(김종복·양재형 2003: p.3).

Stabler (1997, 2011)가 제안하는 최소주의 문법(Minimalist Grammar: MG)은 Chomsky (1995)의 최소주의 프로그램에 따른 통사 이론을 전산적으로 구현할 수 있는 방법을 제공한다. 이는 병합(merge)과 이동(move) 등의 연산을 문법 요소로 직접 포함하고 있어서 언어의 생성성뿐만 아니라 구조의 변형을 구현하기에 적합한 문법이라고 할 수 있다. 본고에서는 통사론에서 가리키는 최소주의 문법 이론과의 혼동을 피하기 위해 최소주의 “전산문법”이라고 표기하도록 하고, 이를 문법 기술에 도입하여 활용하려는 연구자들에게 도움이 되도록 간단한 한국어 예시와 함께 단계별 도출 과정을 상세히 소개한다. 특히 인지언어학에서 매우 중요한 주제 중의 하나인 장거리 의존관계를 분석할 수 있는 전산문법을 제시하고자 한다. 장거리 의존 관계에 대한 분석은 통사적 이동을 가정하느냐 아니냐에 따라, 또한 어떤 종류의 이동을 가정하느냐에 따라 상이한 결과가 나올 수 있으므로 이동을 구현할 수 있는 전산문법을 이용하면 각 통사 분석의 이해도를 높이고 적합성을 판정하는 데 도움이 될 수 있다. 마지막으로 최소주의 전산문법의 보다 다양한 활용 방안과 앞으로의 연구 과제를 논의하고 끝맺는다.

2. 최소주의 전산문법(Minimalist Grammar: MG)

이 장에서는 Stabler가 제안한 최소주의 전산문법(Minimalist Grammar: MG)의

핵심 구성요소들을 Stabler (2011)을 중심으로 소개하고자 한다. 이에 따르면 한 언어의 최소주의 전산문법을 구현한다는 것은 그 언어의 어휘부(lexicon), 즉 어휘소(lexical item)의 유한한 집합을 정의하는 것이며, 각 어휘소는 통사적 자질들과 음성/의미 등 비통사적 자질로 이루어져 있다. 어휘소를 표기하기 위해서는 다음과 같이 어휘소가 갖는 비통사적 자질을 먼저 표기하고 그 어휘소의 통사적 자질들을 이중 쌍점(::) 뒤에 표기한다.

(1) 어휘소 (Stabler 2011:p.619)

[음성/의미 자질] :: [통사 자질1 통사 자질2 ... 통사 자질N]

통사적 자질은 다음의 네 가지 범주로 나뉜다.

(2) 통사 범주 (Stabler 2011:p.620; cf. Stabler 1997:p.73)

- a. 기본 범주(base): X
- b. 선택 범주(select): =X
- c. 자질 부여 범주(licensors): +x
- d. 자질 요구 범주(licensees): -x

우선 기본 범주(base)는 명사(N), 동사(V), 형용사(A), 한정사(D), 시제(T), 보문소(C) 등 최소주의 통사론에서 기본 단위로 가정하는 자질을 가리키며, 주어진 어휘소 자체의 자질을 나타낸다. 선택 범주(select)는 주어진 어휘소가 선택하는 자질을 나타내며, 예를 들어 영어의 “the”와 같은 단어는 D라는 기본 범주와 =N이라는 선택 범주를 갖는 것으로 볼 수 있다. 선택 범주 =X를 갖는 핵(head)이 이에 해당하는 기본 범주 X를 갖는 성분을 선택할 때 외부 병합(external merge)이 일어나는데, 핵이 처음으로 선택한 성분은 보어(complement)로서 핵의 오른쪽에 결합하며, 핵이 하나의 성분을 이미 선택한 다음에 추가적인 논항과 결합하는 경우 이 추가 성분은 명시어(specifier)로서 핵의 왼쪽에 결합한다. 외부 병합의 결과 핵이 갖는 선택 범주 자질(=X)과 선택된 성분이 갖는 기본 범주 자질(X)이 결합하여 두 자질이 사라지게 된다.

나머지 두 통사 범주는 case, wh 등 허가를 필요로 하는 자질을 가리킨다. 예를 들어 영어의 “who”와 같은 단어는 -wh라는 자질 요구 범주를 갖고, 이를 허가하는 보문소(complementizer)는 +wh라는 자질 부여 범주를 갖는다고 볼 수 있다. 하나의 통사 구조를 구성하고 있는 핵의 첫 번째 자질이 자질 부여 범주 +x이고 이 통사 구조 안에 상응되는 자질 요구 범주 -x이 있는 경우 내부 병합(internal merge), 혹은 전통적으로 이동(move)이라고 불려온 연산이 적용되어 자

질 요구 범주를 갖는 성분이 핵의 명시어 자리로 이동한다.¹⁾ 내부 병합의 결과 자질 부여 범주 +x와 자질 요구 범주 -x는 구조 내에서 사라진다.

지금까지 소개한 바를 종합하여 예를 들자면, 영어의 정관사 “the”에 해당하는 어휘소는 다음과 같이 나타낼 수 있다.

(3) the :: =N -case D

여기서 음성/의미 자질은 단순히 해당 단어의 정서법 표기를 이용하며, 이 단어의 통사적 자질의 경우 앞서 기술한 바와 같이 D라는 기본 범주와 =N이라는 선택 범주를 갖고, 거기에 더해 격 점점을 받아야하므로 -case(혹은 -k)라는 자질 요구 범주를 갖는다. 통사 자질을 상세하는 순서는 통사 구조의 도출 과정에서 나타나는 순서를 따르도록 하며, 어휘소의 더 많은 구체적인 예는 다음 절에서 제시하도록 한다.

덧붙여, 문법 상세에는 다음과 같은 부가(adjoin) 연산도 포함시키는 것이 가능하다. YP라는 수식어가 XP에 부가될 때, 부가되는 방향에 따라 << 또는 >> 기호를 써서 표시하도록 한다. YP가 XP의 오른쪽에 부가되는 경우 X<<Y, YP가 XP의 왼쪽에 부가되는 경우 Y>>X 와 같이 표기하며, 기호의 끝이 핵을 가리키는 것으로 직관적으로 이해할 수 있다. 예를 들어 영어에서 형용사구가 명사구의 왼쪽에 부가되어 명사구를 수식하는 구조를 나타내는 문법은 다음과 같이 기술할 수 있다.

