

LKB를 이용한 한국어 자유 어순의 구현

고 성 연
(언어학과 박사과정)

1. 머리말
2. 한국어의 자유 어순과 HPSG의 논항가 자질 구조
3. LKB에 기반한 한국어의 자유 어순 분석
 - 3.1. 단층구조분석(Flat Structure Analysis)
 - 3.2. 양분단선구조분석(Binary Linear Structure Analysis)
 - 3.3. 양분병렬구조분석(Binary Parallel Structure Analysis)
 - 3.3.1. 양분병렬구조분석의 논항가 자질 구조와 규칙
 - 3.3.2. LKB에 기반한 양분병렬구조분석의 구현
4. 양분병렬구조분석과 HPSG
5. 맺음말

1. 머리말

한국어의 문장 구성의 중요한 특징 중의 하나는 문장 성분들간의 순서가 비교적 자유롭다는 점이다.¹⁾

- (1)
 - a. 철수-가 영희-를 사랑한다.
 - b. 영희-를 철수-가 사랑한다.
- (2)
 - a. 철수-가 영희-에게 꽃-을 준다.
 - b. 철수-가 꽃-을 영희-에게 준다.
 - c. 꽃-을 철수-가 영희-에게 준다.
 - d. 꽃-을 영희-에게 철수-가 준다.
 - e. 영희-에게 철수-가 꽃-을 준다.
 - f. 영희-에게 꽃-을 철수-가 준다.

1) "문의 어순은 서술용언이 문말에 오고 기타 문성분의 위치는 비교적 자유롭다. 자연스러운 무표의 어순은 주어-가 먼저 오고 기타 보충어가 뒤따르는 경우이다." (장석진 1993:51)

(1)과 (2)는 각각 2항 동사(두 자리 서술어) 구문과 3항 동사(세 자리 서술어) 구문의 자유어순을 보여주는 예이다. (1a)은 [주어-목적어-타동사]로 이루어진 전형적인 타동사 구문인데, (1b)와 같이 [목적어-주어-타동사]의 순서로 이루어진 구문과 의미상 동치이다(화용론적인 차이는 편의상 논의에서 배제한다.) [주어-목적어-목적어2-수여동사]로 이루어진 (2a)도 (2b-f)와 의미상 구별되지 않는다.

이러한 한국어의 자유 어순을 통사론적으로 분석하기 위해서, 핵어중심 구구조문법(Head-Driven Phrase Structure Grammar, 이하 HPSG)의 논항가 자질(Valence Feature)을 한국어의 통사적 특징에 맞도록 수정하고, 수정된 논항가 자질을 LKB(Linguistic Knowledge Building)에 반영하여 구현(Implementation)이라는 관점에서 그 타당성을 검증하고자 한다.

논문의 구성은 다음과 같다. 2절에서는 Sag et al.(2003)의 논항가 자질이 한국어의 분석에 타당하지 않음을 보이고, 한국어 자유 어순에 대한 몇 가지 접근법을 소개한다. 3절에서는 한국어의 분석에 적합한 새로운 논항가 자질 구조(Valence Feature Structure)와 규칙을 제안하고 이를 바탕으로 한 LKB 문법 구현을 보인다. 4절에서는 3절에서 제안한 문법이 기존 HPSG(Sag et al. 2003) 이론에 대해 갖는 함의를 살펴본다.

2. 한국어의 자유 어순과 HPSG의 논항가 자질 구조(Valence Feature Structure)

먼저, 논의의 편의를 위해 Sag et al.(2003:279)의 핵어-명시어 규칙(Head-Specifier Rule, 이하 HSR)과 핵어-보충어 규칙(Head-Complement Rule, 이하 HCR)을 옮겨 놓는다.

(3) Head-Specifier Rule

$$\begin{bmatrix} \textit{phrase} \\ \text{SPR} < > \end{bmatrix} \rightarrow \boxed{1} \quad \text{H} \begin{bmatrix} \text{SPR} < \boxed{1} > \\ \text{COMPS} < > \end{bmatrix}$$

A phrase can consist of a (lexical or phrasal) head preceded by its specifier.

(4) Head-Complement Rule

$$\left[\begin{array}{l} \textit{phrase} \\ \text{COMPS} < > \end{array} \right] \rightarrow \text{H} \left[\begin{array}{l} \textit{word} \\ \text{COMPS} < \boxed{1}, \dots, \boxed{N} > \end{array} \right] \boxed{1} \dots \boxed{N}$$

A phrase can consist of a lexical head followed by all its complements.

위의 두 규칙에 따르면, 영어는 반드시 보충어(complement)들이 동사구와 먼저 결합하고 나서야 비로소 명시어(specifier)가 결합할 수 있다(규칙순: (4)→(3))²⁾. 영어와 달리 한국어는 동사구의 핵어(head)가 보충어(complement)에 뒤따르는 head-final language이므로, 규칙 (4)를 다음과 같이 수정할 수 있다.

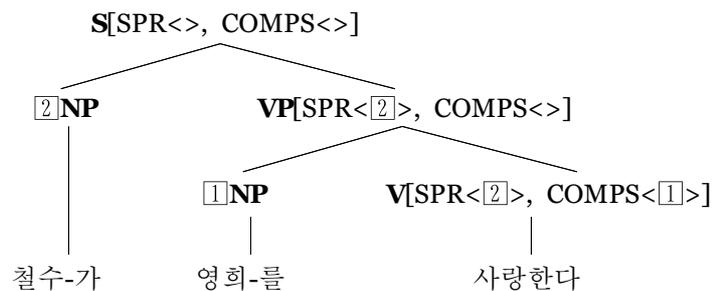
(4') Head-Complement Rule (revised for Korean)

$$\left[\begin{array}{l} \textit{phrase} \\ \text{COMPS} < > \end{array} \right] \rightarrow \boxed{1} \dots \boxed{N} \text{H} \left[\begin{array}{l} \textit{word} \\ \text{COMPS} < \boxed{1}, \dots, \boxed{N} > \end{array} \right]$$

A phrase can consist of a lexical head followed by all its complements.

위 (3)과 (4')의 규칙에 의해 (1a) 철수가 영화를 사랑하는다의 구조를 분석하면 다음과 같다.

(5) (1a) 철수가 영화를 사랑하는다의 수형도



2) 이러한 표현은 설명의 편의를 위한 것일 뿐이며, Serialism에 입각한 규칙의 순차적 적용을 의미하는 것은 아니라는 점을 밝혀둔다.

그러나 (3)과 (4')에 따르면 (1b) 영화-를 철수-가 사랑한다와 같은 한국어 문장은 다음과 같은 이유로 인해 인허(license)되지 않는다.

- (i) 어휘 핵어(lexical head)인 V('사랑한다')는 COMPS value가 *non-empty*이므로 규칙 (3)의 구조기술을 만족하지 않는다. (HSR 적용 불가)
- (ii) (i)에서 VP('철수-가 사랑한다')가 인허된다고 하더라도 이때의 VP는 *word*가 아닌 *phrase*이므로 규칙 (4')의 구조기술을 만족하지 않으므로, '영화-를'과 결합할 수 없다. (HCR 적용 불가)

이러한 문제를 해결하기 위해서 다음과 같은 접근법이 가능할 것이다.

- (6) a. GAP Analysis
- b. Flat Structure Analysis
- c. Word-order Domain approach

(6a)는 Sag et al(2003)에서 원거리 의존관계(Long Distance Dependency)를 다루기 위해 제안한 GAP 자질을 도입하는 분석 방법이다. 즉, (1b) 영화-를 철수-가 사랑한다를 *Mary, John loves*와 같은 주제화된 문장(topicalized sentence)으로 간주하여 처리하는 것이다.

