

英語 構造에 關한 一般句構造文法的 考察

임 경 섭

I. 序 論

일반구구조문법 (Generalized Phrase Structure Grammar)은 70년대 말 내지 80년대 초에 Gerald Gazdar 등이 주창한 생성 문법 이론이다. 이 이론은 70년대의 해석 의미론자 (interpretive semanticist)와 생성의미론자 (generative semanticist) 사이의 언어 이론 전쟁 (linguistic war)에서 이진 해석의미론자들에 의한 변형생성문법 이론에 대한 하나의 경쟁 이론 중의 하나이다. Noam Chomsky가 주축이 된 변형생성문법은 표준 이론 (Standard theory), 확대표준이론 (Revised standard theory), 수정확대표준이론 (Revised extended standard theory) 또는 지배결속이론 (government and binding theory) 등으로 명칭을 바꾸며 발전되어 온 문법 이론으로 심리학적 측면에 중점을 두고 있다. 반면에 일반구구조문법 이론은 수학적인 측면에 중점을 둔 이론으로 몇 가지의 특성이 있다. 첫째는 생성 문법 (generative grammar)이라는 점, 둘째는 수학적인 형식 체제 (formal system)이라는 점, 셋째는 통사부와 의미부가 하나의 문법을 이룬다는 점이다. 이때 의미론은 몬테규의 이론에 기초를 두고 있다. 또 단층 문법 (monostratal grammar)으로 변형이 없는 문법 이론이다. 즉 심층구조와 표층구조의 구분을 버렸다. 그래서 모든 구절구조규칙에 의미번역규칙을 자기 부여하여 표층구조 하나만으로 통사적 일반화 및 의미 구조를 충분히 포착하였다. 즉 문장의 통사 구조와 의미 구조의 분석을 동시에 이룩하려고 하였다.

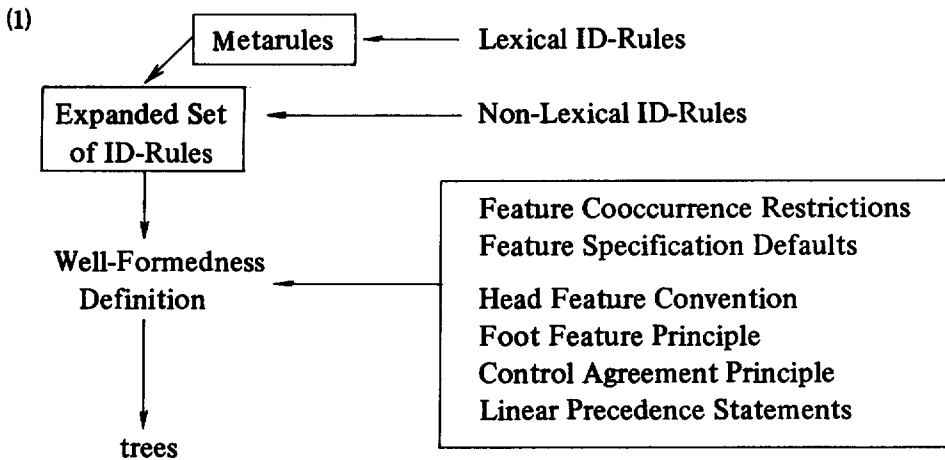
이러한 특성을 지닌 일반구구조문법은 이제까지 야기되어 왔던 관찰적 타당성 (observational adequacy), 기술적 타당성 (descriptive adequacy), 그리고 설명적 타당성 (explanatory adequacy)의 단계에서 흔히 무시되어왔던 기술적 타당성을 중요시하며, 설명적 타당성을 얻기 위한 전제조건으로 철저하면서도 자세한 기술이 필요하다는 방법론적 가설을 제시하고 있다. 특히 단층 기술 문법인 일반구구조문법은 자연 언어에 대해서 철저하면서도 자세한 기술을 하여야 한다. 왜냐하면 대부분의 논리적 번역은 구절구조규칙으로부터 예측할 수 있어야 하기 때문이다. 즉, 합성성의 원칙 (Compositionality principle)에 의하면 의미는 구절 구조에 기초를 두고 있기 때문이다. 다시 말하면 구절 구조 자체가 의미 구조인 것이다. 이처럼 자연 언어의 분석에는 주어진 문장의 구성 성분 요소 (constituent structure)가 어떻게 구성되어 있는지가 중요하다. 필자는 이러한 점에 착안하여 GKPS (1985)가 아직 다루고 있지 않는 영어의 몇 가지 구문이 어떠한 구성 성분 구조를 갖고 있는가를 알아 보고자 한다.

