

제약기반이론에서의 한국어구구조문법과 LKB(Linguistic Knowledge Building) 시스템을 이용한 구문분석기 구축*

김종복 (Jong-Bok Kim)
Kyung Hee University
School of English
jongbok@khu.ac.kr

양재형 (Jaehyung Yang)
Kangnam University
School of Computer and Media Engineering
jhyang@kangnam.ac.kr

To appear in **한국어학(Korean Linguistics)** 20

Abstract

Though various morphological analysers for Korean have been developed, no serious attempts have been made to develop its syntactic parsers, partially because of its structural complexity. This paper presents a result of the on-going project to build up a computationally feasible Korean Phrase Structure Grammar (KPSG) and its implementations into the LKB (Linguistic Knowledge Building) system (Copestake 2002, Oopen et al. 2002). The LKB system (<http://lingo.stanford.edu/ftp>) is a grammar and lexicon development environment for use with constraint-based linguistic formalisms such as HPSG (Head-driven Phrase Structure Grammar). This on-going project first builds up a Korean Phrase Structure Grammar couched in the constraint-based framework of HPSG and then check its feasibility through the implementations in the LKB system. The grammar HPSG (Sag and Wasow 1999) is well suited to the task of multilingual development of broad coverage grammars. HPSG is a constraint-based, lexicalist approach to grammatical theory that seeks to model human languages as systems of constraints on typed feature structures. The grammar adopts the mechanism of *type hierarchy* in which every linguistic sign is typed with appropriate constraints and hierarchically organized. The characteristic of such typed feature structure formalisms facilitates the extension of grammar in a systematic and efficient way, resulting in a linguistically precise and theoretically motivated descriptions of Korean.

Key Words: 구문분석기, HPSG, 한국어, 제약기반 이론, 격, 어순, 뒤섞기, 복합술어, 관계절, 주제화

1 제약기반 이론의 이해

자연언어 기술에 있어서 기술적(descriptive), 설명적(explanatory) 타당성을 가진 문법을 구축하기 위한 전제조건 중의 하나는 적절한 제약성을 포함하고 있어야 한다는 것이다. 예를 들어 (1)과 같이 정확한 언어적 제약성이 없는 규칙으로는 올바른 문법을 기술 할 수 없다.

(1) 동사는 바로 앞에 있는 명사와 존칭 정보에서 서로 일치해야만 한다.

1980년대 이후 발전하기 시작한 제약문법(Constraint-Based Grammar: CBG)의 기본적인 가정은 제약들(constraints)의 집합으로 이루어져 있고 이들 제약간의 상호 작용이나 통합(unification)과 같은 매커니즘이 문법 기술에 중심적인 역할을 한다는 것이다. 오늘날에도 지속적으로 발전하고 있는 대표적인 제약문법으로는 어휘기능문법(Lexical Functional Grammar, LFG), 일반화구구조문법(Generalized Phrase Structure Grammar, GPSG), 핵어중심구구조문법(Head-driven

*본 논문은 2003년 한국어학회 전국 복합술대회에서 발표한 것을 수정, 보완한 것이다. 여러 가지 의견과 비판을 해주신 학회 참여자 여러분과 시정곤, 한정환, 최재용 선생님께서 감사드립니다. 본 연구를 수행하는 데 있어서 첫 번째 저자는 Brain Korea 21 Project 2003의 지원을 받았다.

Phrase Structure Grammar, 이하 HPSG), 구문문법(Construction Grammar) 등을 들 수 있다. 이러한 제약기반이론들이 갖추고 있는 공통된 목표와 기준을 요약하면 다음과 같다.

생성성(generativity): 문법이론은 유한수의 규칙으로 무한 수의 정문의 문장을 생성할 수 있어야 한다. 즉 문법이 만들어 내는 구조들이 (예를 들어 HPSG에서의 자질구조나 GB에서의 나무구조) 정형인지 비정형인지를 구분할 수 있어야 한다.

표현성(expressivity): 문법 기술을 위해 사용되는 메타언어(meta language)가 충분한 표현력을 가지고 있어야 한다.¹ 특정 문법이 사용하는 메타언어가 기술 대상 언어가 생성하는 모든 구조를 설명, 표현할 수 없다면 이는 문법으로서 적절치 않다는 것을 의미 한다.

경험적 타당성(empirical adequacy): 제약문법의 또 다른 중요한 기준은 경험적 자료들을 정확히 기술하여야 한다는 것이다. 모호하고 증명할 수 없는, 혹은 추상적이며 불확실한 원리나 공리(axiom)를 설정하는 문법이 아니라, 기술 대상 언어의 광범위한 자료들을 설명할 수 있는 명시적인 규칙(explicit rules)이나 원리를 가진 문법이어야 한다.

심리적 적절성(psycholinguistic responsibility): 언어의 이해나, 발화하는 과정에서 생성된 문장의 문법성을 판단 할 수 있는 내재적 언어 능력(linguistic competence)을 반영하여야 한다. 이를 위해 다음과 같은 관련 조건들도 만족시켜야한다.

- 비파괴성(nondestructiveness): 문법이 언어 구조를 변형하거나 파괴하는 Move- α 와 같은 이동 규칙을 설정하지 않는다. 문법 규칙이나 혹은 어휘정보들은 언어 구조를 동시에 만족할 수 있도록 상호 작용하는 제약들로만 존재한다.²
- 국부성(locality): 주어진 언어구조의 적형성을 판단할 때, 자신을 포함한 보다 큰 구조나 다른 구조와 비교해서 결정하거나 혹은 관련지어 결정되는 것이 아니라, 그 구조 자체 내에서 관련 제약들을 모두 준수하였는지의 여부만 점검한다.
- 평행성(parallelism): 언어 정보가 음운, 통사, 의미, 화용 등 다양한 정보 계층(level)을 포함하고 있다. 전통적 변형문법에서 설정하는 T-모형과 같이 이들 정보 계층 사이에 위계나 순서가 존재하는 것과는 달리, 제약기반이론에서는 이러한 관련 언어 정보가 동일한 계층 내에서 수평적, 평행적으로 존재한다.
- 비독립성(radical nonautonymy): 다양한 언어정보들이 서로 밀접한 상호 제약관계에 있다는 사실을 반영해야한다. 비록 언어 기술이나 이해에 있어서 경우에 따라 한가지 언어 정보만 필요할 수도 있지만, 많은 언어 현상은 여러 언어 정보 사이의 상호작용의 결과물이라는 사실을 반영해야 한다.³

2 HPSG의 이해

2.1 HPSG의 주요 특성

제약기반이론 중 대표적인 이론인 HPSG(Head-driven Phrase Structure)는 전산언어학적 응용을 목표로 개발된 문법이다. 이러한 이유로 HPSG는 수학적 명시성과 전산언어학적 응용성을 문법 기술의 중요한 척도로 삼고 있다. 즉, 추상성이나 문법적 조작보다는 기술적 정확성을 일차 목표로 하고 있다. 이를 위해 HPSG는 언어를 ‘유형화(typed)된 자질구조(typed feature structure)’에 관한 제약 체계로 설명하려고 시도한다.

2.2 자질구조(Feature Structures)

직관적으로 자질 구조란 (2)에서처럼 한 개체나 대상물(object)이 취할 수 있는 속성과 그 값을 짝으로 기술한 행렬(Attribute-Value Matrix: AVM) 이다.

¹홀어논리, 자질구조, 구표식(phrase marker), PROLOG, LISP 등을 메타언어라 할 수 있다.

²따라서 제약기반이론에서는 의문사이동(wh-movement), 접사이동(affix-hopping), 핵어이동(head-movement)과 같이 구조를 변경할 수 있는 규칙들이 존재하지 않는다. 그 주된 이유는 기존의 구조를 변경할 수 있는 언어 처리 모형이나 분석기를 구축하는 것이 어렵기 때문이다. 특히 우리가 언어를 처리하거나 이해하는 과정이 점진적이며(incrementally) 통합적(integrated)인 과정이라는 사실은 구조 변경규칙을 설정하는 것이 적절치 않다는 것을 말해준다. 1.2 참조

³예를 들어 어순현상에서의 통사, 음운적 정보의 상호작용, 결속이나 영역(scope) 현상에서의 통사, 의미적 정보의 상호 작용, 초점이나 주제 현상에서 음운적 화용적 정보들이 상호 작용 등을 들 수 있다. 최근 10년간의 문법 연구는 이러한 문법 정보간의 상호 작용에 중점을 두고 있다고 해도 과언이 아니다.

$$(2) \begin{bmatrix} \text{Attribute-1} & \text{Value-1} \\ \text{Attribute-2} & \text{Value-2} \\ \dots & \dots \end{bmatrix}$$

예를 들어 저자 중 한사람에 관한 정보를 최소한 다음과 같은 세 가지 속성(attribute)과 그 값으로 이루어진 AVM으로 나타낼 수 있을 것이다.

$$(3) \begin{bmatrix} \textit{person} \\ \text{NAME} & \text{Jong-Bok kim} \\ \text{GENDER} & \text{male} \\ \text{JOB} & \text{linguist} \end{bmatrix}$$

이러한 자질 구조의 중요한 특징은 유형화(typed) 되어 있다는 것이다. (3)과 같은 자질 구조는 *human*이나 *person*과 같은 개체에만 적용될 수 있지, 사물 *object* 나 속성 *property* 유형에 속하는 개체에 적용될 수는 없다. 이처럼 각 개체는 자신의 유형에 따라 어떠한 자질구조나 AVM이 자신에게 적합한지 결정 되어 있다. 예를 들자면 *country*라는 유형은 (4)에 나타난 것처럼 NAME이나 PRESIDENT 속성을 가질 수 있지만, *university* 유형은 MAYOR와 같은 속성을 가질 수는 없다.

$$(4) \quad \begin{array}{l} \text{a.} \begin{bmatrix} \textit{country} \\ \text{NAME} & \text{Korea} \\ \text{PRESIDENT} & \text{Mr. Roh} \end{bmatrix} \qquad \text{b.} \begin{bmatrix} \textit{university} \\ \text{NAME} & \text{Stanford University} \\ \text{MAYOR} & \text{MR. Lee} \end{bmatrix} \end{array}$$

언어 표현들도 모두 유형화 되어 있다. 곧 언어 표현의 각 유형들은 자신들이 취할 수 있는 자질구조가 이미 결정되어 있다는 것을 의미한다. 예를 들어 (5)에 나타난 것처럼 명사 noun 유형은 격 CASE 정보를 가질 수 있지만, 동사가 격 CASE 정보나 명사가 시제 TENSE 정보를 가질 수는 없다.

$$(5) \quad \begin{array}{l} \text{a.} \begin{bmatrix} \textit{noun} \\ \text{CASE} & \textit{nom} \end{bmatrix} \qquad \text{b.} \begin{bmatrix} \textit{verb} \\ \text{CASE} & \textit{dat} \end{bmatrix} \qquad \text{c.} \begin{bmatrix} \textit{noun} \\ \text{TENSE} & \textit{past} \end{bmatrix} \end{array}$$

이처럼 HPSG 이론의 목표는 어떠한 자질구조가 정형의 자질 구조인지를 정의하는 것이라 할 수 있다. 정형의 자질 구조는 문법이 정의하는 제약들을 준수해야한다. 즉 자질구조가 언어 현상의 모형화에 기본이 되는 문법은 다음 제약들을 정의한다고 할 수 있다.