(4) A >> N

이렇게 정의한 MG 문법은 전통적으로 생성통사이론에서 논의해온 병합(merge), 부가(adjoin), 이동(move) 등의 연산을 모두 구현할 수 있다. MG로 정의할 수 있는 문법은 촘스키 위계(Chomsky Hierarchy; Chomsky 1963)로 불리는 형식 언어 위계 상에서 제1유형(문맥의존문법 context-sensitive grammar)과 제2유형(문맥자유문법 context-free grammar)의 사이에 위치한다(Michaelis 1998). 이러한 유형의 문법은 약문맥의존문법(mildly context-sensitive grammar)으로 불리며, 자연언어를 나타내기 위해 가장 적합한 유형의 형식 문법으로 알려져 있다(Joshi 1985).

1) Stabler (2011)는 모든 이동을 기본적으로 구 이동(phrasal movement)으로 가정하고 있다. 핵 이동(head movement)을 포함하는 문법은 Stabler (1997)에서 제시한 바와 같이 선택 범주의 하위 분류를 통해 가능하게 만들 수 있으나, 본고에서는 고려하지 않기로 한다.

3. 기본 구문 분석

이 절에서는 한국어의 통사 현상을 최소주의 전산문법으로 구현하기 위한 기본적인 방법을 소개하고자 한다. 여기서 다루는 각 통사구조에 대한 이론적인 분석은 여러 가지가 있을 수 있으나, 본고에서는 관계절의 전산적 구현을 제시하기 위한 최소한의 예시를 보이는 것을 목표로 한다. 논문을 읽는 연구자들은 이를 참조하여 다양한 이론적인 분석을 구현해볼 수 있을 것이다.

3.1. 논항

한국어의 명사구를 제대로 분석하기 위해서는 조사를 어떻게 다룰 것인가가 중요한 쟁점이 된다. 한국어의 격조사를 최소주의 전산문법으로 구현하기 위한 시도있는 논의는 Lee (2005)에서 찾아볼 수 있다. Lee (2005)에서는 격조사 중첩에 이르기까지 폭넓은 현상을 다루기 위하여 Stabler가 원래 제안한 문법을 수정한 전산문법을 제안하였다.

본고에서는 Stabler의 원래 문법 체계를 벗어나지 않는 한도 내에서 기본적인 수준의 한국어 명사구 분석을 구현하는 예시를 보이려고 한다. 이 분석에서는 격조사가 독립적인 어휘 요소로서 격조사구(CaseP 혹은 KP)의 핵을 나타내는 것으로 해석한다(Ahn 1988, Hoshi 2004, Kayne 1994, Kayne 2003, Neeleman & Weerman 1999). 한국어는 핵 후행 언어(head-final language)로 분석하는 경우가 많으나, 앞절에서 설명한 바와 같이 최소주의 전산문법은 명시어-핵-보어 순서를 기본적으로 가정하므로 한국어의 핵 후행성(head-finality)은 Kayne (1994)의 논의에서처럼 이동을 통해 얻는 것으로 구현할 수 있다. 또한 이 분석에서는 한정사구(DP)의 존재를 가정한다. 많은 경우 한국어 명사구에서 한정사의 존재는 가시적으로 드러나지 않으나, 이는 후술하게 될 운용소 및 관계절 분석을 위해 필요하다. 한정사구가 격조사구의 명시어 자리로 이동한다고 가정함으로써 명사 뒤에 격조사가 나타나는 어순을 설명할 수 있으며(Kayne 2003), 이는 확대투사원리(EPP) 자질을 통해 이루어지는 것으로 설명한다(Hoshi 2004).

이와 같은 통사적 분석을 최소주의 전산문법으로 표현하자면 먼저 명사, 한정사, 격조사는 각각 N, D, K라는 기본 범주를 갖는 것으로 볼 수 있다. 명사는 이러한 기본 범주만 가지며, 한정사와 격조사는 각각 =N과 =D라는 선택 범주를

갖는다. 한정사와 격조사는 또한 각각 자질 부여 범주 +epp와 자질 요구 범주 -epp를 갖는다. 마지막으로 격조사는 격 일치를 위한 자질 요구 범주를 추가로 갖는다. 한국어의 경우 주격과 목적격 조사가 분리되어 있으므로 이에 따른 자질 요구 범주를 각각 -nom과 -acc으로 나타낼 수 있다.

관련 있는 어휘소들의 예시는 다음과 같다. 음성적으로 실현되지 않는 어휘소는 엡실론(ϵ) 기호를 사용하여 음성 자질이 없음을 명시적으로 나타내기로 하며, 실제 프로그래밍에서는 음성 자질 상세를 비워둠으로 구현할 수 있다.

(5) 명사 어휘소 예시

민아 :: N

학생 :: N

책 :: N

(6) 한정사

ϵ :: =N D -epp

(7) 격조사

이 :: =D +epp K -nom

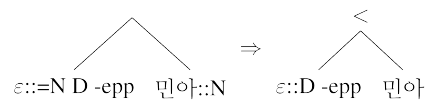
가 :: =D +epp K -nom

을 :: =D +epp K -acc

를 :: =D +epp K -acc

구체적인 도출 과정의 예시로 “민아가”와 같은 주격 명사구가 도출되는 과정은 다음과 같다. 먼저 비가시적 한정사가 명사를 보어로 취하는 외부 병합을 통해 전체적으로 D -epp 라는 자질을 갖는 성분이 도출된다.