이러한 목적으로 2장에서는 일반구구조문법의 개요에 대해서, 3장에서는 소절 (small

clause)에 대해서, 4장에서는 명령문에 대해서 알아 보코자 한다.

II . 일반구조문법

일반구조문법이란 단층 기술 문법으로 오직 하나의 syntactic representation level 인 표층 구조만을 가지며, 변형규칙이 없이 오로지 한 종류의 syntactic object 인 구절구조규칙을 갖는다. 또 모든 syntactic structure 는 model-theoretic semantics 에 의한 의미 해석과 짝이 지어진다. 이러한 문법은 아래와 같은 골격을 갖는다.



먼저 syntactic category 에 관해서 알아 보기로 하면, 모든 syntactic category 는 feature specification 의 집합으로 취급되며, 이 feature specification 은 feature 와 value 의 쌍으로 이루어 졌다. 즉, < feature (name), feature-value > 의 형태로 된 순서쌍으로 feature 는 atomic symbol이며 feature-value 는 atomic symbol 이거나 category 이다. 또한 feature 가운데는 HEAD 나 FOOT 과 같은 특수한 하위자질 집합들이 있다.

- (2) a. HEAD = { N, V, PLU, PER, VFORM, SUBJ, PFORM, AUX, INV, PAST, PRD, ADV, SLASH, AGR, SUBCAT, BAR, LOC }
 b. FOOT = { RE, SLASH, WH }

위와 같은 feature 에 대한 feature-value 의 종류로는, PLU 에 대해선 +, - 의 value, BAR 에 대해선 0, 1, 2 의 value, CASE 에 대해선 NOM, ACC, GEN, ... 등의 value, PFORM 에 대해선 by, to, for 등의 value 를 가지며, AGR 에 대해선 NP [3 PER, + PLU] 과 같은 복합적 통사범주를 value 로 갖는다. 그러나 중요한 것은 모든 categories 는 feature 에서 value 로의 부분 함수 (partial function) 로 주어진다. 즉 부분 함수란 domain 의 어느 한 member 가 value 를 갖지 않아

도 되는 것을 말하는 것으로, 이는 꼭 필요한 것에만 적절히 value 를 주어 아주 일반화된 category 를 갖도록 하며 언급할 수 있도록 한다. 이처럼 undefined 된 category 는 뒤에서 이야기 할 FCR 과 FSD 에 의해서 legal extension 이 된다.

또 이 문장에서 \bar{x} -theory 를 받아들인다. 그래서 major category 인 N, V, A, P 를 primitive features 인 $\pm N$ 과 $\pm V$ 를 이용하여 아래와 같이 구분한다.

(3)

	[+N]	[-N]
[+V]	A	V
[-V]	N	P

특히 중요한 점은 GB-theory 와는 달리 S 를 V 의 projection 으로 본다 는 것이다. 즉 일반구조문법은 VP 와 S 를 V 의 최대 투영으로 취급한다. 그러나 S 는 VP 와 SUBJ feature 에 의해서 구별된다.¹ 또 하나의 주목할 점을 to-부정사의 to 를 V category 로 취급하는 것이다.²