- 언어 현상 기술에 필요한 유형(types)은 무엇인가?
- 주어진 유형의 하위유형(subtypes)과 상위유형(supertypes)들은 무엇인가?
- 각 유형들이 취할 수 있는 자질 구조는 무엇인가 (유형의 정형성 type well-formedness)?
- 또한 각 자질이 어떠한 값을 가질 수 있는가 (자질 값의 타당성 type resolved)?

자질 구조에서 중요한 연산 작용 중의 하나는 통합(unification)이다. 자질 통합이란 기본적으로 양립 가능한 자질들의 정보가 통합되어 하나의 자질 구조를 이루는 것이다. 예를 들어 (6)에서 시제 TENSE 가 과거 past인 자질구조와 서법 MOOD가 평서문 declarative인 정보를 가진 자질구조가 하나의 자질 구조로 통합될 수 있다.

$$(6) \quad \begin{bmatrix} \textit{verb} \\ \text{TENSE} & \textit{past} \end{bmatrix} \sqcup \begin{bmatrix} \textit{verb} \\ \text{MOOD} & \textit{decl} \end{bmatrix} \rightarrow \begin{bmatrix} \textit{verb} \\ \text{TENSE} & \textit{past} \\ \text{MOOD} & \textit{decl} \end{bmatrix}$$

그러나 만약 통합될 두 자질 구조가 상충된 자질 값으로 인해서 서로 양립될 수 없는 경우에는 (7)에서처럼 자질 통합이 이루어 질 수 없다.

$$(7) \quad \begin{bmatrix} \textit{verb} \\ \text{MOOD} & \textit{decl} \end{bmatrix} \sqcup \begin{bmatrix} \textit{verb} \\ \text{MOOD} & \textit{inter} \end{bmatrix} \rightarrow \begin{bmatrix} \textit{verb} \\ \text{MOOD} & \textit{decl} \\ \text{MOOD} & \textit{inter} \end{bmatrix}$$

(7)의 두 자질 구조는 서법 MOOD에 대한 각각 상충된 값을 가지고 있기 때문에 서로 통합될 수 없다.

자질 통합과 함께 자질의 포섭(subsumption) 관계도 중요한 역할을 한다. 포섭관계는 일반적인 정보를 가진 자질 구조가 보다 더 구체적인 정보를 가진 자질 구조와의 관계를 의미한다. 예를 들어 (8)a는 주격 *nom*인 모든 명사를 포함하지만, (8)b는 보다 구체적 정보인 주격 *nom*이면서 존칭 honorific 정보를 가진 명사를 요구하기 때문에 전자가 후자를 포섭한다.

$$(8) \quad a. \begin{bmatrix} \textit{noun} \\ \text{CASE } \textit{nom} \end{bmatrix} \supseteq b. \begin{bmatrix} \textit{noun} \\ \text{CASE } \textit{nom} \\ \text{HONORIFIC } + \end{bmatrix}$$

그러나 (9)a는 주격 *nom*의 명사에 관한 자질 구조이고, (9)b는 목적격 *acc*이며 비존칭 명사에 관한 자질 구조이기 때문에 두 관계는 포섭관계가 성립될 수 가 없다.

$$(9) \quad a. \begin{bmatrix} \textit{noun} \\ \text{CASE } \textit{nom} \end{bmatrix} \text{ NOT } \supseteq b. \begin{bmatrix} \textit{noun} \\ \text{CASE } \textit{acc} \\ \text{HONORIFIC } - \end{bmatrix}$$

이러한 통합과 포섭관계는 언어 표현들이 결합하거나 제약들을 준수하는지 여부를 판단하는 데 중요한 역할을 담당한다.

한편 자질구조들이 취할 수 있는 값은 원자(atomic), 목록(list), 집합(set), 그리고 자질구조 들이다.

$$(10) \quad \begin{bmatrix} \text{FEATURE1 } \textit{atomic} \\ \text{FEATURE2 } \langle \quad \rangle \\ \text{FEATURE3 } \{ \quad \} \\ \text{FEATURE4 } [F \dots] \end{bmatrix}$$

앞에서 살펴 본 것 같이 각 유형은 자신이 취할 수 있는 자질 FEATURE를 정하게 되고(well-formedness), 각 FEATURE는 자신이 취할 수 있는 값이 정해진다(type resolved). 예를 들어 타동사 *v-tran* 유형의 ‘잡다’의 정보를 자질구조로 나타내어 본다면 다음과 같다.

$$(11) \quad \begin{bmatrix} \textit{v-tran} \text{ ORTH } \langle \text{잡다} \rangle \\ \text{SYN(TAX)} \begin{bmatrix} \text{HEAD } \textit{verb} \\ \text{ARG-ST } \langle \text{NP}[\textit{nom}]_i \text{ NP}[\textit{acc}]_j \rangle \end{bmatrix} \\ \text{SEM(ANTICS)} \begin{bmatrix} \text{INDEX } s \\ \text{KEY} \begin{bmatrix} \text{PRED } \textit{catch} \\ \text{ARG0 } s \\ \text{ARG1 } i \\ \text{ARG2 } j \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

위의 자질 구조에서 처럼, 동사 ‘잡다’는 최소한 철자 ORTHOGRAPHY에 관한 정보, 통사부 SYNTAX에 관한 정보, 의미부 SEMANTICS에 관한 정보를 가지고 있다. 통사부 정보는 다시 품사 정보를 포함한 핵어 HEAD에 관한 정보와 논항 구조 ARG-ST (ARGUMENT STRUCTURE)에 관한 정보를 명시하고 있다. 의미부 정보는 ‘잡다’라는 상황 *s*에 두 개체가 참여하고 있다는 것을 포함하고 있다. HPSG 문법은 이처럼 유형에 맞는 자질구조를 체계적으로 정의하는 것이라 해도 과언이 아니다.

2.3 유형 위계(Type Hierarchy)와 다중상속(Multiple Inheritance)

앞에서 살펴본 ‘잡다’가 가지는 모든 어휘 정보를 각 어휘 요소마다 일일이 기술 할 수 없다. 어휘 정보를 표현하는 데 있어서 최소화해야할 것은 수직적 잉여성(vertical redundancy)과 수평적 잉여성(horizontal redundancy)이다. 수직적 잉여성이란 각 어휘 요소들이 다른 어휘 요소들과 공유하는 속성을 의미한다. 예를 들어, (12)에서는 논항의 수나 동사의 의미적 속성에 따라 용언들을 자동사, 타동사, 형용사 등으로 구분할 수 있다.

- (12) a. 자동사: 가다, 오다, 달리다, etc.
 b. 타동사: 먹다, 차다, 빌리다, etc.
 c. 형용사: 푸르다, 아름답다, 높다, etc.

이들 요소들 간의 공통점을 포착하여 어휘부를 조직하여야만이 수직적 잉여성을 최대한으로 줄일 수 있다.

반면 수평적 잉여성이란 일반적으로 어휘요소의 파생이나 굴절 관계에서 나타내는 잉여성을 의미한다.

- (13) 먹다: 먹다, 먹었다, 먹었구나, 먹어라, 먹자, 먹고, 먹는, 먹어야, 먹겠니, 먹음,

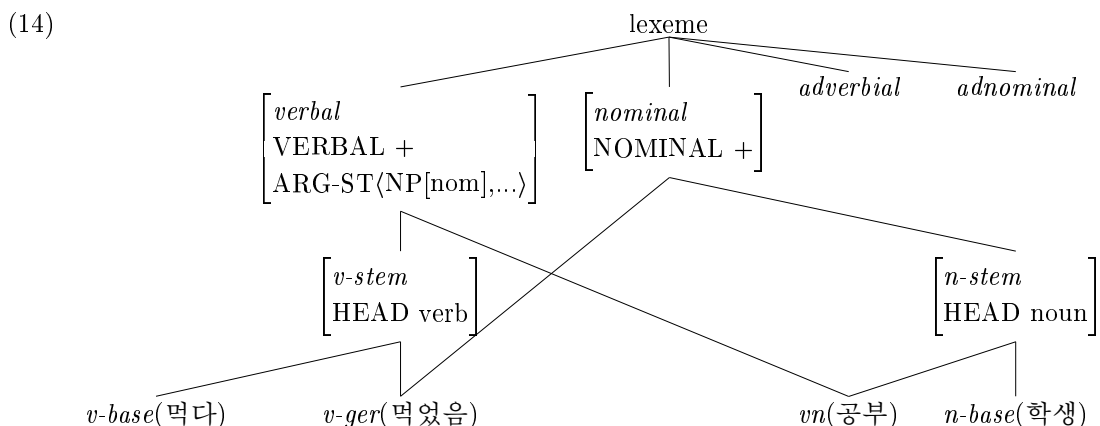
위의 어근 ‘먹다’는 파생과 굴절을 통해 다양한 형태로 실현된다. 효율적인 사전부의 구조는 이러한 잉여성을 제거해야만 한다.