(6b)는 영어의 계층적 구조와 달리 한국어의 문장 구조는 단층구조(flat structure)로 이루어져 있다고 보고, 논항가 자질의 하위 구분(SPR과 COMPS의 구분)을 하지 않는 분석 방법이다. 주어도 보충어의 하나로 보고, 모든 보충어를 동사와 직접 평면적으로 결합하는 규칙을 가정한다.³⁾

(6c)는 어순을 관장하는 문법 영역(domain)을 별도로 설정하여, (1a)와 (1b)의 수형도 자체는 동일하다고 봄으로써 자유 어순 문제를 통사론의 논

3) 예를 들면, 장석진(1993:138)은 SPR과 COMPS를 구분하는 대신 SUBCAT이라는 통합된 자질로 한국어 동사류의 논항 정보를 표현하고 있다.

- | | |
|------------------------------|--------------------------------|
| (i) SUBCAT <[]> | 자동사: 놀다, 자다
형용사: 길다, 정직하다 |
| (ii) SUBCAT <[], []> | 타동사: 보다, 먹다
지정사: 이다, 아니다 |
| (iii) SUBCAT <[], [], []> | 사동사: 시키다, 먹이다
수여동사: 주다, 보내다 |

의의 대상에서 배제하는 접근법이다.

이 논문에서는 이들 접근법 중에서 (6b)의 내용을 중심으로 살펴보면서, 한국어의 자유어순에 알맞은 새로운 논항가 자질 구조(Valence Feature Structure, 이하 VFS)를 제안할 것이다.

3. LKB에 기반한 한국어의 자유 어순 분석

LKB는 미국 스탠포드 대학의 CSLI 연구소에서 개발한 문법 및 어휘부 개발 환경(a grammar and lexicon development environment)으로서, HPSG와 유형화된 자질 구조(Typed Feature Structure)를 바탕으로 하고 있다.⁴⁾ 이 LKB를 기초로 한국어 구문분석을 시도하는 것은 극히 최근의 일이며, 한국어 어순과 관련된 내용을 일부 포함한 것으로는 양재형(2002), 신효필·고성연(2003a,b) 등 한정된 연구만이 있다.

먼저 이들 연구를 검토한 후, 한국어의 분석에 적합한 새로운 VFS와 규칙을 제안하고 이를 바탕으로 한 LKB 문법 구현의 실체를 보여 주려고 한다.

3.1. 신효필·고성연(2003a,b)의 단층구조분석(Flat Structure Analysis)⁵⁾

신효필·고성연(2003a,b)은 한국어의 문장구조를 계층적인 동사구를 구성하지 않고, 장석진(1993)에 기초하여 단층구조(flat structure)로 분석하여 이를 LKB로 구현하였다.

먼저, 한국어에서 명시어(specifier)와 보충어(complement)의 구분이 모호하므로⁶⁾ COMPS만으로 VFS를 단순화하여 정의하였다. 이를 Sag et al.

4) LKB에 대해서는 <http://www-csli.stanford.edu/~aac/lkb.html> 참조.

5) 윤지원의 조언을 따른다면, 양분구조분석과의 대조를 명확히 하기 위해서는, ‘단층구조분석(Flat Structure Analysis)’이라고 하는 것보다 ‘다분구조분석(N-ary Structure Analysis)’이라고 하는 편이 더 적절할 것이다. 또한 ‘단층구조’라는 용어는 자칫 2개 이상의 층위를 허용하지 않는 듯한 오해를 불러일으킬 수 있다. 그러나 여기서는 이러한 용어상의 문제점만 언급하면서, 앞으로의 논의에서는 편의상 기존의 용어를 그대로 사용하기로 한다.

(2003)과 비교하면 다음과 같다.

(7)

Sag et al. (2003)	신효필·고성연(2003a,b)
$\left[\begin{array}{l} \text{val-cat} \\ \text{SPR} \quad \text{list(expression)} \\ \text{COMPS} \quad \text{list(expression)} \end{array} \right]$	$\left[\begin{array}{l} \text{val-cat} \\ \text{COMPS} \quad \text{list(expression)} \end{array} \right]^7$

기본적인 동사의 유형은 다음의 세 가지로 구분하여 정의한다.⁸⁾ 이는 명사어 자질(SPR)이 빠졌다는 점을 빼면, Sag et al.(2003)과 차이가 없다.

- (8) a. intrans-vlxm := verb-lxm & [COMPS < NP[nom] >].
 b. trans-vlxm := verb-lxm &
 [COMPS < NP[nom], NP[acc] >].
 c. ditrans-vlxm := verb-lxm &
 [COMPS < NP[nom], NP[acc], NP[dat] >].

(8c)의 수여동사를 핵어로 하는 문장에서 단층구조분석이 실제로 구현되는 예를 제시하면 다음과 같다. [보충어-첨가어-보충어-보충어-첨가어-핵어]의 구성으로서 6분지 구조를 보여주고 있다.

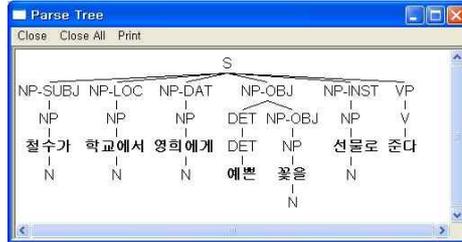
6) 이에 대한 논의는 이 논문에서 다루는 구현 중심의 논의에 대한 이론적 기반으로서 매우 중요한 문제이면서, 또한 쉽게 단정 지을 수 없는 성격의 문제이다. 이 논문에서는 이에 대한 직접적인 언급은 하지 않는 대신, 구현상 더 나은 결과를 보이는 접근법을 제시함으로써 역으로 새로운 이론적 접근의 가능성을 암시하고자 한다.

7) 장석진(1993)을 따른다면, COMPS라는 자질보다는 SUBCAT(subcategorization)이라는 자질을 사용하는 것이 더 적절할 것이다. 실제로 양재형(2002)에서는, 아래에 제시한 것처럼, COMPS에 해당하는 자질로서 SCAT을 사용하였다. 3.2. 참조.

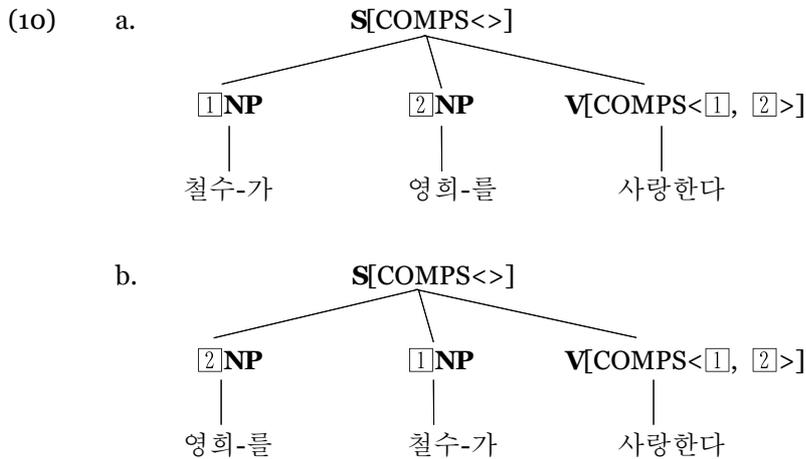
$$\left[\begin{array}{l} \text{val-cat} \\ \text{SCAT} \quad \text{list(expression)} \end{array} \right]$$

8) (8)은 LKB의 types.tdl 파일에 정의되어 있는 유형화된 자질 구조로서 Copestake(2002)의 기술 언어(Description Language)를 따른 것이므로, Sag et al. (2003)의 표기 방식과 다소 차이가 있다. LKB에서 구현한 실제의 유형(type)과 규칙(rule)들의 표기는 앞으로 도 LKB의 표기 방식을 그대로 따른다.

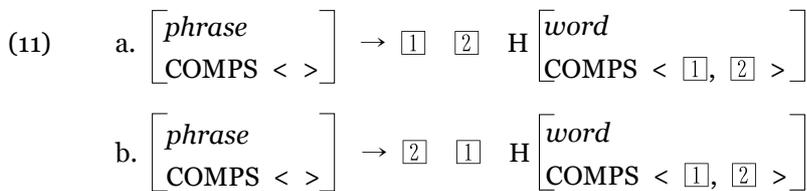
(9)



이러한 다분지(*N-ary*) 구조를 보여주는 가장 단순한 문형을 통해 실제 구현이 어떻게 이루어지는지 살펴보기로 하자. 아래 (10)은 전형적인 타동사 구문의 예로서 a는 [주어-목적어-동사]의 순서, b는 [목적어-주어-동사]의 순서로 이루어진 문장이다. 수형도에 COMPS가 실현되는 양상도 함께 확인할 수 있도록 연동지표(*coindex*)를 상자번호로 표시하였다.