GPSG 는 문법이 구절구조규칙의 집합으로 구성된다는 것을 수정하여 어떤 category 가 어느 구성 성분을 어떻게 지배하고 있는가를 명시하는 Immediate Dominance (ID) 와 어떤 category 가 직접 지배하는 sister nodes 사이의 어순을 명시하는 Linear Precedence (LP) 의 관계로 따로 분리하였다. 이와 같이 ID rules 의 집합과 LP statements 의 집합의 쌍으로 정의하는 것을 ID/LP format 이라고 부른다. 표시 방법은 ID rule 에서는 종래의 구절구조규칙과는 달리 category 사이에 commas 를 사용하여 각각의 category 가 상호 간에 순서가 없음을 의미하도록 하고 LP statement 에서는 ' < ' 의 관계로서 ' A < B ' 의 표시는 ' A 가 B 를 앞선다 ' 는 것을 뜻한다.

또 일반 구조문법은 head 를 'H' 라는 metavariable 을 사용하여 표시하며 동사들은 GB 이론과는 달리 subcategorization frame 을 갖지 않으며, 동사들이 나타나는 구조를 지적하는 numerical indicator 를 갖는다. 즉 SUBCAT feature 를 사용한다.

- (4) a. VP \rightarrow H[1]
 b. VP \rightarrow H[2], NP

위의 (4a) 규칙은 H[1] 이 head 이며 '1' 은 subcategorization information 을 갖는다. (4b) 도 마찬가지이다. 즉 타동사라는 것을 의미한다.

이 문법이론에서 lexical entry 는 다음과 같은 4 가지의 정보를 갖는다.

¹ V^2 [-SUBJ] = VP, V^2 [+SUBJ] = S

² to 를 특별한 nonfinite auxiliary verb 로 본다. (Pullum, Geoffrey K. 1982 참조)

- (5) a. $\langle weep, [[-N], [+V], [BAR\ O], [SUBCAT\ 1], \{ wept \}, weep' \rangle$
 b. $\langle devour, [[-N], [+V], [BAR\ O], [SUBCAT\ 2], \{ \}, devour' \rangle$

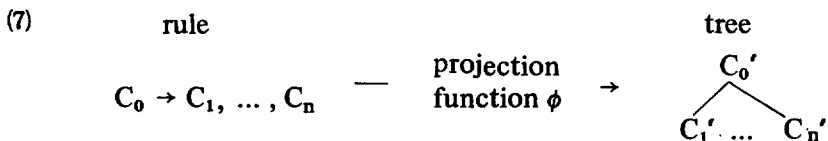
즉 phonological form, category, irregular morphology 의 지적과 meaning 을 갖는다. 이때 (5a)와 같은 lexical entry 는 (4a)의 규칙에, (5b)와 같은 entry 는 (4b)의 규칙에 선택될 수 있다. 왜냐하면 SUBCAT 의 수치가 서로 일치하기 때문이다.

또 ID rules 은 Lexical ID rules 와 Non-lexical ID rules 을 갖는다. 아래의 예문에서와 같이 전자는 lexical head 를 소개하는 규칙으로 SUBCAT 이 나타나지만 후자는 어떤 종류의 subcategorization properties 도 반영하지 않는다.

- (6) a. $VP \rightarrow H[3], NP, PP[to]$
 b. $S \rightarrow NP, VP$

위의 2가지 ID rules 에 덧붙여 Metarules 가 있는데, 이는 이미 문법 속에 포함되어 있는 것에 기초를 두고 새로운 것을 정의하는 체계적인 방법으로 ID rules 의 집합의 principled enlargement 를 정의하는 것이다. 어느 의미에서 보면, metarules 의 역할이 변형문법에서의 transformations 의 역할과 비슷하게 보인다. 그러나 각각의 expressive power 에서 커다란 차이가 있다. 즉 transformation 은 derivational strings 를 derivational strings 로 map 하는 것이나, metarule 은 ID rules 을 ID rules 로 map 하는 것이다. 만약 CF-PSG (Context Free · Phrase Structure Grammar) 에 transformations 를 추가하면 이 이론의 expressive power 를 바꾸는 결과가 되어, 구조는 별개의 'deep structure' 와 'surface structure' 의 levels 이 주어진다. 그러나 metarules 을 이런 식으로 추가하면, 단지 전체의 문법 속에 체계적이며 잘 정의된 방식으로 rules 의 집합을 확장시키는 결과를 갖게 된다. 즉 이 이론의 expressive power 는 변경되지 않는다.