컴퓨터 구현을 위한 문법 설계나 HPSG에서는 이러한 수직적 잉여성을 없애기 위해서 어휘들을 계층적 유형 위계(type hierarchy)로 분류하고, 수평적 잉여성을 위해서는 어휘규칙(lexical rules)들이 사용된다.

2.3.1 다중상속 유형 위계(Multiple Inheritance Hierarchical Lexicon)

어휘부 구성에서 발생하는 수직적 잉여성을 최소화하기 위한 유형 위계 구조는 어휘들을 기본적인 속성에 따라 유형화 하고 이를 다시 하위 혹은 상위 유형으로 구분한다. 이렇게 정의된 유형들은 자신 고유의 제약들을 가지고 있을 뿐만 아니라, 상위 유형이 가지고 있는 모든 제약들을 상속 받게 된다. 이 때 상속의 관계는 단선적인 상위 유형에만 국한되는 것이 아니라, 다양한 상위 유형에서 상속 받을 수 있다.

한국어에서 다중상속(multiple inheritance) 위계구조가 필요하다는 것을 쉽사리 찾아 볼 수 있다. 한국어의 경우 동명사나 서술 명사는 동사성과 명사성을 모두 가지는 것으로 인식되고 있는데, 이를 유형위계로 나타내면 다음과 같이 간략하게 나타낼 수 있다.



위 위계 구조는 전통적인 한국어 품사 분류 체계와 같이 품사들은 용언 verbal, 체언 nominal, 부사어 adverbial, 관형어 adnominal로 구분 한다. 이들 유형들은 각각의 제약들을 가지고 있다. 예를 들자면 verbal 유형은 [VERBAL +] 자질과 주격 명사구 NP[nom]를 논항으로 취하는 제약들을 가지고 있다. 여기서 각 유형들이 자질구조로 표현된 제약들을 가지게 된다는 것은 유형들이 취할 수 있는 자질 속성을 제약하여 준다는 것을 의미한다. 예를 들어 명사 noun은 시제 TENSE에 관한 정보를 가질 수 없다는 것을 정확하게 명시해 준다.

$$(15) \left[\begin{array}{l} * \text{HEAD} \left[\begin{array}{l} \textit{noun} \\ \text{TENSE } \textit{past} \\ \text{HONORIFIC } + \end{array} \right] \end{array} \right]$$

위 위계구조에서 최 하위 구조는 동사, 동명사, 서술명사, 명사 유형인 v-base, v-ger, vn, n-base 유형들로 구분되어 있고, 이들은 각각 자신 고유의 제약들을 가지고 있기도 하지만 자신들의 상위 유형들이 가지고 있는 제약들도 상속한다. 예를 들어 다음 어휘 정보 중에서 음영 처리된 부분들은 각 어휘 요소가 가지는 고유한 정보이지만, 나머지 정보는 상위 유형에서 상속된 정보들이다.

$$(16) \left[\begin{array}{l} \textit{v-trans} \\ \text{ORTH } \langle \text{먹었음} \rangle \\ \text{SYN} \left[\begin{array}{l} \text{HEAD} \left[\begin{array}{l} \textit{verb} \\ \text{VERBAL } + \\ \text{NOMINAL } + \end{array} \right] \\ \text{ARG-ST} \langle \text{NP}[\textit{nom}], \text{NP}[\textit{acc}] \rangle \end{array} \right] \\ \text{SEM } \textit{eat-relation} \end{array} \right]$$

$$(17) \left[\begin{array}{l} \textit{vn-tran} \\ \text{ORTH } \langle \text{공부} \rangle \\ \text{SYN} \left[\begin{array}{l} \text{HEAD} \left[\begin{array}{l} \textit{noun} \\ \text{VERBAL } + \\ \text{NOMINAL } + \end{array} \right] \\ \text{ARG-ST} \langle \text{NP}[\textit{nom}], \text{NP}[\textit{acc}] \rangle \end{array} \right] \\ \text{SEM } \textit{study-relation} \end{array} \right]$$

만약 각 어휘 요소들이 가지고 있는 모든 정보들을 일일이 모두 기술한다면 어휘부가 기술해야 하는 정보의 양도 엄청날뿐만 아니라, 어휘들의 요소가 가지는 공통된 속성을 포착할 수 없게 된다. 그러나 유형화된 다중상속 위계를 이용하면 어휘들이 가지는 수직적 잉여성을 최소화할 수 있으며, 필요한 정보만 어휘부에 기술해 주면 된다.

2.3.2 어휘 규칙(Lexical Rules)

이처럼 유형 위계 구조는 어휘부 정보의 수직적 잉여성을 제거해 주는 반면 어휘 규칙은 수평적 잉여성을 제거해 줄 수 있다. 예를 들어 과거시제 형성 어휘 규칙은 다음과 같이 나타낼 수 있을 것이다.

(18) 과거시제 형성 규칙:

$$\left[\begin{array}{l} \textit{verb-base} \\ \text{ORTH } \boxed{1} \\ \text{SYN } \boxed{3} \left[\text{HEAD } \textit{verb} \right] \\ \text{SEM} \left[\begin{array}{l} \text{INDEX } \textit{s} \\ \text{RELS } \langle \boxed{2} \rangle \end{array} \right] \end{array} \right] \Rightarrow \left[\begin{array}{l} \textit{past-verb} \\ \text{ORTH } \boxed{1} + \text{있} \\ \text{SYN } \boxed{3} \\ \text{SEM} \left[\begin{array}{l} \text{INDEX } \textit{s} \\ \text{RELS } \left\langle \boxed{2}, \left[\begin{array}{l} \text{PRED } \textit{temporally-precede} \\ \text{ARG1 } \textit{s} \\ \text{ARG2 } \textit{now} \end{array} \right] \right\rangle \end{array} \right] \end{array} \right]$$

위 어휘규칙처럼, 입력(input) 정보에 적합한 어휘 요소 X가 있다면 출력(output) 정보를 포함한 Y라는 어휘요소도 있다는 것을 의미한다. 즉 위의 경우 동사 어근에 상응하는 과거 시제 동사도 생성된다. 이 경우 과거 동사는 형태적으로 ‘있’을 첨가하며, 의미적으로는 자신이 지시하는 사건

이 현재(now) 보다 이전(temporally-precede)에 일어났다는 정보를 포함한다. 이와 같이 기본적인 어휘 항목을 어휘 규칙에 따라 체계적으로 확장시킴으로써, 어휘부에 등재되어야 하는 어휘요소의 수를 최대한 줄여 수평적 잉여성을 제거한다.

본 연구는 이러한 이론적 틀내에서 한국어 문법을 구축하고, 이를 LKB를 이용하여 한국어 구문분석기를 구축하는 데 목적으로 두고 있다.

3 LKB(Linguistic Knowledge Building) 시스템과 한국어구구조문법 (KPSG) 구성

언어지식구축 체계 LKB는 스탠포드대학의 CSLI 연구소에서 Ann Copestake와 (Copestake 2002) 여러 전산언어학자들이 HPSG와 같은 제약기반이론을 전산학적으로 구현할 수 있는 환경을 구축하기 위해서 개발되었다(<http://lingo.stanford.edu/ftp>). HPSG와같이 자질구조를 사용하는 문법들을 전산화 할 수 있는 소프트웨어로 개발된 LKB는 현재 영어, 스페인어, 독일어, 일본어, 그리스어, 노르웨이어의 문법 처리를 위해 활발하게 이용되고 있다. 특히 가장 활발하게 개발된 문법인 영어문법 English Resouce Grammar는 현재 전자메일 자동응답 상용화 프로그램을 개발하는 데에도 적용되었고, 그 응용 폭을 넓혀 가고 있다.

한국어구구조문법 개발을 위한 LKB의 기본적인 구조는 아래처럼 크게 문법을 유형화 언어(typed language)로 기술한 부분과 필요한 문법 구동을 위해 필요한 보조 파일들로 구성되어 있다.

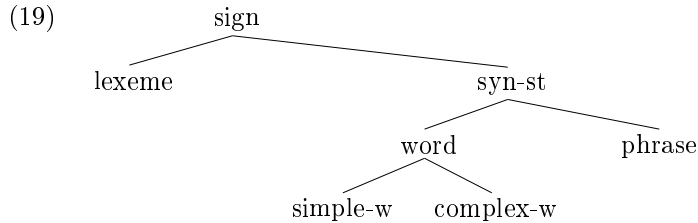
- 한국어구구조문법을 위한 유형 기술 언어 (Types Description Language (TDL) of KPSG)
 - types.tdl : 유형(type)에 관한 정의와 문법 전반에 걸친 위계 구조
 - sem-types.tdl: 의미 유형에 관한 정의와 위계 구조
 - lexicon.tdl : 어휘부 유형
 - infr.tdl : 파생 및 굴절을 포착하기 위한 어휘 규칙
 - lrules.tdl : 통사적 작용을 위한 어휘 규칙
 - grules.tdl : 문법 규칙과 구문들의 유형
- 보조 파일 (Grammar Configuraion for LKB):
 - script: 문법 로딩 파일
 - parse-nodes.tdl: 나무구조 출력을 위한 보조 유형
 - start.tdl: 문장 분석을 시작하기 위한 구조

이러한 LKB에 기반한 문법 시스템은 기존의 구문분석기와 몇 가지 중요한 차이점을 가지고 있다. 첫째는 일반적인 구문분석기와 달리 문법 구축시 **통사부와 의미부가 병행해서** 이루어진다. 유형 위계를 이용하여, 의미 합성의 부분도 통사적 요소들의 결합과 함께 처리한다. 이러한 통사부와 의미부의 상호 작용은 자질구조 내에서 통사, 의미, 화용 정보들이 순서에 상관 없이 같은 계층내에서 연결되어 있기 때문에 가능하다. 특히 LKB 및 HPSG 문법 내에서 개발된 Miminal Recursion Semantics은 양화사들의 영역 문제 등 난해한 **의미 합성**(compositionality) 부분까지 해결 해 주고 있다. 그리고 둘째는 이러한 의미부 정보의 구성은 의미부가 주어졌을 경우 문장 생성도 가능하게 하는 **생성기(Generator)**가 부착되어 있다는 것이다.⁴ 언어 사용은 기본적으로 이해와 생성이라는 점에서 궁극적인 언어처리를 위해서는 필요 불가결한 부분이라 할 수 있다. 현존하는 대부분의 구문분석기들이 문장의 구조를 올바르게 이해하는데 있어서조차 여러 가지 난점이 있다는 점을 고려할 때, 의미부 뿐만 아니라 문장 생성까지 목표한다는 점에서 LKB 시스템의 목표는 한 차원 높다고 할 수 있다.