보충어들의 순서가 자유롭다면, 보충어의 결합 순서에 따라 별개의 규칙이 필요하다. 위 (10)의 a, b를 위해서 다음의 두 규칙을 가정한다.



(11)의 두 규칙은 타동사와 같은 2항 동사(두 자리 서술어) 문형의 핵어와 보충어들의 결합을 정의하는 규칙들이다. 2항 동사 이외의 동사로 이루어진 문장들에 대해서도 핵어와 보충어의 결합 규칙을 정의해 주어야 한다. 이들 1항 동사와 3항 동사의 보충어 결합을 위한 규칙도 보충어의 결합 위치에 따라 여러 개의 규칙 정의가 필요하다. 이 때 산술적으로 필요한 규칙의 개수는, 첨가어(adjunct)의 결합을 배제한 순수 보충어(complement)만 고려하면, 보충어의 개수를 n 이라 할 때 $n!$ 개로 계산한다.⁹⁾

(12)

동사 유형	보충어의 개수 (n)	첨가어 0개일 때	첨가어 1개일 때	첨가어 2개일 때
1항 동사	1	$1!=1$	2	3
2항 동사	2	$2!=2\times 1=2$	6	12
3항 동사	3	$3!=3\times 2\times 1=6$	24	60
	소계	9	32	75
	총계	116		

아래 (13)~(15)는 신호필·고성연(2003b)에서 LKB의 rules.tdl 파일에는 문장 규칙으로 정의한 116개의 규칙들 중의 일부로서, 1항 동사부터 3항 동사까지 보충어만을 고려하여 정의한 것들의 예이다. (12)에서 확인한 바와 같이 규칙의 개수는 총 9개이다.

(13) 자동사문 규칙: 1개
 intrans_s_rule := binary-rule &
 [ARGS < #1, phrase & [HEAD verb, COMPS < #1 >] >].

(14) 타동사문 규칙: 2개
 a. trans_s_rule_1 := ternary-rule &
 [ARGS < #1, #2, phrase & [HEAD verb, COMPS < #1, #2 >] >].
 b. trans_s_rule_2 := ternary-rule &

9) 첨가어가 결합하는 경우의 수의 계산은 이보다 약간 더 복잡하며, 결합이 가능한 첨가어의 개수와 첨가어와 보충어의 위치까지 고려하게 되면 규칙의 개수가 기하급수적으로 늘어나게 된다. 신호필·고성연(2003b)에서는 모두 116개의 문장 규칙을 정의하였는데, 이는 보충어와 첨가어의 결합을 각각 최대 3개와 2개로 한정하여 계산한 결과이다.

[ARGS < #2, #1, phrase & [HEAD verb, COMPS < #1, #2 >] >].

(15) 수여동사문 규칙: 6개

a. ditrans_s_rule_1 := 4nary-rule &

[ARGS < #1, #2, #3, phrase &
[HEAD verb, COMPS < #1, #2, #3 >] >].

b. ditrans_s_rule_2 := 4nary-rule &

[ARGS < #1, #3, #2, phrase &
[HEAD verb, COMPS < #1, #2, #3 >] >].

c. ditrans_s_rule_3 := 4nary-rule &

[ARGS < #2, #1, #3, phrase &
[HEAD verb, COMPS < #1, #2, #3 >] >].

d. ditrans_s_rule_4 := 4nary-rule &

[ARGS < #2, #3, #1, phrase &
[HEAD verb, COMPS < #1, #2, #3 >] >].

e. ditrans_s_rule_5 := 4nary-rule &

[ARGS < #3, #1, #2, phrase &
[HEAD verb, COMPS < #1, #2, #3 >] >].

f. ditrans_s_rule_6 := 4nary-rule &

[ARGS < #3, #2, #1, phrase &
[HEAD verb, COMPS < #1, #2, #3 >] >].

동일한 핵어와 보충어의 결합을 위해 여러 개의 규칙을 설정해야 한다는 것은 문법에 불필요한 부담을 가중시키는 것이 아닐 수 없다.¹⁰⁾ 예를 들어, *철수-가 영화-에게 꽃-을 준다*라는 문장과 동일한 문장성분들(‘철수-

10) 신효필,고성연(2003b)은 이러한 문제점을 스스로 지적하고 있다.

- “한국어의 문법에서 동사, 논항, 첨가어를 포함한 다중 분지관계의 가능성은 사실 그 수가 열려 있다고 할 수 있다. 따라서 이를 LKB에서 규칙화하기에는 비효율적이며 불가능하다.”
- “LKB의 구조적인 제약 때문에 단층구조에 의한 분석은 논항과 첨가어들의 순서에 민감하다. (중략) 논항과 첨가어들 간의 순서 변화에 따라 세분된 규칙을 기술할 수밖에 없는데 이 점은 LKB에서 단층구조분석을 시도하는 방법의 한계이다.”

그러나, 이러한 한계는 단층구조분석 자체의 한계라기보다는 LKB에서의 분석의 한계로 보아야 할 듯하다. 즉, 단층구조분석은 그 자체가 양분구조분석을 전제로 하는 LKB 시스템과 맞지 않는다는 것이다. 따라서 단층구조분석을 뒷받침할 만한 다른 시스템, 예컨대 MEAT(The Multilingual Environment for Advanced Translations)와 같은 자질연산 시스템에서는 적은 수의 문법 규칙으로 효과적인 단층구조 기술이 가능할 것이다. <http://crl.nmsu.edu/~ahmed/Meat/intro.html> 참조.

가’, ‘영화-에게’, ‘꽃-을’, ‘주다’)로 이루어진 6개의 문장에 대해 6개의 규칙이 필요하다고 보는 것은, 규칙에 부과하는 문법의 부담을 줄이고 어휘에 보다 많은 정보를 의존하려는 HPSG의 기본 경향에 어긋나는 것이다.

더구나 (1a) *철수가 영화를 사랑한다*와 (1b) *영화를 철수가 사랑한다*와 같은 두 문장이 완전히 동일한 핵어와 보충어로 이루어졌음에도 불구하고 서로 다른 규칙에 의해 설명되어야 한다면, 그 규칙에서 어떠한 일반성도 발견하기 어려울 것이다.

3.2. 양재형(2002)의 양분단선구조분석(Binary Linear Structure Analysis)

규칙의 수를 다소 줄일 수 있는 분석 방법이 없지는 않다. 양재형(2002)은 다음과 같은 4개의 규칙들로 3항 동사 문형까지 설명하고 있으므로 단층구조분석에 비해 경제적인 분석이라고 할 수 있다.¹¹⁾

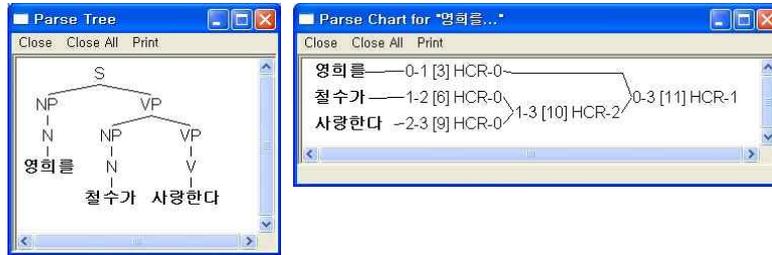
- (16) a. head-complement-rule-1 := binary-rule &
 [SCAT #2,
 ARGS < #1, phrase & [SCAT [FIRST #1, REST #2]] >].
- b. head-complement-rule-2 := binary-rule &
 [SCAT < #1 >,
 ARGS < #2, phrase & [SCAT < #1, #2 >] >].
- c. head-complement-rule-3 := binary-rule &
 [SCAT < #1, #3 >,
 ARGS < #2, phrase & [SCAT < #1, #2, #3 >] >].
- d. head-complement-rule-4 := binary-rule &
 [SCAT < #1, #2 >,
 ARGS < #3, phrase & [SCAT < #1, #2, #3 >] >].