또 syntactic description 의 중요한 한 과제는 ID rules 와 trees 사이의 관계를 잘 설명하는 것인데, 이는 아래와 같이 rules 을 trees 로 투사하는 함수로서 생각한다.



이때 ID rules 을 structural descriptions 로 map 하는 이론의 핵심은 local admissibility 라는 관계로서, 이는 한 local tree 가 rule 으로부터 locally admissible 하려면 아래의 경우들을 만족시키는 admissible projection 이 되어야 한다.

- (8) a. all Feature Cooccurrence Restrictions are true of ϕ (C).
 b. ϕ (C) is compatible with all Feature Specification Defaults.
 c. ϕ meets the Head Feature Convention, the FOOT Feature Principle, and the Control Agreement Principle.
 d. ϕ respects all LP-statements.

위의 내용들을 좀더 설명하기로 한다. 먼저 Feature Cooccurrence Restrictions (FCRs)은 한 언어에 있어서의 가능한 categories 의 부류를 제약하는 것으로, ‘legal extension of a category’ 의 개념과 ‘fully specified syntactic category’ 의 개념의 정의의 한 부분을 형성한다. 그리고 FCRs 은 features 사이의 어떤 의존관계를 나타내며, 또 categories 가 underspecified 에서 fully specified 로 될 때 지켜져야 하는 절대 제약이다.

- (9) a. FCR 2 : [VFORM] \supset [-N, +V]
 b. FCR 6 : [SUBCAT] \supset ~ [SLASH]

위의 (9a)는 만약에 한 category 가 VFORM feature 를 가지면 그것은 반드시 [-N, +V] 이어야 한다는 것이며, (9b)는 gap이 있는 구성 성분은 phrasal constituent이기 때문에 lexical head 만이 가지고 있는 SUBCAT feature 에 대해선 undefined 된다는 것을 의미한다.

Feature Specification Default (FSD)는 어떤 feature에 대해서 전혀 언급이 없을 때에는 특정한 feature 가 자동적으로 첨가된다는 것이다.

- (10) a. FSD 1 : [-INV]
 b. FSD 8 : [NFORM] \supset [NFORM NORM]

위의 (10a)는 어떤 ID rule 이나 다른 일반 제약에 의해 [+INV] 라고 상술되지 않으면 항상 [-INV] 가 된다는 것을 뜻한다. (10b)는 일반적으로 NFORM 의 값은 NORM이 된다는 뜻이며, 만일 ID rule 이나 다른 일반제약에 의해 *there* 나 *it* 와 상술되면 이 FSD는 무시될 수 있음을 말하는 것이다.

특히 앞에서 언급한 FCR 은 ID rules 와 trees 사이의 관계에 있어서 feature 의 inheritance 와 instantiation 과는 무관하게 항상 지켜져야 하는 절대 제약인 반면에, FSD 는 instantiation 에 관한 제약이다. 여기서 ID rules 에 의해서 직접 결정되는 feature specification 이 그 categories 에 inherited 되었다고 하며, categories 에 자유롭게 feature specification 이 첨가되는 것은 instantiated 되었다고 한다.

이밖에도 몇 가지의 feature instantiation 에 대한 일반적인 원칙이 있다. 즉 mother category 와 head daughter category 간의 feature 의 조응 관계를 설명하는

Head Feature Convention (HFC) , sister categories 간의 features 에 관한 Control Agreement Principle (CAP), mother category 와 non-head daughter category 간의 관계를 설명하는 Foot Feature Principle (FFP)가 있다. 먼저 HFC 부터 알아 보자. 그 정의는 아래와 같다.