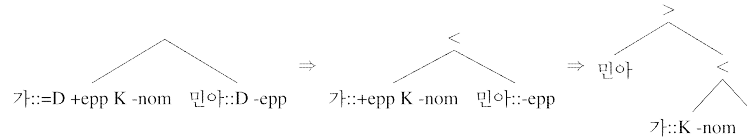
(8) 한정사구 도출



그 다음 격조사가 한정사구를 보어로 취하는 외부 병합이 일어나고, epp 자질이 유도하는 내부 병합을 통해 한정사구가 격조사의 명시어 위치로 이동한다. 이 결과 K -nom이라는 자질을 갖는 성분이 도출된다. 마찬가지로 “책을”과 같은 대격 명사구는 최종적으로 K -acc 자질을 갖는 것으로 도출된다. 아래의 도

식에서 한정사구의 내부 구조는 지면상 생략하였으며, 앞으로 도출 과정에 영향을 미치지 않는 내부 구조는 생략하기로 한다.

(9) 격조사구 도출



한편, 한국어에서는 명사구가 비가시적으로 나타나는 경우가 많으므로 음성적으로 실현되지 않는 *pro*에 해당하는 어휘소를 문법 상세에 포함시킬 필요가 있다. 이러한 *pro*는 EPP를 요구하지 않는 비가시적 격조사와 결합하는 것으로 볼 수 있다.

(10) *pro*

$\varepsilon :: D$

(11) 비가시적 격조사

$\varepsilon :: =D K -nom$

$\varepsilon :: =D K -acc$

3.2. 술어

한국어의 동사 및 형용사 등 용언의 내부 구조를 기술하는 데 있어서는 크게 두 가지 분석이 가능하다. 예를 들어 “먹었다”와 같은 동사의 내부 구조를 “먹” “었” “다”가 각각 VP, TP, CP의 핵에 해당하는 것으로 분석할 수도 있고 (Koopman 2005), 혹은 “먹었다”라는 동사 전체가 VP의 핵이며 비가시적인 T와 C를 갖는 것으로 분석할 수도 있다(Sells 1995). 전자는 통사적 분석, 후자는 어휘적 분석으로 명명할 수 있으며, 두 분석 모두 최소주의 전산문법을 통한 전산적인 구현이 가능하나 여기서는 설명의 편의를 위해 어휘적 분석의 예를 들어 구현 방법을 제시하도록 한다.

동사의 기본 범주는 V로 표시할 수 있고, 논항의 수에 따라 추가적인 선택 범주에 그에 따르는 자질 부여 범주를 갖게 된다. 자동사의 경우 선택 범주가 없고, 타동사의 경우 목적어를 취하고 여기에 격을 부여할 수 있으므로 =K라는

선택 범주와 그에 따르는 자질 부여 범주 +acc를 갖는다.

(12) 동사 어휘소 예시

잔다 :: V

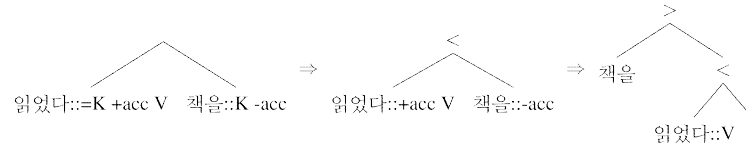
웃었다 :: V

먹는다 :: =K +acc V

읽었다 :: =K +acc V

예를 들어 “책을 읽었다”라는 동사구의 도출은 다음과 같다. 먼저 타동사 “읽었다”의 선택 범주 =K 와 목적격 명사구 “책을”의 기본 범주 K가 결합하면서 외부 병합이 이루어지며, “읽었다”의 자질 부여 범주 +acc가 자질 요구 범주 -acc를 갖고 있는 “책을”을 이동시킨다. 도출 결과 최종적으로 동사 자체의 자질과는 무관하게 모든 동사구는 V라는 자질만을 갖게 된다.

(13) 동사구(VP) 도출

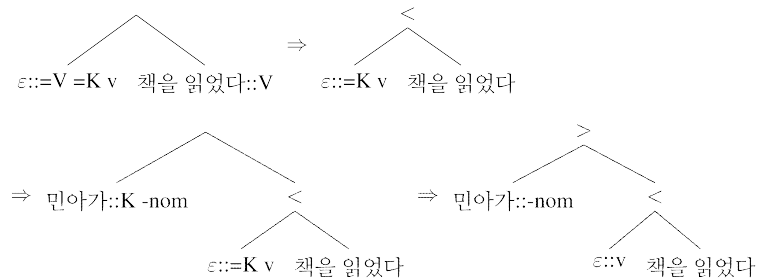


이렇게 형성된 동사구 VP는 경동사(little v)의 보어로 결합하면서 외부 병합이 이루어지며, 이는 다시 주어에 해당하는 명사구를 명시어로 하는 외부 병합을 통해 논항 구조를 완성한다.

(14) little v

ε :: =V =K v

(15) 동사구(vp) 도출



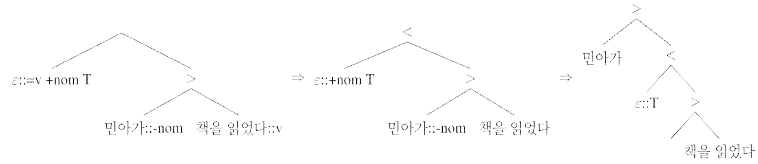
어휘적 분석에 의하면 완결된 문장을 만들기 위해서는 이 구조에서 더 나아가 비가시적인 시제와 보문소가 결합하게 된다. 시제는 T라는 기본 범주를 갖고 v를 선택하며, 주격을 부여하는 역할을 하므로 일반적으로 다음과 같은 어휘소로 정의할 수 있다.

(16) 시제소

$\varepsilon::=v +nom\ T$

시제와 동사구 vp가 만나면 먼저 v 자질에 의한 외부 병합이 이루어지며, vp 내의 주어는 격 점검을 받고 이동한다.

(17) 시제구(TP) 도출

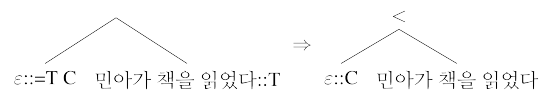


보문소는 C라는 기본 범주를 갖고 T를 선택한다. 단순 평서문을 나타내는 보문소를 나타내는 어휘소는 다음과 같으며, 이러한 보문소와 시제구의 병합 결과 최종적으로 완결된 문장을 얻는다.