⁴KPSG의미부와 생성부에 관해서는 지금 연구 단계 중이며, 본 연구에서는 지면 관계상 상세하게 다루지 않기로 한다.

4 제약기반이론에서의 한국어구구조문법과 구현

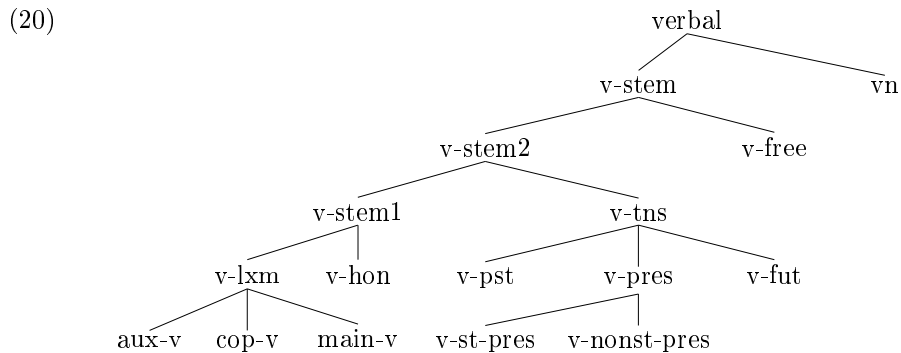
앞에서 언급한 바와 같이 HPSG에서는 어휘 표현뿐만 아니라 통사적 요소를 포함한 모든 언어 표현을 기호(sign)로 보며 이들 기호들은 다시 하위 유형으로 계층화 되어 있다. LKB내의 한국어 구구조문법(Korean Phrase Structure Grammar: KPSG) 역시 동일하게 모든 표현들이 유형화 계층내에 포함되어 있다. 즉 (19)에 간략하게 나타낸 것 처럼, 한국어 문법은 크게 어휘부 구조와 통사부 구조로 나누어져 있다.



어휘부 내의 적절한 어휘소(lexeme)들이 어휘규칙 등에 의해서 만들어지며, 이들은 관련 제약에 따라 통사 구조 *syn-st*에 나타날 수 있는 언어표현으로 투영된다.

4.1 어휘부 구조

어휘부 구조의 출발은 품사의 분류라 할 수 있다. KPSG에서의 한국어의 품사들은 (14)에서 살펴 보았지만, 전통적인 분류 방법과 크게 다르지 않게, 용언 verbal, 체언 nominal, 부사어 adverbial, 관형어 adnominal로 구분 한다. 이 중 용언 verbal의 유형위계 구조를 간략하게 살펴보면 다음과 같다.

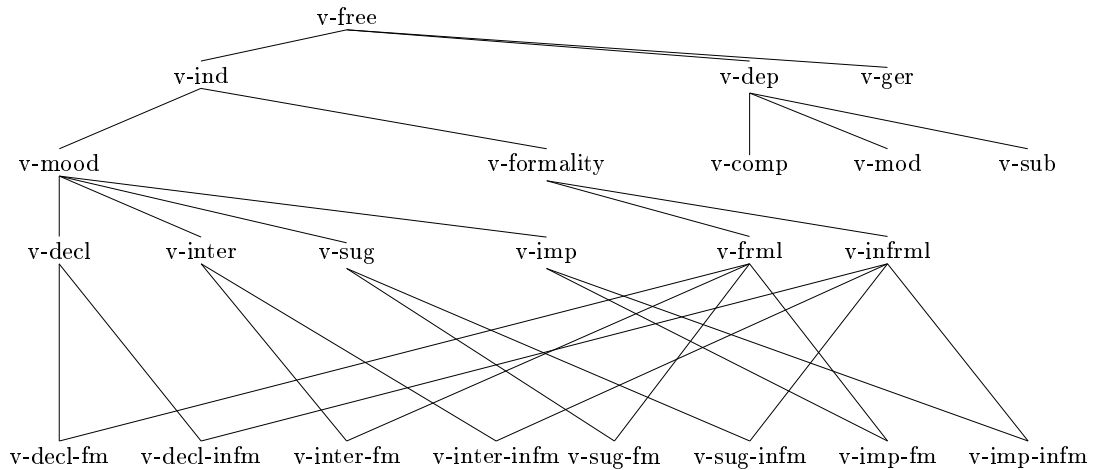


이와 같은 유형 구조는 한국어 용언 형성의 기본적 과정을 포착하기 위한 것이다. 굴절 언어인 한국어의 용언이 생성되는 과정은 엄격한 순차적 제약을 가지고 있으며, 통사부에 나타나기 위해서는 *v-free-stem*의 단계까지 만들어져야 한다.

- (21) a. [[[[[잡 + (히)]+(시)] +(있)]+ 다] + 고]
 b. *v-lxm* → *v-hon* (*v-stem1*) → *v-tns* (*v-stem2*) → *v-free* (*v-stem*) → *v-comp*

예를 들어 어근 ‘잡’이 통사부에 나타날 수 있는 자격을 가지는 어말어휘가 되기 위해서는 최소한 존칭접사(*v-stem1*)나 시제접사(*v-stem2*) 등과 결합하는 과정을 거칠 수도 있다. 이러한 과정을 거쳐서 만들어진 *v-free* 만이 통사부에서 나타날 수 있고, 이들의 종류를 살펴보면 다음과 같다.

(22)



즉 통사부에 나타날 수 있는 용언은 먼저 주절의 용언이 될수 있는 독립절 용언 어말을 가지는 v-ind(ependent), 종속적 용언 어말을 가진 v-dep(endent)로 구분된다. 특히 문장 서법을 나타내는 독립절의 용언들은 다시 서법과 형식성에 따라 다시 분류된다. 이러한 여러 하위 유형들이 가지는 제약을 LKB내 KPSG는 다음과 같이 정의한다.

```
v-ind := v-free &
[ SYN #syn & [ HEAD [ IC +,
                    NOMINAL -,
                    MOD <> ] ],
  SEM #sem,
  ARGS < [ SYN #syn, SEM #sem ] > ].
```

```
v-decl := v-mood &
[ SYN.HEAD.MOOD decl,
  SEM.MODE statement,
  ARGS < v-stem2 > ].
```

```
v-inter := v-mood &
[ SYN.HEAD.MOOD inter,
  SEM.MODE question,
  ARGS < v-stem2 > ].
```

...

이러한 제약들은 ‘먹다, 먹니, 먹자, 먹어라’나 혹은 ‘먹습니다, 먹습니까, 먹읍시다, 먹으시오’와 같은 어말의미를 형성하게 된다. 이들은 어휘들은 자신들의 상위 유형이 가지는 제약뿐만 아니라 자신이 속한 유형이 제약들을 준수하게 된다. 예를 들어 ‘먹다’는 v-decl-infm 유형이므로 v-ind의 제약뿐만 아니라 v-decl 제약도 상속하게 되기 때문에, 최소한 다음과 같은 제약들을 상속하게 된다.

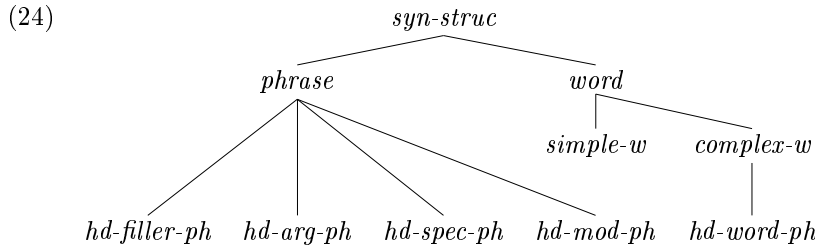
(23)
$$\left[\begin{array}{c} \text{verb} \\ \text{IC +} \\ \text{SYN.HEAD } \left[\begin{array}{c} \text{NOMINAL -} \\ \text{MOD } \langle \quad \rangle \\ \text{MOOD } \textit{decl} \end{array} \right] \\ \text{SEM.MODE } \textit{statement} \end{array} \right]$$

4.2 통사부 구조

4.2.1 한국어 X' Syntax

어휘부와 마찬가지로 정형의 통사구조 역시 유형에 따라 구(phrase)와 어휘 요소인 단어(word)로

대별 된다.

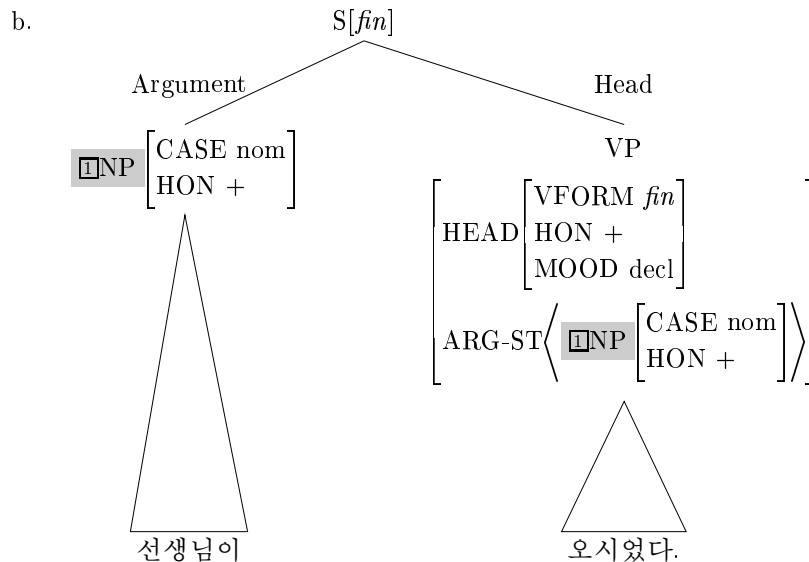
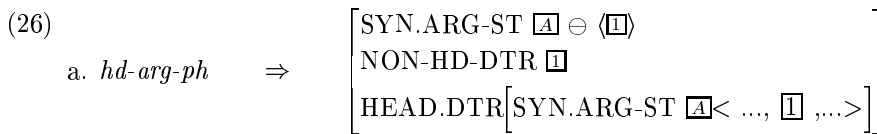


위의 위계 구조에서 구 phrase 유형은 한국어에서의 정형의 구구조를 나타내는 X' 구조를 의미한다. 즉 한국어가 허용하는 구들의 종류를 의미하며, 이들을 간단하게 나타내면 다음과 같다.