(11) *Definition 8 : Head Feature Convention (HFC)*

Let ϕ_r be the set of projections from r which meet the FFP and the CAP, and where $r = \langle \langle C_0, \{C_1, \dots, C_n\} \rangle m \rangle, W_H \rangle$. Then

$\phi \in \phi_r$ meets the HFC on r if and only if

(i) $\forall C_i \in W_H [\phi(C_0) \cap \psi(C_i, \phi_r) | \text{HEAD} \subseteq \phi(C_i) | \text{HEAD}]$

(ii) $(\bigcap_{C_i \in W_H} \phi(C_i) \cap \psi(C_0, \phi_r)) | \text{HEAD} \subseteq \phi(C_0) | \text{HEAD}$

(iii) $\forall C_i \in W_H [\text{BAR} \in \text{DOM}(\phi(C_0)) \cap \text{DOM}(\phi(C_i))]$

위의 HFC는 ID rules 을 structure로 map 하는데 지켜야 할 것으로 local tree 에서 categories 가 갖고 있는 자질들이 어떻게 nodes 에 분배되어지는가에 대한 제약이다. 간략하게 말하자면, 첫째는 오직 하나의 head 만 있을 경우에는 head 의 HEAD feature specifications 는 mother 와 head 의 free feature specifications 의 intersection에 의해 생긴 category 의 HEAD features 의 extension 이며, mother 의 HEAD feature specifications 는 head 와 mother 의 free feature specifications 의 intersection에 의해 생긴 category 의 HEAD features 의 extension 이다. 둘째는 multiple heads 이나 ‘problematic’ feature specification 이 일어나지 않는 경우로서 mother 의 HEAD feature 는 모든 head daughters 의 intersection 을 취함으로써 생겨난 category 의 HEAD features 와 동일하다는 것이다. 마지막으로 ‘head’ 라는 개념은 X-bar system의 가정과 밀접한 관계가 있는 것으로, minor categories (즉 BAR specification이 결여된 instantiated categories)는 heads 가 되거나 heads 를 가질 수 없다. 따라서 mother 와 모든 head daughters 에는 BAR specification (반드시 똑같은 specification 은 아님)이 나타나도록 표현해야 한다.

다음은 CAP 에 대해서 알아 보자. CAP 의 정의는 아래와 같다.

(12) *Definition 6: Control Agreement Principle (CAP)*

Let ϕ_r be the set of projections, from r , where $= C_0 \rightarrow C_1, \dots, C_n$.

Then $\phi \in \phi_r$ meets the CAP on r if and only if

(i) if $\phi(C_j)$ controls $\phi(C_i)$, then

$\phi(C_i)(f_i) = x(\phi(C_j) \cup \phi(C_i)) | \{f_i\}$, where f_i is the CONTROL feature of $\phi(C_i)$.

- (12) (ii) if there is a $\phi(C_i)$ which is a predicative category with no controller, then $\phi(C_i)(f_i) = \phi(C_0)(f_0)$, where f_i and f_0 are the CONTROL features of $\phi(C_i)$ and $\phi(C_0)$, respectively.

위의 CAP 는 의미적 요소와 통사적 요소를 함께 가지고 있는 것으로 controller 와 controllee 의 개념에 바탕을 두고 있다. 이때 control 의 개념은 functor-argument 의 관계에 기초를 둔다. 예를 들면, 하나의 VP 가 하나의 NP 를 argument 로 취하여 문장 해석이 된다고 할 때 이 VP 가 controllee 가 되며 이때의 argument 인 NP 는 controller 가 된다. 즉, NP 와 VP 사이의 agreement 현상을 설명하는 것이다. 위의 정의를 간략하게 알아보자. 첫째는 local tree 안의 potential agreement target C 가 controller C' 을 갖는 경우에는 C 의 CONTROL feature 의 값은 C' 와 같아야 한다. 둘째는 C 가 controller 가 없는 predicative category³ 인 경우에는 C 의 CONTROL feature 의 값은 그것의 mother 인 C₀ 의 CONTROL feature 의 값과 동일해야 한다. 다음은 FFP 에 대한 정의이다.