(18) 보문소

$\varepsilon::=T\ C$

(19) 보문소구(CP) 도출



4. 장거리 의존 관계 분석: 한국어의 관계절

이 절에서는 장거리 의존 관계 분석과 이를 최소주의 전산문법으로 구현하는 방법을 관계절을 중심으로 설명하고자 한다. 여기서 소개하는 두 가지 분석, 즉 부가 분석과 인상 분석은 관계절 내부로부터 어떤 종류의 이동을 가정하느냐에

따라 분류한 것이며, 이동을 아예 가정하지 않는 분석도 가능하지만(채희락 2012 참조) 통사적 이동을 전산적으로 구현하는 문법을 소개하는 것이 목표인 본고에서는 다루지 않기로 한다.

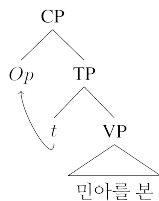
본고에서 “관계절”로 분석하고자 하는 대상은 전형적인 관계절, 즉 관형절 내에 공백이 존재하고 이 공백이 관형절이 수식하는 명사와 상응하는 경우로 한정한다. 이는 이동 분석이 가장 널리 받아들여지는 구문이기 때문이며, “도둑이 도망 가는 것을 잡았다”와 같은 이른바 내핵 관계절(head-internal relative clause)이나 “바람이 부는 소리를 들었다” 등 명시적인 공백이 없는 관계절과 같이 이동을 가정하지 않는 통사적 분석이 보다 널리 받아들여지고 있는 종류의 구문은 이 절에서 제시하는 문법을 적용시킬 수 있는 대상에서 제외한다.

4.1. 부가 분석

관계절에 대한 부가 분석(Adjunction analysis)은 관계대명사가 의문사와 유사하게 wh 자질을 갖고 Spec-CP로 이동한다고 가정한다(Chomsky 1977, Yang 1990, J.-I. Han 1992, J.-H. Han & Kim 2004). 한국어의 경우 명시적인 관계대명사가 존재하지 않으므로, 관계대명사에 해당하는 보이지 않는 운용소(operator)가 비가시적인 이동을 하는 것으로 가정할 수 있다.

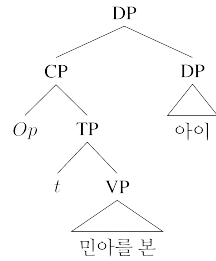
관계절을 포함하는 “민아를 본 아이”라는 성분을 예로 들어보자. 주어가 비어 있는 관계절 “민아를 본”은 보이지 않는 관계대명사 운용소 Op가 논항 자리에서 wh 자질 점검을 받기 위해 Spec-CP로 이동하여 이루어진다.

(20) 부가 분석 1단계: 운용소 이동



이 관계절이 핵 명사의 왼쪽에 부가되면 “민아를 본 아이”라는 성분이 완성된다. 이 분석에서는 관계절 내의 문장 요소와 핵 명사 사이의 의미적 연관 관계(즉, 민아를 본 것이 아이라는 의미)가 비통사적으로 도출된다. 핵 명사가 관계절 내부의 성분과 통사적으로 아무런 연관이 없고 다만 관계절이 핵 명사에 부가될 뿐이므로 “부가 분석”이라는 이름을 붙일 수 있다.

(21) 부가 분석 2단계: 관계절 부가



이와 같은 분석은 최소주의 전산문법으로 다음과 같이 구현해 볼 수 있다. 먼저 *wh* 자질을 갖는 비가시적 관계대명사 운용소(*Op*)는 다음과 같은 어휘소에 해당하는 것으로 볼 수 있다.

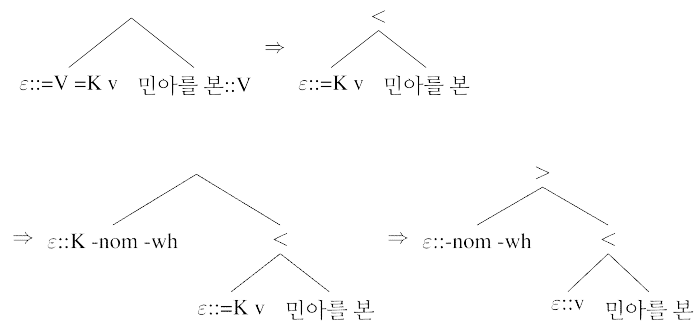
(22) 관계대명사 운용소

$\varepsilon :: D -wh$

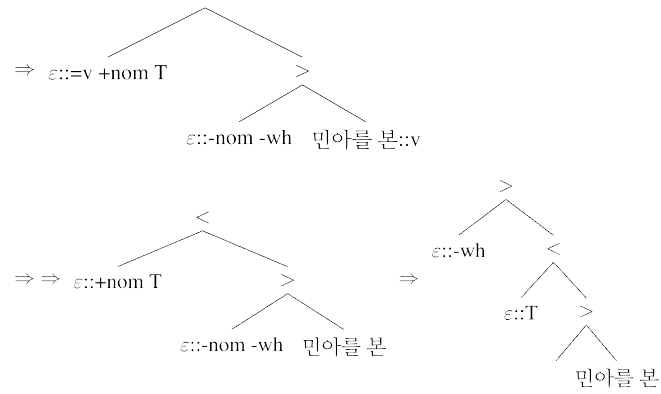
이러한 *wh* 운용소는 *pro*와 마찬가지로 (11)과 같은 비가시적 격조사와 결합한다고 분석할 수 있다. 예를 들어 주격 관계대명사 운용소는 음성 자질을 갖지 않으나 *K -nom -wh*라는 통사 자질을 갖는 것으로 도출된다.

논항 구조를 완성하는 동사구가 도출되는 과정과 시제구가 도출되는 과정은 *wh* 자질과 무관하게 이루어지므로 앞서의 절에서 소개한 바와 동일한 도출 과정을 거친다. 예를 들어 주격 관계대명사 운용소를 갖는 동사구와 시제구가 도출되는 과정은 다음과 같다.