(16a)는 핵어의 SCAT 중 첫 번째 요소와 핵어의 결합을 위한 규칙(HCR-1¹²⁾)으로서, *철수가 영화를 사랑한다*와 같은 문장에서 목적어 NP(‘영화-

11) 여기 제시된 양재형(2002)의 핵어-보충어 결합 규칙은 ARGS에 두 개의 요소의 결합만을 허용하는 양분구조의 규칙들이며, SCAT value에 보충어들을 단선적으로 배열하고 있으므로, 양분단선구조분석(Binary Linear Structure Analysis)이라 이름붙일 수 있을 것이다.

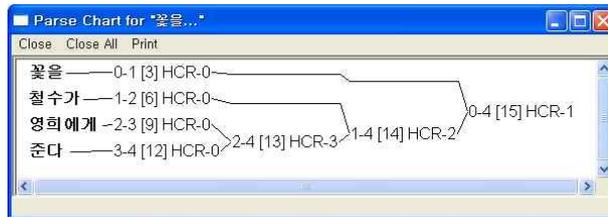
요소와 핵어의 결합을 위한 규칙들이다(HCR-2, HCR-3, HCR-4). 먼저 규칙 b(HCR-2)는 어휘 항목 ‘사랑하다’의 SCAT을 < 목적어, 주어 >로 하였을 때, ‘철수-가’와 같은 주어가 동사와 먼저 결합하여 ‘철수-가 사랑한다’와 같은 VP가 만들어지도록 한다. 이 VP가 규칙 a에 의해 다시 ‘영희-를’과 결합함으로써, *영희-를 철수-가 사랑한다*와 같은 [목적어-주어-동사] 순의 문장이 적형문으로 인가되게 된다. 아래 (19)의 구문분석차트에서 ‘철수-가 사랑한다’가 HCR-2에 의한 결합임을 확인할 수 있다.

(19) *영희-를 철수-가 사랑한다*의 Parse Tree와 Parse Chart



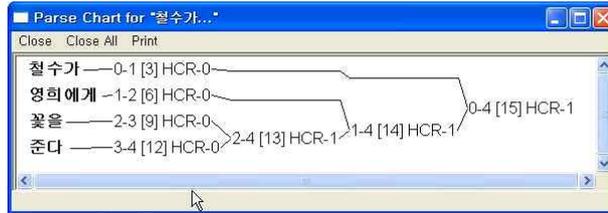
이외에 규칙 c(HCR-3)와 규칙 d(HCR-4)는 3항 동사의 둘째 혹은 셋째 논항의 우선 결합을 인가하고, 충족되지 않은 두 개의 논항에 대해서는 이후 b-a, 또는 a-a 규칙 적용을 통해 적형문을 만들어 내도록 한다.

(20) *꽃-을 철수-가 영희-에게 준다*의 Parse Chart

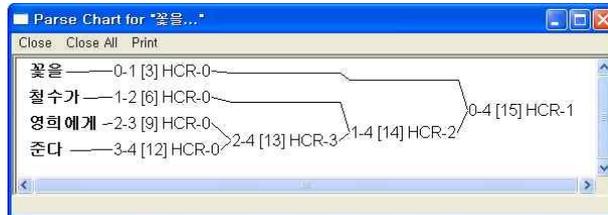


확실히, 이러한 분석은 3.1에서 살펴본 단층구조분석에 비해 탁월한 경제성을 보여준다고 할 수 있다. 그러나 이러한 비교우위는 양분분석과 LKB 시스템의 태생적 궁합에 다분히 의존하는 것으로 보인다. 오히려 양분단선 구조분석의 핵어-보충어 결합규칙에서 우리는 몇 가지 문제점을 발견할 수가 있다. 먼저, 다음의 세 문장에 대한 구문분석차트를 보자.

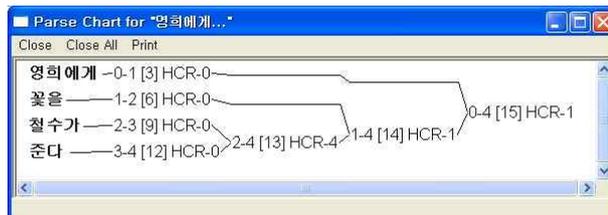
(21) a. 철수가 영화-에게 꽃-을 준다의 구문분석차트



b. 꽃-을 철수가 영화-에게 준다의 구문분석차트



c. 영화-에게 꽃-을 철수가 준다의 구문분석차트



세 문장의 구문분석차트를 잘 살펴보면, ‘철수가’의 결합에 관여하는 규칙이 (21a)에서는 HCR-1, (21b)에서는 HCR-2, (21c)에서는 HCR-4로 문장마다 다른 반면에, ‘꽃-을’의 결합에 관여하는 규칙은 한결같이 HCR-1임을 알 수 있다. 주어는 문장 안에서의 위치에 따라 다른 규칙을 적용하고, 목적어는 위치와 상관없이 동일한 규칙 HCR-1만을 적용하는 것이다. 이러한 자의적인 규칙 적용은 우리의 언어적 직관에 부합하지 못한다.

또한, (21b)의 HCR-3은 수여동사 문장에서 동사 바로 앞에 나타나는 경우의 NP[dat]와 동사 핵어의 결합에만 관여하는 규칙이며, 다른 위치의 NP[dat]이나 다른 격의 NP와 동사의 결합에 대해서는 아무런 역할도 하지 않는다. 이와 마찬가지로, (21c)의 HCR-4는 동사 바로 앞에 나타나는 경우의 NP[nom] 결합만을 위해 존재하는 규칙이다. 반면에, HCR-1은 NP

의 격에 상관없이 매우 광범위한 적용 범위를 갖는 규칙이다. 어떠한 문법에서 규칙의 개수가 충분히 적다는 점은 바람직한 것이라 하겠으나, 얼마 되지 않는 규칙들간의 기능부담량이 이처럼 불균형적인 것은 좋지 않다. 다시 말하면, HCR-3과 HCR-4와 같은 규칙은 너무 ‘비싼’ 규칙이다.

자연언어에서 논항을 네 개 이상 요구하는 경우가 있다면 ‘더 비싼’ 규칙들이 추가적으로 정의되어야 한다.

(22) SCAT에 네 개의 요소가 올 수 있다고 가정할 경우 필요한 규칙들

- a. head-complement-rule-5 := binary-rule & [SCAT < #1, #3, #4 >, ARGS < #2, phrase & [SCAT < #1, #2, #3, #4 >] >].
- b. head-complement-rule-6 := binary-rule & [SCAT < #1, #2, #4 >, ARGS < #3, phrase & [SCAT < #1, #2, #3, #4 >] >].
- c. head-complement-rule-7 := binary-rule & [SCAT < #1, #2, #3 >, ARGS < #4, phrase & [SCAT < #1, #2, #3, #4 >] >].

또 한 가지 언급할 문제점으로 Sag et al. (2003)의 Head-Complement Rule과 (16a)의 HCR-1과의 구조적인 차이를 들 수 있다. 아래 (23)과 (24)를 비교해 보자.

$$(23) \quad \left[\begin{array}{l} \textit{phrase} \\ \text{COMPS} < \boxed{2} > \end{array} \right] \rightarrow \boxed{1} \quad \text{H} \left[\begin{array}{l} \textit{word} \\ \text{COMPS} < \boxed{1}, \boxed{2} > \end{array} \right]$$

$$(24) \quad \text{head-complement-rule-1} := \text{binary-rule} \ \& \ [\text{SCAT} \ \#2, \text{ARGS} < \#1, \text{phrase} \ \& \ [\text{SCAT} \ [\text{FIRST} \ \#1, \text{REST} \ \#2]] >] .$$

(23)은 (16a)를 최대한 Sag et al. (2003)의 규칙 형식에 가깝게 표현한 것이고,¹⁴⁾ (24)는 (16a)를 다시 편의상 옮겨 적은 것이다. 문제는 이 둘이 서

14) COMPS와 SCAT은 같은 자질의 다른 표현으로 간주한다.