- (13) *Definition 2: Foot Feature Principle (FFP)*

Let Φ_r be the set of projections from r , where $r = C_0 \rightarrow C_1, \dots, C_n$.

Then $\phi \in \Phi_r$ meets the FFP on r if and only if

$$\phi(C_0) \mid \text{FOOT} \sim C_0 = \bigsqcup_{1 \leq i \leq n} \phi(C_i) \mid \text{FOOT} \sim C_i$$

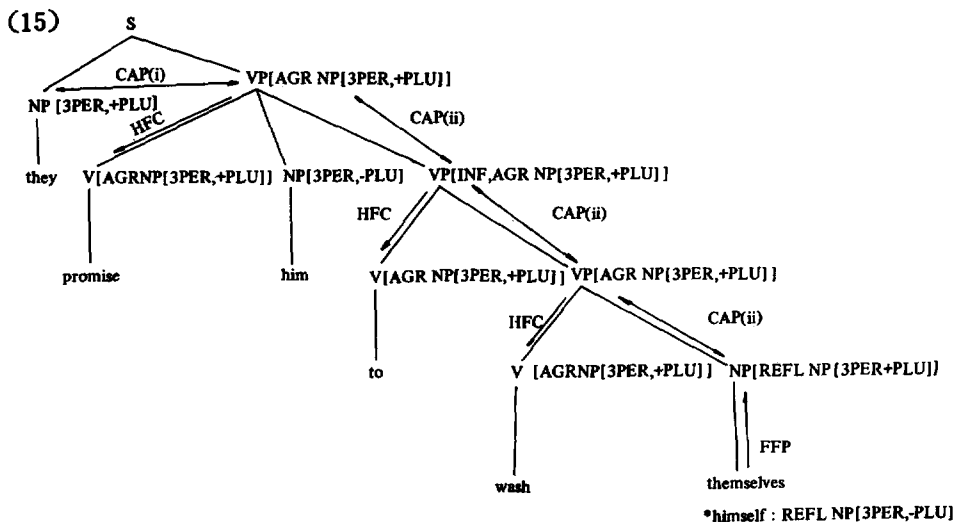
FFP 는 (2b)에서 나타난 FOOT features 에 관한 원리로서 대략 다음과 같은 뜻이다. 즉, tree 에서 mother category 에 instantiated 된 FOOT feature specifications 는 모든 daughter categories 에 있는 instantiated FOOT features specifications 의 unification 과 동일하여야 한다는 뜻이다.

이상의 3 가지 원리들이 아래의 예문에서 어떻게 작용하는지 살펴 보자.

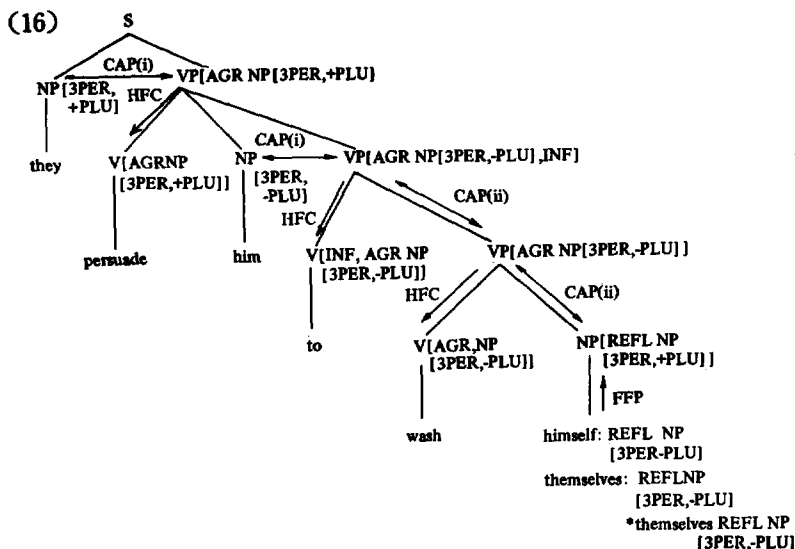
- (14) a. They promise him to wash themselves.
 b. *They promise him to wash himself.
 c. They persuade him to wash himself.
 d. *They persuade him to wash themselves.