(23) 동사구(vp) 도출



(24) 시제구 도출

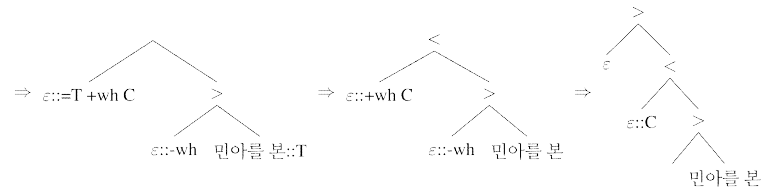


관계절의 보문소는 +wh 자질 부여 범주를 가지는 것으로 볼 수 있으며, 따라서 -wh 자질 요구 범주를 갖는 관계대명사 운용소를 이동시킨다.

(25) wh 보문소

$\epsilon::=T +wh C$

(26) 관계절 도출

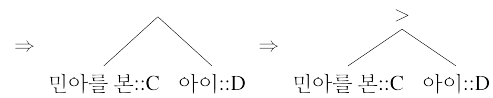


이렇게 도출한 관계절은 다음과 같이 핵 명사의 왼쪽에 부가될 수 있다.

(27) 관계절 부가 문법

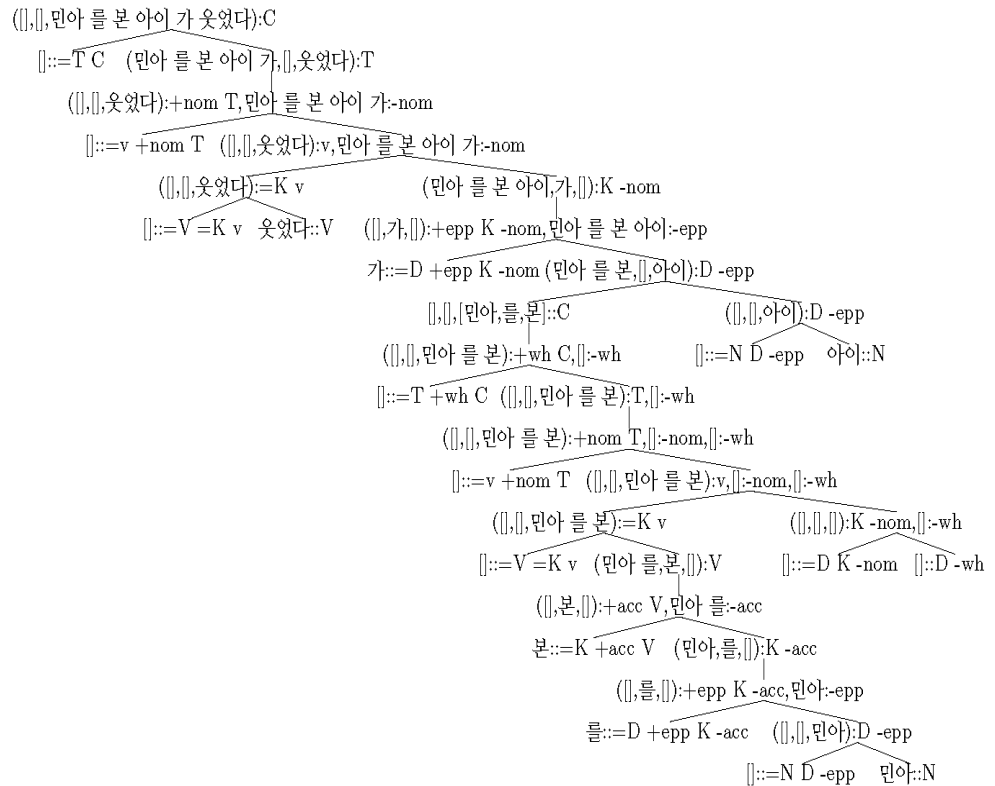
$C >> D$

(28) 관계절 부가



지금까지의 논의를 종합하여 “민아를 본 아이가 웃었다”와 같은 완결된 문장의 전체 도출 과정을 다음과 같이 하나의 수형도로 나타낼 수 있다. 도출 과정을 나타내는 수형도(derivation tree) 안에서 이분지(binary branch) 선은 외부 병합, 단분지(unary branch) 선은 내부 병합을 나타낸다.²⁾ 병합의 결과로 얻은 통사 구조는 괄호 안에 3개의 항목, 즉 명시어, 핵어, 보어 순서로 각각의 음성 자질을 선형적으로 나열되어있으며, 음성 자질이 비어있는 경우 각괄호 []로 표시한다. 통사 자질은 어휘소의 경우 이중 쌍점(::) 뒤에, 도출된 성분의 경우 쌍점(:) 뒤에 나열되어있다.

(29) 관계절 부가 분석을 적용한 전체 문장 도출

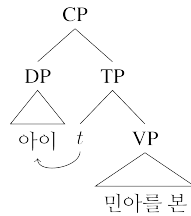


2) 이 수형도에서는 선형적인 위치를 나타내는 정보는 빠져있으며, 이분지 구조 내에서 어느 쪽 성분이 핵인지를 >나 <로 나타내는 대신 항상 왼쪽 성분이 핵인 것으로 이해할 수 있다.

4.2. 인상 분석

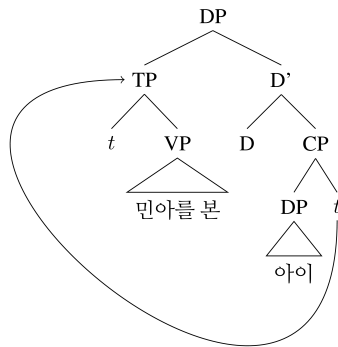
관계절에 대한 인상 분석(Promotion analysis)은 핵 명사가 관계절 내부에서 생성된 후 이동한다고 가정한다(Kayne 1994, Murasugi 2000, Hoshi 2004, Frellesvig & Whitman 2011). 따라서 부가 분석에서처럼 wh 운용소를 가정하는 대신 인상 분석에서는 핵 명사구 자체가 wh 자질을 갖는 것으로 분석한다. 예를 들어 “민아를 본 아이”와 같은 관계절에서 핵 명사 “아이”는 관계절 내에서 생성된 뒤 wh 자질 점점을 받기 위해 Spec-CP로 이동한다.