로 다른 규칙이라는 점이다. 더 정확히 말하면, Sag et al. (2003)은 아무 리 해도 (24)의 규칙이 표현하는 바를 표현할 수가 없다. 앞서 규칙들의 기능부담량이라는 표현을 쓰면서 관련 내용을 언급한 바가 있지만, (24)의 규칙은 SCAT 리스트에 논항이 하나인 경우(자동사)와 둘인 경우(타동사), 그리고 셋인 경우(수여동사) 모두에 있어서 첫 번째 논항이 동사 앞에 나올 때에 이 둘을 결합시키도록 하는 규칙이다. LKB에서는 이러한 규칙의 정의가 가능하겠지만, 불행하게도 Sag et al. (2003)의 HPSG에서는 그렇지 않은 것 같다. (16)을 Sag et al. (2003)과 호환되는 규칙으로 재정의하면 다음과 같이 규칙의 개수가 5개로 증가한다.

- (25)
- a. head-complement-rule-1 := binary-rule &
 - [SCAT < #2 >,
 - ARGS < #1, phrase & [SCAT < #1, #2 >] >].
 - b. head-complement-rule-2 := binary-rule &
 - [SCAT < #1 >,
 - ARGS < #2, phrase & [SCAT < #1, #2 >] >].
 - c. head-complement-rule-3 := binary-rule &
 - [SCAT < #2, #3 >,
 - ARGS < #1, phrase & [SCAT < #1, #2, #3 >] >].
 - d. head-complement-rule-4 := binary-rule &
 - [SCAT < #1, #3 >,
 - ARGS < #2, phrase & [SCAT < #1, #2, #3 >] >].
 - e. head-complement-rule-5 := binary-rule &
 - [SCAT < #1, #2 >,
 - ARGS < #3, phrase & [SCAT < #1, #2, #3 >] >].

여기서 지적하고 싶은 것은, ‘결국 규칙 1개가 더 증가한다는 점’이 아니다. 단층구조분석에 비해 규칙의 수가 대폭 줄어든 것은 사실이지만, 위 규칙들의 ARGS 구조를 들여다보면 논항들의 위치와 결합 순서의 경우의 수만큼 규칙이 필요하다는 근본적인 특징은 단층구조분석과 별반 다르지 않다는 점이다. 이것은 단층구조분석(신호필·고성연 2003a,b)과 양분단선 구조분석(양재형 2002)이 공통적으로 가정하고 있는 단선구조(Linear Structure)의 보충어 리스트가 한국어의 자유 어순을 설명하는 데에는 적합하지 않음을 보여준다.

계다가 (23)의 COMPS에서 보듯이 실현되지 않은 리스트값을 상위 노

드의 COMPS 리스트값으로 승계하도록 하는 것이 과연 Sag et al. (2003)의 HPSG의 Head-Complement Rule의 변종으로서 가능한 것인지 의문이다. 이것은 Sag et al. (2003)에서 GAP¹⁵⁾을 도입한 것과 방법상 유사한 듯한데, *철수가 영화를 사랑한다*와 같은 가장 무표적인 어순의 문장에서 실현되지 않은 논항 ‘철수-가’를 중간 노드의 VP(‘영화를 사랑한다’)가 자신의 COMPS 리스트값으로 갖게 되는 것을 볼 때¹⁶⁾, 내용상으로는 GAP의 특수성(topicalized sentence, *wh*-question, relative clauses 등)과는 무관해 보인다.¹⁷⁾

3.3. 양분병렬구조분석(Binary Parallel Structure Analysis)

Sag et al. (2003), 양재형(2002), 신호필·고성연(2003a,b) 등이 모두 COMPS value의 *list(expression)*에 논항들을 단선적 순서(linear order)에 따라 배열하는 단선구조분석(Linear Structure Analysis)의 입장을 취한 것¹⁸⁾과 달리, 이 논문에서는 각 논항을 Valence Feature의 하위 자질(sub-feature)로서 병렬적으로 배치하는 병렬구조분석(Parallel Structure Analysis)의 입장을 취한다. 다시 말하면, nominative, accusative, dative 등의 격을 갖는 논항들을 하나의 COMPS list에 직렬로 늘어놓지 않고, VAL의 하위자질 SUBJ, OBJ, S-OBJ를 병렬 구조로 두고, 각각의 value로 *list(expression)*을 가정한다는 것이다. 그리고, 신호필·고성연(2003a,b)의 단층구조분석(Flat Structure Analysis)과 달리, 양재형(2002)과 같은 양분구조분석(Binary Structure Analysis)을 채택¹⁹⁾하여 과도한 규칙의 수를 줄

15) Sag et al. (2003)에서 COMPS의 리스트값 중 실현되지 않은 일부를 새로운 자질 GAP의 리스트값으로 할당하고 GAP Principle을 도입하여 비국부적 의존관계(nonlocal dependency)를 해소할 수가 있었다.

16) *영화를 철수가 사랑한다*의 ‘영화-를’이 주제화되었다고 인정한다손 치더라도 *철수가 영화를 사랑한다*의 ‘철수-가’가 주제화되었다고 말하기엔 무리일 것이다.

17) 오히려 신호필, 고성연(2003)에서는 실현되지 않은 COMPS 값을 이와 같이 상위 노드의 COMPS 값에 남겨 두는 방식으로 관계절을 포함한 구문분석을 시도하여 일정한 성과를 얻었다.

18) 엄밀히 말하면, Sag et al. (2003)의 VFS는 양분구조분석(binary structure analysis)와 단층구조분석(flat structure analysis) 간의 절충이라고 할 수 있을 것이다. 즉, SPR와 Head의 결합은 양분구조로, Head와 COMPS의 결합은 단층구조로 분석하는 것이다.

이고자 한다. 이러한 접근법을 양분병렬구조분석(Binary Parallel Structure Analysis)이라고 이름붙일 수 있을 것이다.²⁰⁾

3.3.1. 양분병렬구조분석의 논항자 자질 구조와 규칙

먼저 양분병렬구조분석의 Valence의 자질 구조를 다음과 같이 제안한다.

$$(26) \quad \left[\begin{array}{l} \text{val-cat} \\ \text{SUBJ } \textit{list(expression)} \\ \text{OBJ } \textit{list(expression)} \\ \text{S-OBJ}^{21)} \textit{list(expression)} \end{array} \right]$$

이때 list value는 ‘없거나, 있으면 하나의 NP’만을 허용하는 것으로 보며, 이러한 제약은 아래의 (27)에 동사 어휘소(verb lexeme)의 자질 구조에 명시된다.

$$(27) \quad \left[\begin{array}{l} \text{verb-lxm} \\ \text{HEAD } \textit{verb} \\ \text{VAL } \left[\begin{array}{l} \text{val-cat} \\ \text{SUBJ } < \text{NP[nom]} > \\ \text{OBJ } < (\text{NP[acc]}) > \\ \text{S-OBJ } < (\text{NP[dat]}) > \end{array} \right] \end{array} \right]$$

19) 이 논문이 양분구조분석의 입장에 서는 것은 리스트 구조에 바탕을 둔 머리-꼬리(head-tail)의 양분구조만을 허용하는 LKB의 Typed Feature Structure의 구조적인 제약 때문이기도 하다. Copestake(2002) 참조.