- (25) Korean X' Syntax (simplified):
- a. hd-arg-ph: $XP \rightarrow \boxed{A}, H[ARG-ST < \dots \boxed{A} \dots >]$
 - b. hd-spec-ph: $XP \rightarrow [SPEC \boxed{A}], H[\boxed{A}]$
 - c. hd-mod-ph: $XP \rightarrow [MOD \boxed{A}], H[\boxed{A}]$
 - d. hd-filler-ph: $XP \rightarrow \boxed{A}, H[GAP < \boxed{A} >]$
 - e. hd-word-ph: $X[LEX +] \rightarrow [word], H$

이러한 구들은 한국어에서 허용되는 통사적 구성소들의 구성 조건이라 할 수 있다. 이러한 조건들은 물론 자질구조들로 형식화 되어 LKB 내에서 구현될 수 있다. 구체적인 예와 이들 구들이 가지는 제약들을 자질구조로 나타내면 다음과 같다.

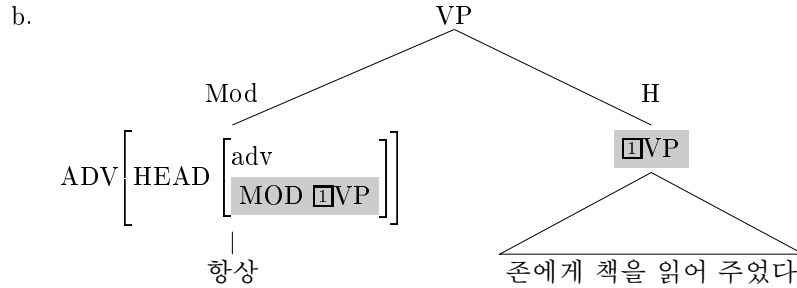
- 논항과 이를 선택하는 핵어로 이루어진 구: hd-arg-ph (head-argument-phrase)



위의 자질구조의 제약은 핵어가 요구하는 논항 중의 한 요소와 이 핵어가 결합하였을 때 정형의 구를 형성할 수 있다라는 것을 의미한다. 예를 들어 '오시었다'라는 동사는 어휘적으로 존칭 주어를 논항으로 취하고 있고, (26)에서 이러한 요구 조건을 만족하는 '선생님이'와 결합하여 hd-arg-ph를 이룬다.

- 수식어와 핵어로 이루어진 구: hd-mod-ph (head-modifier-phrase)

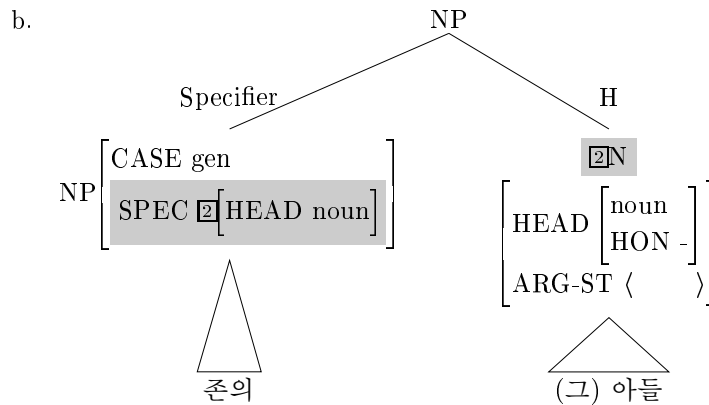
(27) a. $hd-mod-ph \Rightarrow \left[\begin{array}{l} SYN.HEAD.SPEC \langle \rangle \\ NON-HD-DTR [SYN.HEAD.MOD \boxed{\square}] \\ HEAD.DTR \boxed{\square} \end{array} \right]$



유형 *hd-mod-ph*에 부여된 기본적인 제약은 핵어를 수식하는 수식어구가 이 핵어와 결합하면 정형의 구를 이룬다는 것이다. (27)에서 부사 ‘항상’은 동사구를 수식하는 어휘적 정보를 가지고 있고, 이를 충족시키는 동사구와 결합하여 정형의 *hd-mod-ph*를 형성한다.

- 명시어와 핵어로 이루어진 구: *hd-spec-ph* (head-specifier-phrase)

(28) a. $hd-spec-ph \Rightarrow \left[\begin{array}{l} SYN.HEAD[NOMINAL +] \\ NON-HD-DTR [SPEC \langle \boxed{\square} \rangle] \\ HEAD.DTR \boxed{\square} \end{array} \right]$

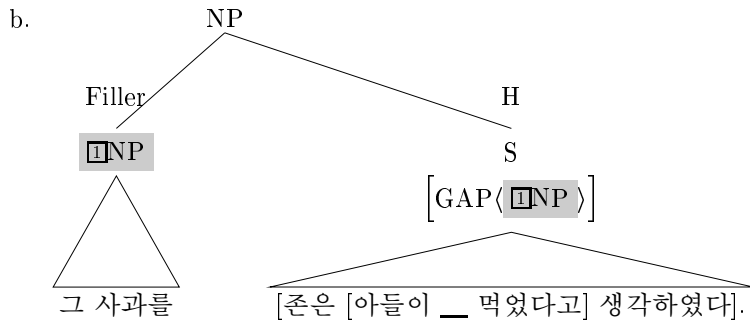


명시어와 핵어로 이루어진 구는 정형이라는 위의 제약은 ‘존의’와 같은 명시어가 핵어 ‘그 아들’이나 ‘아들’과 같은 명사구와 결합하여 정형의 구를 이루도록 한다.⁵

- 충전소와 핵어로 이루어진 구: *hd-filler-ph* (head-filler-phrase)

(29) a. $hd-filler-ph \Rightarrow \left[\begin{array}{l} SYN.VAL.GAP \langle \quad \rangle \\ NON-HD-DTR \boxed{\square} \\ HEAD-DTR.SYN.VAL.GAP \langle \boxed{\square} \rangle \end{array} \right]$

⁵이 구는 *hd-mod-ph*와 여러 가지 면에서 유사하지만, 많은 차이점들도 가지고 있다. 5.4 참조.

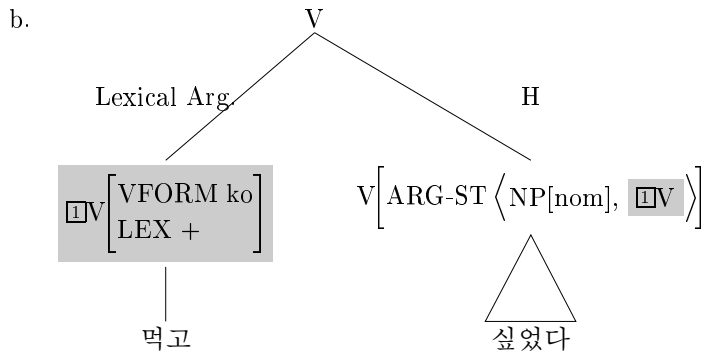


주제화나 관계화 구문에서와 같이 충전소(filler)와 공소(gap)가 인접한 위치에 있지 않는 경우의 구를 생성하기 위한 제약이다. 이 경우 한 공소를 포함하고 있는 핵어가 이 공소와 동일한 통사, 의미적 정보를 가진 충전소와 결합했을 경우에 정형의 구가 된다는 것을 의미한다. (29) 예문에서는 충전소 ‘그 사과를’이 이를 공소 GAP 값으로 가지고 있는 핵어 문장과 결합하여 정형의 hd-filler-ph를 형성하고 있다.

- 어휘 요소와 핵어로 이루어진 구: hd-word-ph (head-word-ph)⁶

(30) a.

$$hd\text{-}word\text{-}ph \Rightarrow \left[\begin{array}{l} SYN.LEX + \\ NON\text{-}HD\text{-}DTR \textit{word} \\ HEAD.DTR \textit{word} \end{array} \right]$$



한국어는 영어와 달리 두 어휘 요소가 결합하여 구보다는 적은 하나의 통사적 단위를 이룬다. 특히 이러한 단위들은 복합술어(complex predicate) 구문에서 쉽게 찾아 볼 수 있다. (30)의 예는 보조용언인 ‘싶었다’가 본동사 ‘먹고’와 결합하여 하나의 통사적 단위를 이루고 있는 경우를 보여준다.

4.2.2 보편 제약

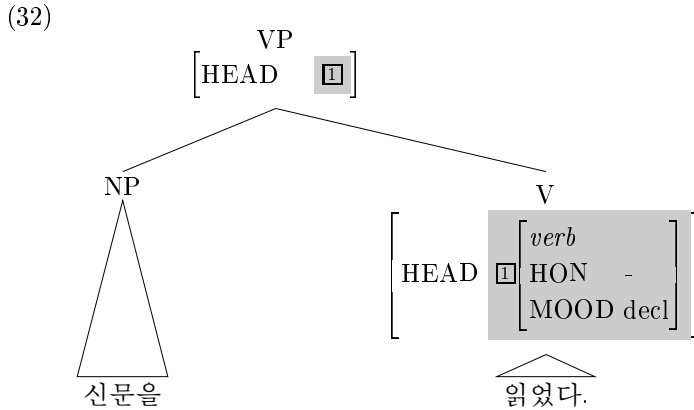
지금까지 살펴본 이러한 정형의 구들의 상위 유형 syn-struct는 보편적 제약인 핵어자질원리 Head Feature Principle(HFP)과 항가 항가 원리 Valence Principle(VALP)을 준수한다. HFP 제약은 핵어의 핵자질 HEAD 값들은 모범주(mother)로 투영되어야 한다는 것이다.