(14a,b)를 아래의 tree 로 나타낼 수 있다.

³ GKPS (1985) p. 86 참조.



위의 tree 에서 볼 수 있듯이 VP 들은 각각의 head daughter 인 V들과 HFC 에 의해서 features 가 연결된다. 또한 CAP (i) 에 의해서 AGR 값이 controllee 인 VP 에 연결되고, 그 다음의 local tree 에서는 controller 가 없기 때문에 control feature 인 AGR 이 daughter VP 에 연결된다. 이때의 local tree 에도 controller 가 없어서 daughter VP 에 연결된다. 이와 동시에 themselves 의 FOOT feature REFL 이 FFP 에 의하여 삼투되어 올라와, VP 의 AGR 값과 REFL 의 값이 일치된 themselves 는 정문이 되나, 값이 일치하지 않는 himself 는 비문이 된다. 그러나 (14c,d) 에서는 다르다. 다음의 tree 를 보자.



위의 tree 에서 볼 수 있듯이 persuade 는 semantic type 이 promise 와 달라서 him이 controller 가 되어 CAP(i) 에 의해서 오른쪽의 VP 의 AGR 값을 결정한다. 그래서 FFP에 의해 삼투된 REFL 값 중에서 VP 의 AGR 값과 일치한 himself 일 경우에는 정문이나, themselves 일 경우에는 REFL 값과 AGR 값이 일치하지 않아서 비문이 된다.

이상의 FCR 과 FSD 그리고 3가지의 원리를 만족시키고, 마지막으로 LP(linear precedence) statement 를 지킨 tree 만이 admissible 하다. 이때 LP statement 란 어떤 category 가 직접 지배하는 sister nodes 사이의 어순 관계를 설명하는 것이다. 아래의 예를 통해서 알아 보자.

(17) [SUBCAT] <~ [SUBCAT]

위의 LP 는 lexical category 가 non-lexical category 를 항상 앞선다는 뜻으로, VP 에서는 항상 V가 선행하고, NP 에서는 determiner 가 앞선다는 등의 일반성을 포착한 것이다. 그러나 이러한 LP statement 는 각 언어에 따라 차이가 있다. 즉 어떤 언어는 고정된 어순을 갖는 반면에 (예 : 영어, 불어 등.) 다른 언어는 어느 정도의 자유어순을 허용한다.(예 : 한국어, 일본어, Walpiri 등.)

이제까지의 설명은 통사부에서 admissible tree 를 만족시키기 위한 원리와 제약들에 관한 것이다. 다음은 간단하게 의미부가 하는 일을 알아 보자

GPSG 에서는 의미부를 'Extended Montague Grammar' 의 semantics 에 기초를 두고 있다. 이 Montague semantics 는 자연 언어 의미론에 대해서 model-theoretic 하며 truth-conditional 한 접근 방법을 택한다. 특히 한 언어의 의미 이론의 기본적인 임무중의 하나는 언어의 well-formed expressions 의 의미 해석을 귀환적으로 결정하는 것인데, Montague 에 의해 발전된 model-theoretic semantics 에는 2가지로 나뉘어 졌다. 첫째는 각각의 통사적으로 결정된 category 표현의 possible denotations 를 상술하는 것이고, 둘째는 complex expression 의 denotations 가 그들의 constituents 의 denotations 의 function 으로 생성되는 방법을 상술하는 것이다. 그러나 Montague 는 syntactic category 가 직접 sets of possible denotations 로 나아가는 것 보다는 artificial language 인 Intensional logic (IL) 을 통해서 간접적으로 나아가는 것이 더 편리하다는 것을 알고 IL 을 사용하였다. IL 의 기본 types 은 e와 t이고, 이를 기초로 complex types 이 귀납적으로 정의된다. GPSG 에 있어서 통사 범주와 type 의 관계는 TYP 함수로 정의된다. TYP 는 domain 이 syntactic categories 인 function 이어서, X가 하나의 category 일 때마다 TYP(X) 는 IL 의 type 이다.