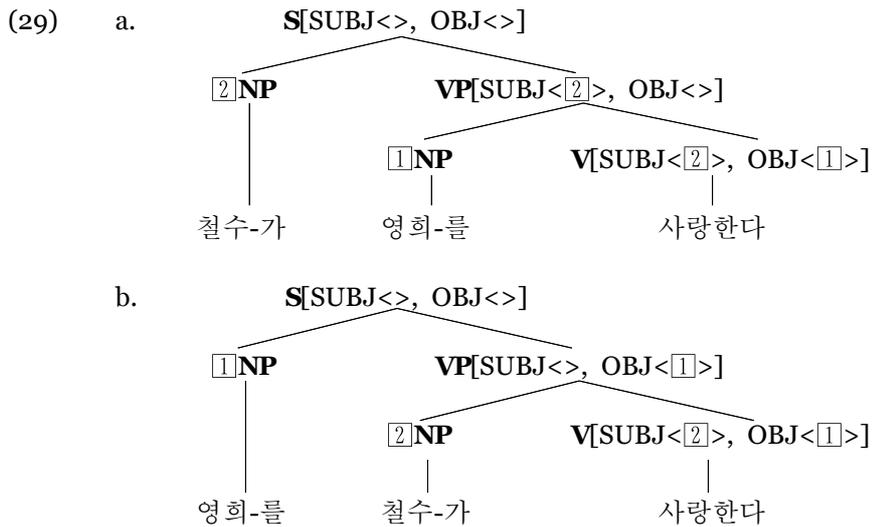
20) ‘단선(linear)’과 ‘병렬(parallel)’을 이와 같이 구분한다면, 신호필·고성연(2003a,b)의 분석은 단층구조분석이라기보다는 단층단선구조분석(Flat Linear Structure Analysis)라고 불릴 만하다. 양재형(2002)의 분석을 양분단선, 신호필·고성연(2003a,b)의 분석을 단층단선, 이 논문에서 제안하는 분석을 양분병렬이라고 할 때, 우리는 또 하나의 분석 방법의 논리적 가능성, 즉 단층병렬구조분석(Flat Parallel Structure Analysis)를 떠올릴 수 있을 것이다. 그러나, ‘병렬(parallel)’ 구조 자체는 하나의 list value로 하나의 요소만을 가정하기 때문에 ‘단층병렬구조’는 성립하지 않는다.

21) 처음에는 SUBJ, OBJ, DAT와 같은 자질로 표시하였는데, DAT는 문법 관계(grammatical relation)를 나타내는 명칭으로 부적합하므로 S-OBJ(secondary object)로 수정하였다. 이 부분에 대한 유은정 선생님의 지적과 조언에 감사드린다.

위 (27)을 바탕으로 하여, 동사의 유형은 동사가 요구하는 논항의 개수와 제약에 따라 다음의 세 종류로 구분된다.

- (28) a. 자동사: *가다, 자다*
 intrans-vlxm := verb-lxm &
 [HEAD verb, VAL val-cat & [SUBJ < NP[nom] >, OBJ < >, S-OBJ < >]].
- b. 타동사: *사랑하다, 때리다*
 trans-vlxm := verb-lxm &
 [HEAD verb, VAL val-cat & [SUBJ < NP[nom] >, OBJ < NP[acc] >, S-OBJ < >]].
- c. 수여동사: *주다, 보내다*
 ditrans-vlxm := verb-lxm &
 [HEAD verb, VAL val-cat & [SUBJ < NP[nom] >, OBJ < NP[acc] >, S-OBJ < NP[dat] >]].

이러한 접근법에서 가정하는 문장의 기본 구조를 (5), (10), (17)과 비교가 가능하도록 (1)의 두 문장에 대한 예로써 제시하면 다음과 같다.



(28)의 동사 유형에 따른 논항들의 실현을 위해서는 주어-핵어 결합 규칙 (Subject Valence Rule), 목적어-핵어 결합 규칙(Object Valence Rule), 이차목적어-핵어 결합 규칙(S-Object Valence Rule)이 각각 다음과 같이 정의 되어야 한다.

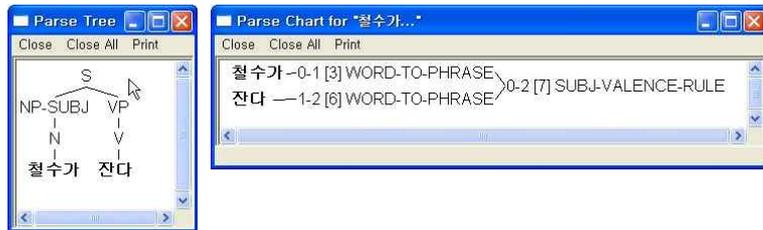
- (30) a. subj-valence-rule := binary-phrase &
 [HEAD verb,
 VAL.SUBJ < >,
 VAL.OBJ #2,
 VAL.S-OBJ #3,
 ARGS < #1, phrase &
 [HEAD verb,
 VAL.SUBJ < #1 >,
 VAL.OBJ #2,
 VAL.S-OBJ #3] >].
- b. obj-valence-rule := binary-phrase &
 [HEAD verb,
 VAL.SUBJ #1,
 VAL.OBJ < >,
 VAL.S-OBJ #3,
 ARGS < #2, phrase &
 [HEAD verb,
 VAL.SUBJ #1,
 VAL.OBJ < #2 >,
 VAL.S-OBJ #3] >].
- c. dat-valence-rule := binary-phrase &
 [HEAD verb,
 VAL.SUBJ #1,
 VAL.OBJ #2,
 VAL.S-OBJ < >,
 ARGS < #3, phrase &
 [HEAD verb,
 VAL.SUBJ #1,
 VAL.OBJ #2,
 VAL.S-OBJ < #3 >] >].

ARGS에서 실현되는 논항을 제외하면, head daughter의 나머지 VAL 자질은 mother의 VAL 자질과 동일해야 함을 알 수가 있다. 이것은 이 논문에서 제안하는 바와 같이 VAL의 자질 구조와 규칙을 바꾸어도, Sag et al. (2003)의 Valence Principle(각주 22 참조)은 여전히 만족된다는 것을 보여준다.

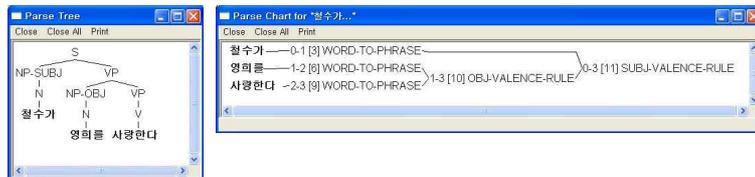
3.3.2. LKB에 기반한 양분병렬구조분석의 구현

이제 (30)의 3개 규칙만으로 (13)~(15)의 9개 규칙이 담당했던 핵어와 보충어 간의 결합 기능을 온전히 수행할 수가 있다. 이와 관련하여 LKB에서 다음 9개 문장을 어떻게 구문분석하는지 그 실체를 살펴보도록 하자. (31)는 자동사 문형을 새로 제시한 것이고, (32), (33)은 (1), (2)를 편의상 다시 쓴 것이다.

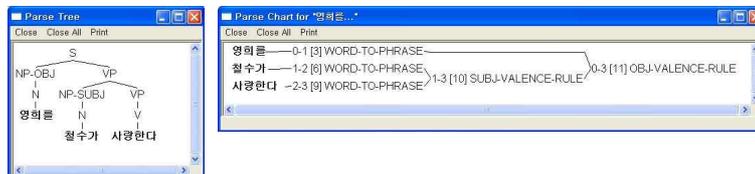
(31) 철수-가 잔다.



(32) a. 철수-가 영화-를 사랑한다.

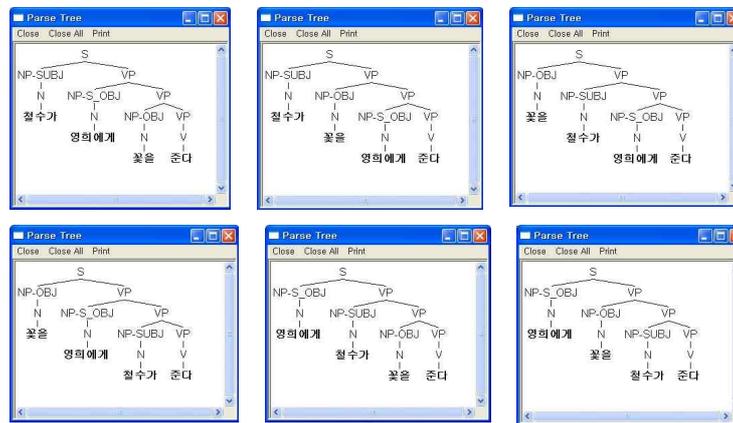


b. 영화-를 철수-가 사랑한다.