(31) Head Feature Principle:

$$syn\text{-}struct \Rightarrow \left[\begin{array}{l} HEAD \quad \square \\ HD\text{-}DTR \quad \left[\begin{array}{l} HEAD \quad \square \end{array} \right] \end{array} \right]$$

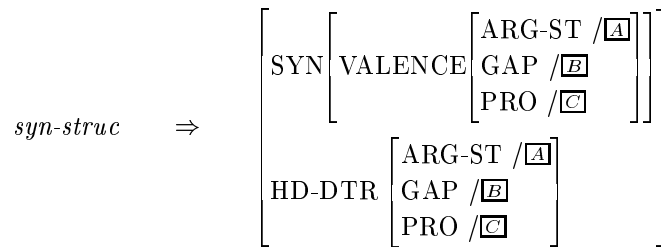
⁶ 엄격하게 따지자면, 이 처럼 두 어휘 요소가 결합하여 다른 어휘적 요소를 이룬다는 점에서 phrase라 할 수 없고, hd-word-str가 보다 적합한 용어라 할 수 있다. 통사적 구조내에서 이루어지는 모든 결합을 동일한 이름으로 표현하기 위해서 이와 같은 용어를 사용한다.

예를 들어 (32)에 주어진 것과 같이 동사 ‘읽었다’의 HEAD 자질은 HFP에 따라 VP의 HEAD 자질과 동일하게 된다.



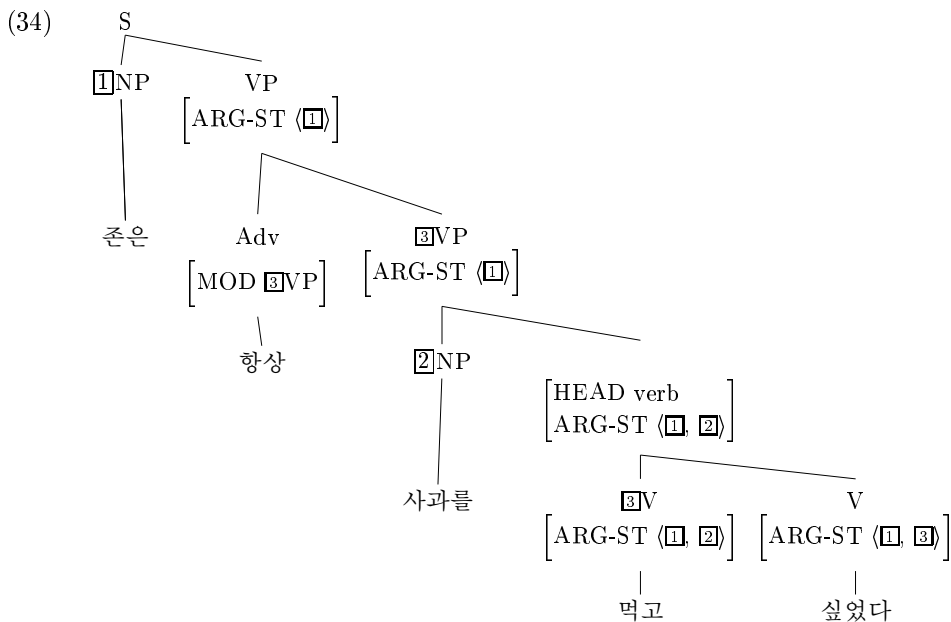
항가원리에 관한 제약은 기본적으로 범주문법(Categorial Grammar)에서의 함수적용(functional application)과 유사하게 핵어가 논항과 결합했을 때 이를 취소하는 기능을 한다. 핵어의 논항구조 ARG-ST를 포함한 항가 VALENCE 값은 기본적(default)으로는 자신의 모범주와 동일하다.

(33) Valence Principle:



항가 VALENCE 값에 관한 기본 값 default는 하위유형에 의해서 변화될 수 있다. 예를 들어 hd-arg-ph와 같은 경우에서 논항과 결합하는 경우나 혹은 hd-filler-ph 처럼 논항의 한 요소가 GAP으로 실현되는 경우에는 이에 항가 값이 변경된다.

지금까지 살펴본 이러한 X' 구조에 관한 제약, HFP, VALP 제약들은 관련 표현이나 구조에서 모두 준수되어야만 한다. 한 예를 살펴 보자.



위 문장의 구성 단계를 bottom-up 방향으로 살펴보면, 동사 ‘싶었다’는 주어와 본동사를 논항으로 취하며, 먼저 본동사 ‘먹고’와 결합하여 hd-word-ph를 이룬다. 이 결과는 다시 ‘먹고’의 목적어인 ‘사과를’과 결합하여 hd-arg-ph를 이루고, 다시 수식어 ‘항상’과 결합하여 hd-mod-ph를 이룬다. 그리고 이러한 동사구는 마지막으로 주어 논항과 결합하여 hd-arg-ph를 이룬다. 모든 구들이 문법이 허락하는 정형의 구들이다. 또한 최 상위 S의 HEAD 값은 하위 구조의 핵어에서 상속되어 HFP도 준수하고 있다. 그리고 VALP 원리가 요구하는 것과 같이, 각 핵어의 논항구조는 논항과 결합하였을 때, ARG-ST의 값에서 알 수 있듯이 그 값이 취소되어 있다. HFP와 VALP가 각 구조에서 준수되어 있다. 여기서 주목할 것은 관련 제약들이 준수되고 있는 여부를 전체 문장을 살펴보는 것이 아니라 각각의 국부(local)구조에서만 확인한다는 것이다. 각각의 국부 구조가 모든 제약들을 준수하였을 때, 전체 문장이 완성된다.

5 한국어 주요 구문의 구현

5.1 Basic Sentence Types

한국어 기본 문형은 용언인 동사나 형용사가 취하는 논항의 종류에 따라 여러 가지로 구분 될 수 있다.

- (35) a. 비가 왔다.
 b. 존이 사장이 되었다.
 c. 도둑이 잡혔다.
 d. 존이 병에 걸렸다.
 e. 존이 그 책을 읽었다.
 f. 존은 메리가 그 책을 읽었다고 생각하였다.
 g. 존은 사장에게 해고를 당했다.
 h. 존이 그 책을 메리에게 주었다.
 i. 존이 메리에게 영어를 공부하도록 설득했다.
- (36) a. 메리가 예쁘다.
 b. 존에게 책이 많다.
 c. 존이 돈이 없다.
 d. 존이 메리가 무섭다.
 e. 존이 학생이다.

이러한 기본 문장들의 생성은 어휘부의 정보에서 투영된다. 예를 들어 자형용사인 ‘예쁘다’는 a-intr로 아래의 LKB내의 기술처럼 논항이 하나라는 제약을 가지지만, v-np-tr 유형에 속하는 ‘읽다’와 같은 타동사는 목적격 NP도 함께 취하는 어휘적 정보를 가지고 있다.

```
a-intr := stative-v &
[ SYN.VAL.ARG-ST < [ ] > ].
```

```
v-np-tr := v-tr &
[ SYN.VAL.ARG-ST < [ ],
  [ SYN.HEAD.CASE acc ] > ].
```

이러한 어휘 요소가 논항들과 결합할 수 있도록 허가하는 것은 4장에서 살펴본 한국어 X' 규칙 중의 하나인 hd-arg-ph의 제약이다. 아래는 이를 LKB 내에 자질 기술로 구현시킨 일 부분이다.

```
head-arg-rule-1 := hd-arg-ph &
[ SYN.VAL.ARG-ST #2,
  ARGS < #1 & [ SYN.HEAD.PRD - ],
  syn-st & [ SYN.VAL.ARG-ST [ FIRST #1,
    REST #2 ] ] > ].
```

즉 ARGS의 주 요소중 후자는 핵어를 나타내고, 이 핵어는 논항으로 #1과 #2 두 요소를 취한다. 이 핵어가 두 요소 중 #1과 결합하게 되면, 상위 부분 ARG-ST 값은 #2만 남게 된다. 이는 결국 어휘 핵의 논항들이 하나 씩 채워지게 되며, 논항구조에서 제소(discharge) 된다는 것을 의미한다.

5.2 Basic Sentences with Adverbs

한국어 부사는 통사적 위치에 따라 크게 두 가지 유형으로 구분할 수 있다. ‘매일, 항상’과 같이 위치에 상관 없이 모든 용언들을 (V, VP, or S)을 수식 할 수 있는 부사와 ‘잘, 좀, 더, 다’와 같이 어휘적 요소 앞에서만 나타날 수 있는 부사로 나눌 수 있다.

- (37) a. 존은 매일 학생들에게 영어를 가르쳤다.
 b. 존은 학생들에게 매일 영어를 가르쳤다.
 c. 존은 학생들에게 영어를 매일 가르쳤다.
- (38) a. 존이 밥을 잘 먹었다.
 b. 존이 밥을 다 먹었다.

부사의 이러한 분배 제약을 포착하기 위해서 LKB내의 KPSG는 다음 처럼 부사어 adverbial에 관한 일반적인 제약을 설정하고 두 하위 유형 adv-phmod와 adv-wmod도 설정한다.

```
adverbial := lexeme &
[ SYN [ HEAD adv & [ MOD < [ SYN.HEAD verb,
                           SEM.INDEX #index ] > ],
  VAL [ ARG-ST <>,
        PRO <! !> ] ],
SEM [ INDEX event & #index,
      RELS [ LIST.REST #last,
            LAST #last ] ] ].
```

```
adv-phmod := adverbial.
```

```
adv-wmod := adverbial &
[ SYN.HEAD.MOD < simple-w & [ SYN.HEAD.AUX - ] > ].
```

위 기술에 따르면, 부사어 adverbial는 통사적 정보로는 [HEAD verb]인 요소를 수식하지만, 논항을 전혀 취하지는 않는다. 한편 구를 수식하는 adv-phmod에는 추가적인 제약이 없지만, 어휘적 요소만 수식하는 adv-wmod는 수식하는 요소가 본동사이어야 하며 동시에 어휘적 요소라야 한다는 자신 고유의 제약을 가지고 있다. 이러한 제약은 기본적으로 (39) 같은 비문을 막게 된다.