(30) 인상 분석 1단계: 핵 명사구 이동



이렇게 생성된 관계절은 비가시적인 한정사의 보어로 결합하며, 관계절 내의 시제구가 이 한정사의 명시어 자리로 이동하여 관계절이 핵 명사보다 앞에 나오는 어순이 완성된다(Kayne 1994:94).

(31) 인상 분석 2단계: 시제구 이동



이와 같은 분석은 최소주의 전산문법으로 다음과 같이 구현해 볼 수 있다. 먼

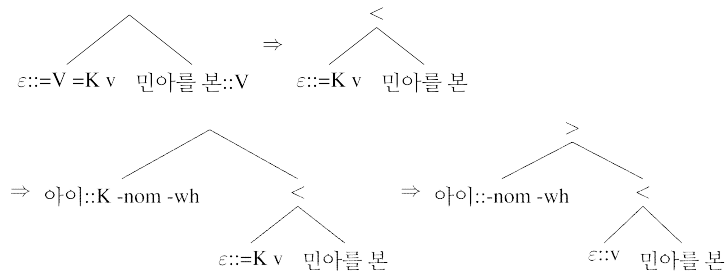
저 wh 자질을 갖는 비가시적 한정사를 가정하고, 핵 명사가 이와 결합하여 한정사구를 이루는 것으로 분석한다.

(32) wh 한정사

$\varepsilon::=N\ D\ -wh$

이는 다시 비가시적 격조사와 결합하여 격조사구를 이루며, 이렇게 wh 자질을 갖는 KP는 앞 절에서 부가 분석을 설명했던 때와 마찬가지로 동사구(vp)를 완성하는 과정에서는 여타 명사구가 쓰인 경우와 동일한 도출 과정을 거친다. 시제구의 도출 과정 역시 EPP 자질을 갖는 시제소와 결합한다는 것 이외에는 일반적인 경우와 동일하다. 예를 들어 wh 자질을 갖는 “아이”라는 주격 명사구는 K -nom -wh라는 통사 자질을 가지며, 이를 포함하는 동사구와 시제구의 도출 과정의 예시는 다음과 같다.

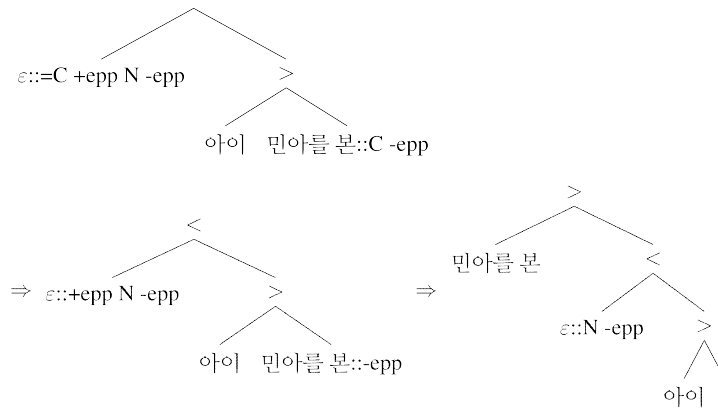
(33) 동사구(vp) 도출



(34) 시제소

$\varepsilon::=v\ +NOM\ T\ -epp$

(38) 관계절 도출 2단계



지금까지의 논의를 종합하여 “민아를 본 아이가 웃었다”와 같은 완결된 문장의 전체 도출 과정을 다음과 같이 하나의 수형도로 나타낼 수 있다. 이와 같은 도출 수형도(derivation tree)의 구성 요소는 앞서 관계절의 인상 분석을 적용한 예시에서 설명한 바와 같으며, 인상 분석의 경우 부가 분석에 비해 더 많은 이동을 가정한다는 것을 가시적으로 드러낸다.

(39) 관계절 인상 분석을 적용한 전체 문장 도출

$$\begin{array}{c}
\langle [], [], \text{민아를 본 아이가 웃었다} \rangle : C \\
\begin{array}{c}
\text{[::=T} C \text{ (민아를 본 아이가, [], [], 웃었다):T} \\
\langle [], [], 웃었다 \rangle : +nom T, \text{민아를 본 아이가:nom} \\
\text{[::=v} +nom T \text{ ([], [], 웃었다):v, 민아를 본 아이가:nom} \\
\langle [], [], 웃었다 \rangle : K v \quad (\text{민아를 본 아이가, [], [], }): K -nom \\
\text{[::=V} = K v \text{ 웃었다::V} \quad (\text{ [], [], }): +epp K -nom, \text{민아를 본 아이: -epp} \\
\text{[::=D} +epp K -nom \text{ (민아를 본, [], 아이):D -epp} \\
\text{[::=C} +epp D -epp \text{ (민아를 본, [], 아이):D -epp} \\
\text{[::=T} +wh C \text{ ([], [], 민아를 본):T -epp, 아이: -wh} \\
\langle [], [], 민아를 본 \rangle : +nom T -epp, [::nom, 아이: -wh} \\
\text{[::=v} +nom T -epp \text{ ([], [], 민아를 본):v, [::nom, 아이: -wh} \\
\langle [], [], 민아를 본 \rangle : K v \quad (\text{ [], [], }): K -nom, \text{아이: -wh} \\
\text{[::=V} = K v \text{ (민아를 본, [], }): V \quad \text{[::=D} K -nom \text{ ([], [], 아이):D -wh} \\
\langle [], \text{본, []} \rangle : +acc V, \text{민아를: -acc} \quad \text{[::=N} D -wh \text{ 아이: N} \\
\text{본::=K} +acc V \text{ (민아를, []):K -acc} \\
\langle [], \text{를, []} \rangle : +epp K -acc, \text{민아: -epp} \\
\text{를::=D} +epp K -acc \text{ ([], [], 민아):D -epp} \\
\text{[::=N} D -epp \text{ 민아: N}
\end{array}
\end{array}$$