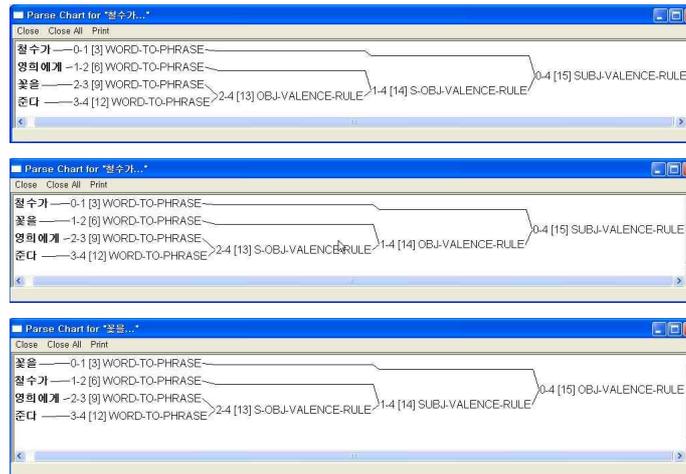


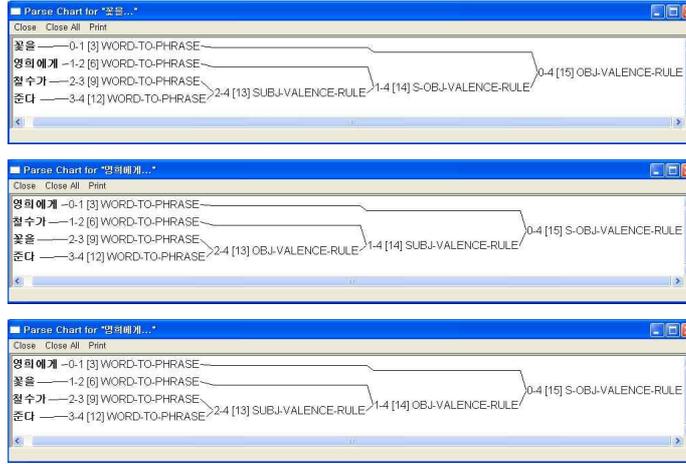
- (33) a. 철수-가 영희-에게 꽃-을 준다.
 b. 철수-가 꽃-을 영희-에게 준다.
 c. 꽃-을 철수-가 영희-에게 준다.
 d. 꽃-을 영희-에게 철수-가 준다.
 e. 영희-에게 철수-가 꽃-을 준다.
 f. 영희-에게 꽃-을 철수-가 준다.

<Parse Trees>



<Parse Charts>





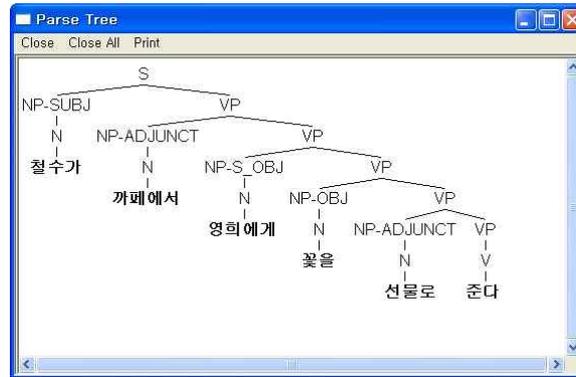
(31)~(33)의 9개 문장에 대해 신효필, 고성연(2003b)의 단층단선구조분석에서는 9개의 규칙으로 설명하였고, 양재형(2002)의 양분단선구조분석에서는 문장성분별로 일관성이 없는 4개(또는 5개)의 규칙으로 설명하였지만, 이 논문에서 제안한 양분병렬구조분석은 (30) a~c의 3개의 규칙만을 이용하여 이들 규칙의 다양한 조합을 만들어냄으로써 다양한 문형과 자유어순에 대해 일반적이고 경제적인 분석을 하는 데에 성공하고 있다.

여기에 첨가어의 결합까지를 고려하면, 양분병렬구조분석의 일반성과 경제성이 더욱 잘 드러날 수 있다. (12)에서 설명한 것처럼, LKB를 이용한 단층구조분석이 3항 동사, 2개 첨가어까지 고려할 때 116개의 규칙이 필요한 것과 달리, 양분병렬구조분석은 (30)의 규칙들과 유사한 첨가어-핵어 결합 규칙(Adjunct Rule) 1개만을 더 필요로 할 뿐이다.

- (34) a. 첨가어 결합 규칙
 adjunct-rule := binary-phrase &
 [HEAD verb,
 VAL.SUBJ #1,
 VAL.OBJ #2,
 VAL.S-OBJ #3,
 ARGS < NP[adjunct], phrase &
 [HEAD verb,
 VAL.SUBJ #1,
 VAL.OBJ #2,

b. 첨가어(adjunct)의 결합 예시:

철수가 *카페-에서* 영희-에게 꽃-을 선물-로 준다.



4. 양분병렬구조분석과 HPSG

이 절에서는 3.3에서 제안한 문법이 기존 HPSG(Sag et al. 2003) 이론에 대해 갖는 함의를 살펴보고, 보편 문법의 관점에서 3.3의 제안에 대한 일반화를 시도하려 한다.

3.3에서 제안한 Subject Valence Rule, Object Valence Rule, S-Object Valence Rule의 세 가지 규칙들은 한국어에서 동사 핵어와 그 논항들의 결합을 설명하는 데에 적합한 경제적인 방안임에 틀림없다.²²⁾ 그러나 이들 세 규칙이 대체한 HPSG의 Head-Specifier Rule과 Head-Complement Rule은 동사 핵어에만 국한되는 규칙이 아니라, NP, VP, AP, ADVP 등의 모든 구(phrase), 이른바 XP에 두루 통용되는 일반 규칙이었다. (3), (4)의

22) 또한 이 규칙들은 다음과 같은 기본 원리를 모두 충족시키는 규칙들이다. Sag et al. (2003) 참조.

- Head Feature Principle (HFP)
In any headed phrase, the HEAD value of the mother and the HEAD value of the head daughter must be identical.
- Valence Principle
Unless the rule says otherwise, the mother's values for the VAL features are identical to those of the head daughter.

HSR과 HCR을 앞으로의 논의 전개를 위해 여기에 다시 옮겨 제시한다.

(35) Head-Specifier Rule

$$\begin{bmatrix} \textit{phrase} \\ \text{SPR} < > \end{bmatrix} \rightarrow \boxed{\text{I}} \text{ H} \begin{bmatrix} \text{SPR} < \boxed{\text{I}} > \\ \text{COMPS} < > \end{bmatrix}$$

A phrase can consist of a (lexical or phrasal) head preceded by its specifier.

(36) Head-Complement Rule

$$\begin{bmatrix} \textit{phrase} \\ \text{COMPS} < > \end{bmatrix} \rightarrow \text{H} \begin{bmatrix} \textit{word} \\ \text{COMPS} < \boxed{\text{I}}, \dots, \boxed{\text{N}} > \end{bmatrix} \boxed{\text{I}} \dots \boxed{\text{N}}$$

A phrase can consist of a lexical head followed by all its complements.

SUBJ, OBJ, S-OBJ와 같은 자질을 병렬적으로 구분하여 얻을 수 있었던 규칙 기술의 경제성과 SPR, COMPS와 같은 자질로써 얻을 수 있었던 모든 구 구조에 대한 설명의 일반성을 동시에 얻을 수는 없을까? 다음과 같이 자질의 이름을 바꾸면 두 접근법의 장점을 함께 살릴 수 있을 것이다.