- (39) a. *존이 밥을 먹고 [잘 싶다].
 b. *존이 잘 밥을 먹었다.

이러한 구문처리 방법은 결국 한국어의 통사구조를 이분지(binary) 구조만 허용하게 되는 결과를 가져온다. 이러한 분석은 전산 처리에서 어려운 난제 중의 하나인 한국어 문장내의 뒤섞이(scrambling) 현상을 효율적으로 처리할 수 있게 해준다. 예를 들어 아래와 같은 5가지의 통사적 요소로 이루어진 문장이 허용하는 어순의 종류는 24(4!) 가지나 된다.

- (40) a. 매일 존은 학생들에게 영어를 가르쳤다.
 b. 존은 매일 학생들에게 영어를 가르쳤다.
 c. 학생들에게 존은 매일 영어를 가르쳤다.
 d. 영어를 존은 매일 학생들에게 가르쳤다.
 e. 존은 학생들에게 영어를 매일 가르쳤다.
 f. 영어를 매일 존은 학생들에게 가르쳤다.
 g. ...

hd-arg-ph가 기본적으로 논항구조 ARG-ST 내의 아무 요소와 결합하도록 허용하고, 또한 hd-mod-ph는 수식어구와 수식어가 결합하도록 하기 때문에 KPSG에서는 이동과 같은 통사적 규칙을 설정하지 않고도 이러한 모든 어순들이 효율적으로 처리된다.

5.3 Case

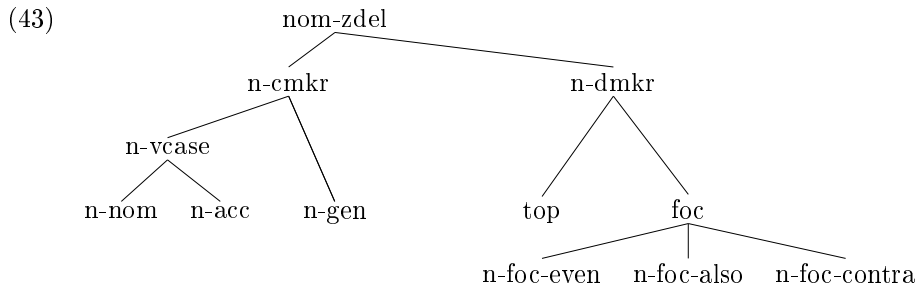
한국어 격은 명시적으로 실현될 수 있지만, 적절한 문맥이 주어지면 선택적으로 나타나지 않거나 조사로도 대체될 수 있다.

- (41) a. 존이 메리를 만났다.
 b. 존은 메리를 만났다.
 c. 존은 메리 만났다.

KPSG에서 격조사는 독립적인 어휘요소로 취급하지 않고 파생접사와 유사하게 취급한다는 점에서 기존의 문법과 차이가 있다 (Cho and Sells 1995, Kim 1997). 즉 격조사와 명사가 두 개의 독립적인 단어로 존재하는 것이 아니라 하나의 어휘 요소로 분석된다. 한국어 명사 어형은 (42)의 예시 처럼 기본적으로 어간-접미사-조사로 이루어져 있고, 엄격한 순서를 가지고 있다.

- (42) a. 선생-님-들-께서-만-은
 b. *선생-님-만-들-은-께서

격조사는 마지막 조사와 상보적(complimentary) 관계로 나타난다. 이러한 속성을 포착하기 위한 일환으로 KPSG에서는 먼저 격조사가 실현된 체언을 nom-zdel (nominal-zdelimiter) 유형으로 정의하고 이를 다시 ‘-이/가/의’ 격조사와 함께 나타난 경우 n-cmkr (nominal-case-marker)로, ‘-는/도’ 처럼 (z-delimiter) 담화표지어로 실현된 경우 n-dmkr (nominal-discourse-marker)로 구분한다. 이를 유형 위계 구조로 나타내면 다음과 같다.



그리고 LKB 내에 기술된 이들 하위 유형들이 취할 수 있는 CASE 값이나 DMKR 값은 다음과 같이 정의된다.

```

noun := pos &
  [ MOD    <>,
    PLU    bool,
    CASE    case,
    DMKR    d-marker ].
  
```

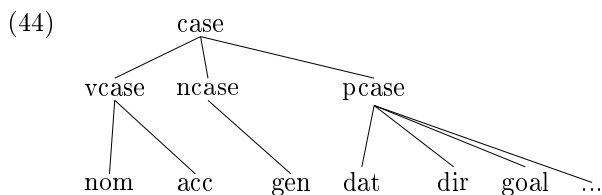
```

n-nom := n-vcase &
  [ SYN.HEAD.CASE nom ].
  
```

```

n-acc := n-vcase &
  [ SYN.HEAD.CASE acc ].
  
```

여기서 주목할 것은 일반 명사 noun은 CASE 값으로 case 값을 갖지만, 조사가 붙은 경우는 아래 유형위계 처럼 nom, acc, dat, gen 등으로 나타난다는 것이다.



이러한 기본적 시스템과 함께 자질 통합(unification)과 포섭(subsumption)이 격 실현이나 생략 현상을 설명하는 데 중요한 역할을 한다. case 값은 상위 유형이기 때문에 아래 (45)a는 (45)b를 포섭한다. 즉 어떤 요소가 (45)b와 같은 정보를 요구하는 경우에 실질적으로 (45)a와 같이 실현되어도 자질 통합이 일어날 수 있다는 것이다.

- (45) a. [SYN.HEAD.CASE *case*]
 b. [SYN.HEAD.CASE *nom*]

즉 ‘만났다’와 같은 동사는 논항으로 (46)a 같은 정보를 요구하지만, (46)b와 같이 실현되어도 문제가 없게 된다.

- (46) a. [SYN.VAL.ARG-ST <NP[*nom*], NP[*acc*]>]
 b. [SYN.VAL.ARG-ST <NP[*case*], NP[*case*]>]

살펴본 바와같이 격이나 명사형에 관한 적절한 유형 위계 구성과 각 유형에 적합한 제약들을 설정, 자질구조의 기본적인 연산 작용인 통합과 포섭이 한국어 격 실현을 효과적으로 처리할 수 있게 해 준다.

5.4 Relative Clause

한국어 관계절의 주요 특징은 관계사(relative pronoun)가 없고, 관형어 역할을 하는 용언이 시제 정보에 따라 각각 다른 형태로 실현된다는 것이다.

- (47) a. [존이 __ 읽-은] 책
 b. [존이 __ 읽-던] 책
 c. [존이 __ 읽-을] 책

또한 관계절의 공소(gap)는 관계절 내보다 더 깊은 위치인 관계절 내의 내포문에 존재할 수도 있는 장거리(long-distance) 의존 관계를 가질 수도 있다.

- (48) a. [존이 [메리가 __ 먹었다고] 말한] 사과
 [존이 [메리가 __ 읽었다고] 믿는] 책

KPSG내의 관계절 분석은 기본적으로 두 가지로 구분된다. 먼저 관형어 어미를 가진 동사들은 수식 자질인 MOD 값을 갖게 되는 것과, 어휘 핵의 논항 중 한 요소가 공소 GAP으로 실현되어 논항으로 결합하지 않아도 되는 규칙이다. 관련 문법 기술을 살펴보면 아래와 같다.

```
v-mod := v-dep &
[ SYN #syn & [ HEAD.MOD < [ SYN.HEAD noun ] > ],
  SEM.RELS [ LIST [ FIRST [ ARGO #s,
                        ARG1 #u ],
                    REST.FIRST [ PRED "now",
                                ARGO #u ],
                    REST.REST #list ],
            LAST #last ],
  ARGS < v-stem1 & [ SYN #syn,
                    SEM [ INDEX #s,
                          RELS [ LIST #list,
                                LAST #last ] ] ] > ].
```

```
binary-start-gap-rule-1 := binary-sg &
[ SYN.VAL [ GAP <! #1 !>,
            ARG-ST <> ],
  ARGS < #2 & [ SYN.HEAD.PRD - ],
        syn-st & [ SYN.VAL.ARG-ST < #1, #2 > ] > ].
```

위의 자질 기술을 보면 v-mod는 명사를 수식하며 시제 정보를 가지는 것으로 명시되어 있다. binary-sg(start-gap)은 논항 중 한 요소인 #1이 GAP으로 실현되게 되면 논항 조건이 만족된다는 것을 보여주고 있다.⁷

이와 같은 관계절 처리 방법은 KPSG 문법의 다른 부분과 상호 작용하여 여러 관련 관계 구문들을 처리할 수 있다. 예를 들어 한국어는 (49)a처럼 소위 말하는 ‘소유관계절(Genitive relative clauses)’을 가지고 있다.

- (49) a. [존의 [멋진 옷]]
 b. [멋진 [그 신사의 옷]]

KPSG에서는 [멋진 옷]이 관계절 조건에 의해서 형성되어 명사구를 이루고, 관형어 ‘존의’가 이 명사구를 수식하는 정형의 구가 생성된다.

5.5 Complex Predicate Construction

한국어 구문 중 가장 흔히 찾아 볼 수 있는 구문 중의 하나는 복합술어 구문이다. 복합술어 구문은 기본적으로 두 개의 독립된 술어가 결합하여 마치 하나의 술어처럼 행동하는 경우들로, 여기서는 조동사 구문과 경동사 구문이 어떻게 KPSG내에서 처리되는지를 간단하게 살펴보기로 한다.

5.5.1 Auxiliary Constructions

조동사 구문은 본동사-조동사의 엄격한 어순 순서를 지키며, 본 동사는 특정한 어말어미 형태를 취해야한다. 또한 이 두 동사는 마치 하나의 어휘 요소처럼 분리될 수 없는 밀접한 관계를 가지고 있다.

- (50) a. 존은 사과를 먹고/*어 싶었다.
 b. *먹고 사과를 존은 싶었다.

이러한 점에서는 조동사가 본동사를 선택하지만, 문장 전체의 논항의 종류와 수를 결정하는 것은 본동사이다.

- (51) a. 존은 편지를 휴지통에 넣어 버렸다.
 b. 존은 술어 버렸다.