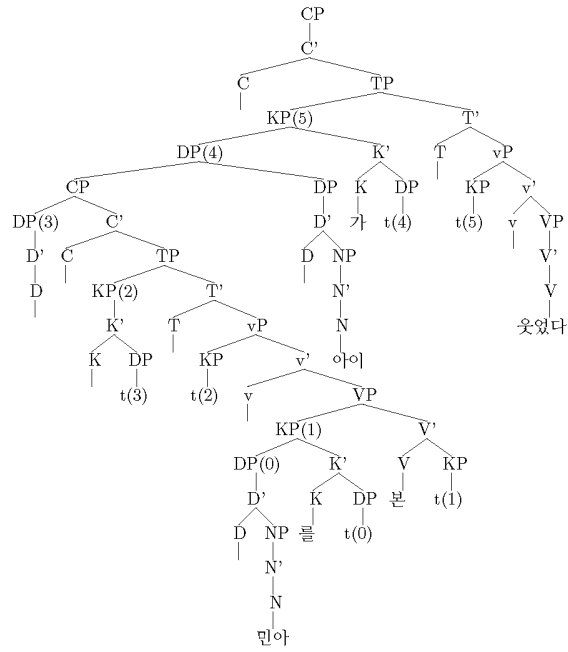
5. 활용 방안

지금까지 설명한 최소주의 전산문법을 기반으로 한 통사 구조의 도출은 본문에서는 설명을 위해 단계별로 제시했으나 실제로는 구문분석기(parser)를 이용하여 전 단계를 자동으로 생성할 수 있다. 최소주의 전산문법의 경우 이미 다양한 종류의 구문분석기가 개발되어 있으며 인터넷 상에서 쉽게 구할 수 있다.³⁾ 본문에서 전산 문법의 실제 구현과 문장 단위의 도출 수행도를 얻는 데에는 프롤로그(Prolog) 프로그래밍 언어를 이용한 구문분석기(mgcky-swipl)을 사용하였다.

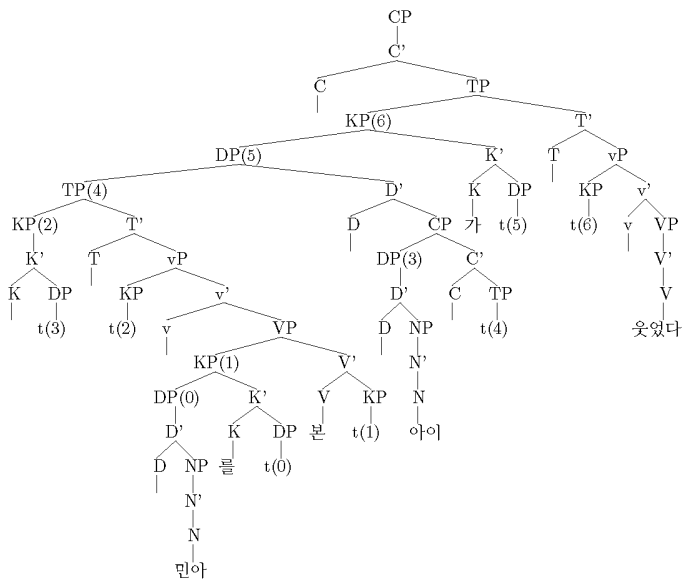
이렇게 구현한 전산문법의 활용 방안으로는 어떤 것들이 있을까? 먼저 전산 문법 개발의 일차적인 목표는 자동 구문 분석에 활용하는 것으로, 최소주의 통사 이론을 배우는 학생들과 이를 수정하고 발전시키려는 연구자들에게 수작업으로 생성한 통사 구조를 자동으로 검증하는 도구를 제공함으로써 도움이 될 수 있다. 본문에서 이용한 구문분석기의 경우 최소주의 모형에 따른 도출 수행도(derivation tree)뿐만 아니라 그에 상응하는 전통적인 X-bar 구조를 가진 수행도 또한 다음의 예시와 같이 자동으로 생성할 수 있으므로 분석 및 설명에 더욱 유용하다.

3) <https://linguistics.ucla.edu/people/stabler/coding.html> 참조.

(40) 부가 분석 (29)의 X-bar 수형도



(41) 인상 분석 (39)의 X-bar 수형도



더 나아가 같은 통사적 현상에 대해 서로 상충하는 이론들이 있을 때, 이들을 전산적으로 구현하여 구문 분석을 해보고 어느 이론의 구현 결과가 실제 경험적인 관찰 결과에 더 잘 부합하는지를 살펴봄으로서 각 이론의 경험적 적합성을 가늠해보는 용도로 활용할 수 있다. 예를 들어 Hale (2006)은 영어 관계절을 최소주의 전산문법으로 구현했을 때 인상 분석을 적용한 경우에는 실험언어학 연구 결과와 일치하는 결과를 얻었으나, 부가 분석을 적용한 경우에는 그렇지 못했다. 이는 영어 관계절에 대한 인상 분석의 경험적 적합성을 뒷받침하는 근거로 들 수 있다.

또 다른 활용 방안으로는 실험언어학 연구를 통해 얻은 관찰 결과의 원인을 설명하는 데 도움이 될 수도 있다. 예를 들어 Yun et al. (2015)은 한국어 및 기타 동아시아 언어들을 대상으로 최소주의 전산문법을 활용하여 구문 분석을 했을 때 예측되는 통사 구조들로 인한 전산적 정보량의 불확실성(information entropy)이 변화하는 패턴과, 이들 언어를 대상으로 한 기존 실험언어학 연구에서 밝혀진 문장 처리에 소요되는 인지적 요구량(cognitive load)이 변화하는 패턴이 일치한다는 것을 보여주었다. 이는 문장을 처리하는 인지 과정을 설명하는 데 최소주의 통사론이 기여할 수 있음을 시사하고 있다.