(37)

$$\begin{bmatrix} \textit{val-cat} \\ \text{SPR} \quad \textit{list(expression)} \\ \text{COMP}_1 \quad \textit{list(expression)} \\ \text{COMP}_2 \quad \textit{list(expression)} \end{bmatrix}$$

Sag et al. (2003)과 3.3(26)의 VFS로부터 이 둘을 절충한 새로운 자질 구조를 보이면 다음과 같다.

(38)

Sag et al. (2003)	3.3 양분병렬구조분석의 제안
$\left[\begin{array}{l} \textit{val-cat} \\ \text{SPR} \quad \textit{list(expression)} \\ \text{COMPS} \quad \textit{list(expression)} \end{array} \right]$	$\left[\begin{array}{l} \textit{val-cat} \\ \text{SUBJ} \quad \textit{list(expression)} \\ \text{OBJ} \quad \textit{list(expression)} \\ \text{S-OBJ} \quad \textit{list(expression)} \end{array} \right]$



3.3 제안의 수정
$\left[\begin{array}{l} \textit{val-cat} \\ \text{SPR} \quad \textit{list(expression)} \\ \text{COMP}_1 \quad \textit{list(expression)} \\ \text{COMP}_2 \quad \textit{list(expression)} \end{array} \right]^{23)}$

OBJ와 S-OBJ를 각각 COMP₁, COMP₂와 같은 일반적 자질로 바꿈으로써, 기본 타동사문과 기본 수여동사문 이외의 다양한 모든 구문(지정사문, 존재사문, 사동사문 등)을 표현하는 것이 가능해진다.

(38)에서 제안된 자질 구조를 바탕으로, Sag et al. (2003)의 HSR과 HCR을 다음 (39)와 (40)과 같이 수정할 수 있다.

(39) Head-Specifier Rule (revised)

23) 보다 일반적으로는 다음과 같이 보충어의 개수를 제한하지 않을 수도 있을 것이다.

$$\left[\begin{array}{l} \textit{val-cat} \\ \text{SPR} \quad \textit{list(expression)} \\ \text{COMP}_1 \quad \textit{list(expression)} \\ \text{COMP}_2 \quad \textit{list(expression)} \\ \vdots \\ \text{COMP}_n \quad \textit{list(expression)} \end{array} \right]$$

$$\begin{bmatrix} \text{phrase} \\ \text{SPR} < > \end{bmatrix} \rightarrow \boxed{1} \quad \text{H} \begin{bmatrix} \text{SPR} < \boxed{1} > \end{bmatrix}$$

(40) Head-Complement Rule (revised)

$$\begin{bmatrix} \text{phrase} \\ \text{COMP}_n < > \end{bmatrix} \rightarrow \text{H} \begin{bmatrix} \text{COMP}_n < \boxed{1} > \end{bmatrix} \quad \boxed{1}$$

위 (39)의 수정된 HSR을 보면, COMPS(수정 제안된 구조에서는 COMP_1 과 COMP_2 에 해당함)의 value에 대한 언급이 없다는 점이 기존 Sag et al. (2003)의 HSR인 (35)와 다르다. 한국어에서는 명시어(주어)가 보충어보다 먼저 핵어와 결합할 수도 있기 때문이다.

(40)은 Sag et al. (2003)의 HCR과 비교하여, 핵어가 word여야 한다는 제약이 없다. 또한 COMPS 대신 COMP_n 으로 표시하여, COMP_1 결합 규칙, COMP_2 결합 규칙, COMP_3 결합 규칙 등을 하나의 규칙 형식으로 나타내었다. 일종의 표기규약을 이용한 것이다.

규칙 (39)를 언어 보편적 HSR로 가정하여, 한국어에서는 이 규칙을 그대로 적용하고, 영어의 HSR은 이 규칙에서 우변의 핵의 COMP value가 모두 empty여야 한다는 제약을 추가하여 적용하면 될 것이다.

(41) English Head-Specifier Rule

$$\begin{bmatrix} \text{phrase} \\ \text{SPR} < > \end{bmatrix} \rightarrow \boxed{1} \quad \text{H} \begin{bmatrix} \text{SPR} < \boxed{1} > \\ \text{COMP}_1 < > \\ \text{COMP}_2 < > \\ \vdots \\ \text{COMP}_n < > \end{bmatrix}$$

규칙 (40)도 언어 보편적 HCR로 가정하여, 한국어에서는 이 규칙을 그대로 적용하고²⁴⁾, 영어의 HCR은 이 규칙에 우변의 핵의 SPR value가 반드시 non-empty여야 한다는 제약을 더한다. 영어의 적형문에서는 핵어가 명

24) (39)와 (40)에서 SPR과 COMP가 그 이름만 다를 뿐 실제 행동은 동일하다는 점에서, 과연 한국어에서 주어를 보충어가 아닌 명시어로 구분할 필요가 있는지 의문이다.

시어와 결합하기 이전에 먼저 모든 보충어들과 결합해야 하기 때문이다.

(42) English Head-Complement Rule²⁵⁾

$$\left[\begin{array}{l} \text{phrase} \\ \text{COMP}_n < > \end{array} \right] \rightarrow \text{H} \left[\begin{array}{l} \text{SPR} < \text{X} > \\ \text{COMP}_n < \boxed{1} > \end{array} \right] \quad \boxed{1}$$

이러한 판단의 기저에는, 어순이 자유로운 언어는 그만큼 어순을 결정짓는 규칙에 제약이 적을 것이라는 가정이 놓여 있다. 그러나 이러한 분석이 보편적 타당성을 얻기 위해서는 문장의 어순과 **Valence** 구조의 상관관계에 대한 유형론적인 연구가 뒷받침되어야 할 것이다.

5. 맺음말

지금까지 한국어의 자유어순을 효율적으로 처리하기 위하여, 양분병렬구조 분석의 입장에서 새로운 VFS를 도입하고 이를 LKB에서 구현하여 그 타당성을 검증하여 보았다. 양분병렬구조분석은 단층구조분석이나 양분단선 구조분석에 비해 규칙의 개수가 적어 경제적이고 Sag et al. (2003)에서 제시한 기본 구조 및 원리를 크게 벗어나지 않는다는 점에서도 더 나은 것 같다. 이 논문에서 제안한 자질과 구조에 대한 이론적 연구와 유형론적인 비교 연구는 앞으로의 숙제라 하겠다.

Selected Reference

김종복·양재형. 2004. 조동사 복합술어 구문 분석 및 전산학적 구현. 어학 연구 40-1. 서울대학교 언어교육원.

25) (42)의 영어 HCR은 보충어들간의 고정된 순서를 표현하지 못하므로 불완전하다. 이를 구현하기 위해서는, 최대 2개까지의 보충어를 가정할 때, 이들 보충어 각각에 대해 하나씩의 HCR 규칙이 필요하다. 이러한 영어의 규칙수의 증가가 단점으로 보일 수도 있으나, 양분병렬구조분석을 이용하면 GAP과 관련된 규칙의 수를 줄일 수가 있게 되어 결국은 전체 규칙수를 감소시킨다. 그러나, 이 내용에 대한 자세한 논의는 지면의 한계상 생략한다.

- 신효필·고성연. 2003a. LKB로 구현한 한국어 단층구조(Flat Structure) 분석. 한국언어정보학회 2003년 학술대회 발표논문집.
- 신효필·고성연. 2003b. LKB(Linguistic Knowledge Building)에 기초한 한국어 단층구조 구현. 미발표원고.
- 양재형. 2002. LKB - Applications for Korean. 한국언어정보학회 구구조 문법연구회 2002-2학기 연구발표회 발표 자료.
- 장석진. 1993. 정보기반 한국어 문법. 언어와 정보.
- Copestake, Ann. 2002. *Implementing Typed Feature Structure Grammars*. CSLI Publications.
- Sag, Ivan A., Thomas Wasow, and Emily M. Bender. 2003. *Syntactic Theory: A Formal Introduction*, second edition. CSLI Publications.

영문 제목: An LKB Implementation of Free Word Order Variations in
Korean