위 예문 (51)에서 처럼 조동사 ‘버리다’는 논항의 종류나 형태에 영향을 끼치지 않는다. 본동사 ‘넣다’와 ‘술다’가 전체 문장의 논항의 종류를 결정하는 것이다.

이러한 기본적인 속성을 포착하기 위해서 KPSG는 논항 합성(argument composition)을 이용한다. 논항합성은 hd-wd-ph가 어휘 요소와 이를 논항으로 취하는 핵어로 구성된 경우에 적용된다. 논항 합성에 관한 KPSG 기술은 다음과 같다.

```
head-wd-arg-rule-1 := hd-wd-ph &
[ SYN.VAL.ARG-ST #argst,
  ARGS < word & #2 & [ SYN.VAL.ARG-ST #argst & [ FIRST #1 ] ],
  v-word & [ SYN.VAL.ARG-ST < #1, #2 > ] > ].
```

위 규칙이 의미하는 것은 ARGS에 있는 두 요소 중, 후자 조동사는 논항으로 주어 #1과 본동사 #2를 취하며, 이 본동사 역시 동일한 주어 #1을 취하는 것으로 명시되어 있다. 이 두 요소가 결합을 하게 되면, ARG-ST는 본동사의 논항구조의 요소 #argst만 요구하게 된다는 것을 명시하고 있다. 이는 결국 핵어 조동사가 본동사와 결합하고 이 본동사가 취하는 논항의 값을 자신이 가져와 합성하는 결과와 마찬가지로이다.⁸

⁷관계절은 일반 hd-mod-ph와는 달리 수식 대상이 noun이라는 부가적인 제약을 가진다. 이를 위해서 수식-핵어 구 hd-mod-ph의 하위유형으로 hd-rel-mod-ph를 설정하거나, hd-rel-mod-rule을 설정할 수 있다. 현재 KPSG에서는 후자를 선택하고 있다.

⁸그러나 조동사가 논항에 전혀 영향을 끼치지 않는 것은 아니다. ‘주다’와 같은 조동사는 여격 논항을 첨가한다.

- (i) 존은 메리에게 책을 읽어 주었다.

위 문장에서 논항 ‘메리에게’는 조동사 ‘주다’에 의해서 부가된 것이다. 이러한 경우를 처리하기 위해서는 부가적인 규칙이 필요하다.

5.5.2 Light verb constructions

경동사 구문도 조동사 구문과 많은 속성을 공유한다. 차이점은 단지 경동사의 어휘적 논항이 본동사가 아니라 서술명사(verbal noun: vn)라는 것이다.

- (52) a. 그 회사는 미국에 자동차를 수출(을) 하였다.
b. 존은 물리학을 공부(를) 하였다.

이러한 경동사 구문들을 처리하기 위해서 KPSG는 경동사 ‘하다’의 어휘 정보를 다음과 같이 설정한다.

```
light-v := pos-main-v &
[ SYN.VAL.ARG-ST < phrase & [ SYN.HEAD.CASE nom,
                               SEM.INDEX #arg1 ],
  n-word & [ SYN.HEAD.VERBAL +,
             SEM.INDEX #arg2 ] >,
  SEM.KEY [ ARG1 #arg1,
            ARG2 #arg2 ] ] .
```

위 어휘 정보에 따르면 ‘하다’ 경동사는 주어와 동사성을 가진 명사 n-word 즉 서술 명사를 논항을 취한다. 이는 곧 경동사가 이 어휘적 논항과 결합하게 되면 조동사 구문에서와 마찬가지로 hd-wd-ph를 형성하게 되고, 논항 합성이 일어난다는 것을 의미한다.

6 맺는 말

지금까지 제약기반이론을 이용하여 한국어구구조문법을 구축하고 이를 전산학적으로 어떻게 구현시킬 수 있는가를 살펴 보았다. 유형화된 자질구조와 다중상속 위계는 한국어구구조 문법을 간결하면서 정확하게 구축할 수 있게 해 주었다. 특히 관련 언어 표현 즉 유형들이 취하는 여러 제약들이 상호작용하여 문법성 정도를 판단하여 줄 수 있다는 것도 살펴 보았다. 이러한 제약기반관점에서 구축된 문법은 LKB 시스템에 적용, 구현하는 데에도 큰 어려움이 없다는 것도 살펴 볼 수 있었다. 비록 여기서 의미부에 관한 자세한 설명을 제공하지 못했지만, 이러한 문법은 의미 합성까지 효율적으로 처리할 수 있게 해 준다는 점에서 전산학적 적용 가능성을 높혀 준다.

이러한 한국어구구조문법의 컴퓨터 구현이 가질 수 있는 긍정적 측면을 요약하면 다음과 같다.

- 한국어 문법을 위해 설정한 가설들의 정확성을 검증할 수 있고 개발 시킬 수 있다.
- 한국어 문장 처리를 위한 효율적 방법을 제공할 수 있다.
- 언어학적으로 정확할뿐만 아니라 경험적/이론적 바탕위에 출발한 KPSG는 실제 상황에 응용할 수 있는 여러 가지 방향성을 제공한다.

본 연구는 10여년 전부터 이러한 유사 시스템내에서 문법개발 및 구문분석을 꾸준히 진행해 온 영어에 비해 처리할 수 있는 경험적 자료들의 종류와 량에 극히 제한되어 있다. 그러나 구축된 KPSG를 몇몇 주요 구문에 적용, 처리해 본 결과는 상당히 고무적이라 할 수 있다.

References

- 고창수. 2002. “자질연산문법이론.” 도서출판 월인.
임홍빈, 이홍식 외. 2002. “한국어 구문분석 방법론.” 한국문화사.
홍윤표 외. 2002. “한국어와 정보화.” 태학사.
Chung, Chan and Jong-Bok Kim. 2003. Internally Headed Phrase Structure Grammar. In J. Kim and Stefan Wexler (eds.), *The Proceedings of the 9th International Conference on Head-Driven Phrase Structure Grammar*. (Book URL: <http://csli-publications.stanford.edu/HPSG/3/hpsg02.htm>) Stanford: CSLI Publications.
Chung, Chan and Jong-Bok Kim. 2002. Korean Copula Constructions: A Construction and Linearization Perspective. *Korean Journal of Linguistics* 27.2: 171-193.

- Chan Chung, Jong-Bok Kim, Byung-Soo Park, and Peter Sells. 2001. Mixed Categories and Multiple Inheritance Hierarchies in English and Korean Gerundive Phrases. *Language Research* 37.4 763–797.
- Cho, Young-Mee Yu, and Peter Sells. 1995. A lexical account of inflectional suffixes in Korean. *Journal of East Asian Linguistics* 4, 119-174.
- Cho, Sey-youn and Jong-Bok Kim. 2002. Echoed Verb Constructions in Korean: A Construction-Based HPSG Analysis. *Korean Journal of Linguistics* 27.4: 661-681
- Copestake, Ann. 2002. *Implementing Typed Feature Structure Grammars*. CSLI Publications.
- Kim, Jong-Bok. 2001. A Constraint-Based and Head-driven Approach to Multiple Nominative Constructions. In Dan Flickinger and Andreas Kathol (eds.), *Proceedings of the HPSG-2000 Conference*. University of California, Berkeley. Stanford: CSLI Publications. (Book URL: <http://csli-publications.stanford.edu/HPSG/1/hpsg00-toc.html>)
- Kim, Jong-Bok. 2000. *The Grammar of Negation: A Constraint-Based Approach*. Stanford. CSLI Publications.
- Kim, Jong-Bok. 1999. Korean Short Form Negation and Related Phenomena: A Lexicalist, Constraint-Based Analysis. *Language and Information* 3.2.
- Kim, Jong-Bok. 1999. Constraints on the Formation of Korean and English Resultatives. In Pius Tamanji, Masako Hirotsu, and Nancy Hall (eds.), *NELS 29*: 137–151 University of Mass, Amherst, Mass.: GLSA
- Kim, Jong-Bok. 1999. A Local Treatment of Nonlocal Relativization. *Proceedings of the 26th Annual Meeting of the Berkeley Linguistic Society (BLS 26)*: 157 - 170. Berkeley, CA: Berkeley Linguistics Society
- Kim, Jong-Bok. 1998. A Head-driven and Constraint-Based Analysis of Korean Relative Clause Constructions: With a Reference to English. *Language Research* 34.4: 1–41
- Kim, Jong-Bok. 1998. On the Mixed Properties of Pseudo Relative Clauses. In B.S. Park and J. Yoon (eds), *Proceedings of the 11th International Conference on Korean Linguistics*, 83-93. ICKL
- Kim, Jong-Bok. 1998. Interface between Morphology and Syntax: A Constraint-Based and Lexicalist Approach. *Language and Information* 2: 177-233.
- Kim, Jong-Bok and Byung-Soo Park. 2000. The Structure of LFN and Argument Composition. *Language Research*, 36.4: 715-733
- Kim, Jong-Bok and Byung-Soo Park. 2000. Grammatical Interfaces in Korean Relatives. In Ronnie Cann, Claire Grover and Philip Miller (eds.), *Grammatical Interfaces in HPSG*, 153-170. Stanford: CSLI Publications.
- Kim, Jong-Bok and Ivan Sag. 2002. Negation Without Movement. *Natural Language and Linguistic Theory*, Volume 20, Number 2, 339-412.
- Oepen, Stephan, Dan Flickinger, Jun-ichi Tusujii, and Hans Uszkoreit. 2002. *Collaborative Language Engineering*. CSLI Publications.
- Pollard, Carl and Ivan Sag. 1994. *Head-driven Phrase Structure Grammar*. CSLI Publications.
- Sag, Ivan and Tom Wasow. 1999. *Syntactic Theory: A Formal Approach*. Stanford: CSLI Publications.
- Siegel, Melanie and Emily M. Bender. 2002. Efficient Deep Processing of Japanese. In *Proceedings of the 3rd Workshop on Asian Language Resources and International Standardization*. Coling 2002 Post-Conference Workshop. Taipei, Taiwan.
- Siegel, Melanie. 2000. HPSG Analysis of Japanese. In W.Wahlster(ed.), *Verbmobil: Foundations of Speech-to-Speech Translation*., Springer Verlag.