6. 결론

지금까지 최소주의 전산문법(Minimalist Grammar)을 이용하여 한국어의 통사 구조를 전산적으로 구현하는 방법을 살펴보았다. 최소주의 통사이론은 전산언어학적 응용성이 떨어지는 것으로 여겨지는 경우가 많으나, 이를 기반으로 한 최소주의 전산문법은 본문에서 소개한 바와 같이 통사 이론을 체계적이면서도 비교적 손쉽게 구현이 가능하다는 것이 큰 장점이며 구현 이후에는 이론언어학 교수법에 활용하거나, 이론언어학 연구 결과의 경험적 적합성을 가늠하거나, 실험언어학 연구 결과의 원인을 설명하는 등 많은 활용 방안이 있다.

종합하여 보면 최소주의 전산문법은 이론언어학, 실험언어학, 전산언어학이 상호적으로 발전하는 데 기여하는 역할을 할 수 있다고 기대된다. 본고에서는 예시를 위해 한정된 몇몇 구문만을 다루었으나 보다 다양한 통사 구조를 분석할 수 있는 문법 개발을 추후의 연구 과제로 남긴다.

참고문헌

- 고희정(2018), “생성통사이론의 변천과 발전 전망”, 『한국어학』 81, 1-33.
- 김종복 · 양재형(2003), “제약기반이론에서의 한국어구구조문법과 LKB(Linguistic Knowledge Building) 시스템을 이용한 구문분석기 구축”, 『한국어학』 20, 1-40.
- 채희락 (2012), “한국어에 과연 관계절이 존재하는가”, 『언어』 37(4), 1043-1065.
- Ahn, Hee-Don (1988), Preliminary remarks on Korean NP, *Papers from Sixth International Conference on Korean Linguistics*, 1-15.
- Chomsky, Noam (1963), Formal properties of grammars, In R. D. Luce, R. R. Bush, & E. Galanter ed., *Handbook of Mathematical Psychology* 2, New York: Wiley, 323-418.
- Chomsky, Noam (1977), On wh-movement, In P. W. Culicover, T. Wasow, A. Akmajian ed., *Formal syntax*, New York: Academic, 71-132.
- Chomsky, Noam (1995), *The minimalist program*, Cambridge, MA: MIT Press.
- Frellesvig, Bjarke, & John Whitman (2011), Prenominal complementizers and the derivation of complex NPs in Japanese and Korean, *Japanese and Korean Linguistics* 18, 73-87.
- Hale, John (2006), Uncertainty about the rest of the sentence, *Cognitive Science* 30, 643-72.
- Han, Chung-hye & Jong-Bok Kim (2004), Are there “double relative clauses” in Korean?, *Linguistic Inquiry* 35, 315-37.
- Han, Jongim (1992), Syntactic movement analysis of Korean relativization, *Language Research* 28, 335-57.
- Hoshi, Koji (2004), Parametrization of the external D-system in relativization, *Language, culture and Communication* 33, 1-50.
- Joshi, Aravind Krishna (1985), Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions? In D. Dowty, L. Karttunen & A. Zwicky ed., *Natural Language Processing: Theoretical, Computational and Psychological Perspectives*, New York: Cambridge University Press, 206-50.
- Kayne, Richard (1994), *The antisymmetry of syntax*, Cambridge, MA: MIT press.
- Kayne, Richard (2003), Antisymmetry and Japanese, *English Linguistics* 20, 1-40.
- Kim, Jong-Bok, Jaehyung Yang, Sanghoun Song & Francis Bond, (2011), Deep processing of Korean and the development of the Korean Resource Grammar,

- Linguistic Research* 28, 635-72.
- Koopman, Hilda (2005), Korean (and Japanese) morphology from a syntactic perspective, *Linguistic Inquiry* 36(4), 601-33.
- Lee, Yong-Hun (2005), A Computational Minimalism Approach to Korean Case Markers, *Studies in Generative Grammar* 15, 489-510.
- Michaelis, Jens (1998), Derivational minimalism is mildly context - sensitive, In *Proceedings of International Conference on Logical Aspects of Computational Linguistics 1998*, New York:Springer, 179-198.
- Neeleman, Ad & Fred Weerman (1999), *Flexible syntax: A theory of case and arguments*, Springer Science & Business Media.
- Murasugi, Keiko (2000), Japanese complex noun phrases and the antisymmetry theory, *Step by step: essays on Minimalist syntax in honor of Howard Lasnik*, Cambridge, MA: MIT Press, 211-34.
- Sells, Peter (1995), Korean and Japanese morphology from a lexical perspective, *Linguistic Inquiry* 26(2), 277-325.
- Stabler, Edward (1997), Derivational minimalism, In C. Retoré. ed., *Logical aspects of computational linguistics*, Berlin: Springer, 68-95.
- Stabler, Edward (2011), Computational perspectives on minimalism, In C. Boeckx ed., *Oxford handbook of linguistic minimalism*, OUP Oxford, 617-643.
- Yang, Hyun-Kwon (1990), *Categories and barriers in Korean*, Doctoral dissertation, University of Texas at Austin.
- Yun, Jiwon, Zhong Chen, Tim Hunter, John Whitman, & John Hale, (2015), Uncertainty in processing relative clauses across East Asian languages, *Journal of East Asian Linguistics* 24, 1-36.

<Abstract>

Computational Modeling of Long-distance Dependency in Korean Using Minimalist Grammar

Jiwon Yun

This article presents a computational model of Korean syntax, focusing on the long-distance dependency introduced by relative clauses. The model utilizes a mildly context-sensitive grammar formalism called Minimalist Grammar (Stabler 1997, 2011), which directly formalizes syntactic theories based on Chomsky's Minimalist Program (Chomsky 1995). Two syntactic analyses of relative clauses are computationally implemented using the model, namely the adjunction analysis (Chomsky 1977) and the promotion analysis (Kayne 1994). Possible applications of this computational model and the direction of future research are discussed in the light of theoretical linguistics, experimental linguistics, and computational linguistics.

Key words: syntax, computational linguistics, computational grammar, parsing, relative clauses

논문 접수: 2020.07.18.

논문 수정: 2020.08.20.

게재 결정: 2020.08.21.