또한 grammatical relations 에 대해서는 LFG (lexical-functional grammar)와 달

리⁵ Montague의 syntax와 semantics 이론의 framework에 입각한 Dowty의 제안을 받아들인다. 그의 설명은 주어는 문장을 만들기 위해 VP와 맨 마지막으로 연결된 term이고, 직접목적어는 주어 term 바로 직전에 VP와 연결하는 term이고, 간접목적어는 직접목적어 term 바로 직전에 연결하는 term이라는 것이다.

또한 Bach의 rule-to-rule hypothesis를 받아들인데, 이는 문법 내의 모든 syntactic rule에 대해서, 그 규칙에 의해 분석되는 일종의 structures가 어떻게 해석되는가를 상술하는 상응한 semantic rule이 기술되어야 한다는 것이다. 그리고 번역이 syntactic tree와 어떻게 관련되는가를 도식적으로 보이기 위해서 interpreted tree의 장치를 받아들인다. 이 interpreted tree는 일반의 constituent structure tree와 비슷하나 각각의 nodes가 syntactic categories로서 뿐만 아니라 IL의 meaningful expressions에 의해 명칭이 붙을 수 있다는 점이 다르다. 그러나 현행 GPSG에서는 이러한 standard rule-to-rule hypothesis를 TYPE-driven Translation으로 그리고 local trees에 나타난 정보를 참조하여 translation한다는 것으로 수정한다.

마지막으로, Semantic Interpretation Schema⁶는 문법에 의해 허용되는 interpreted tree의 집합에 대한 유도적인 정의를 제공하는 것으로, 이것은 기본적으로 compositional translation schema로 구성된다. 즉 translations이 daughter nodes에 할당되어진 interpreted tree가 주어지면, 이 schema는 root node가 daughters의 번역의 functions인 translation이 할당된 새로운 interpreted tree를 생성한다. 이러한 유도의 기초는 terminated interpreted local trees의 집합이다. 이때 terminated local tree는 오직 daughter가 terminal symbol - 언어의 단어 - 인 tree이다. 그리고 terminated local tree의 properties는 lexicon, 특히 tree의 daughter에 대한 lexical entry에 의해서 결정된다. 그래서 하나의 terminated interpreted local tree는 root node가 lexical category 뿐만 아니라 translation에 의해서 명명되는 것이다. 그리고 non-local tree에 semantic interpretation을 할당하는 절차는 lexical categories에 의해 명명된 nodes에서 시작하여, Semantic Interpretation Schema에 입각하여 그들의 mothers의 번역을 계산하고, 그 다음 level까지 이러한 절차를 계속하여 tree의 root에 도달할 때까지 한다.

다음 장에서는 이러한 GPSG framework 내에서 아직은 미진한 소절과 명령구문을 연구하고자 한다.

Ⅲ. 소절

소절에 관해서는 여러 학자들이 논의하였는데 여기서는 간단히 소절의 개념과 몇 사람

⁵ LFG에서는 grammatical relations를 theoretical primitives로 받아들인다.

⁶ GKPS (1985) pp 230-31 참조.

의 논의를 소개하고 GPSG framework로서의 해결 방법을 제시한다.

- (18) a. They consider that John is honest
 b. They consider John to be honest.
 c. They consider John honest.

위의 예에서 (18c)를 보문소절이라고 부르며 의미는 (18. a,b)와 같다고 한다. 그러나 (18c)의 constituent structure를 어떻게 볼 것인가의 문제가 있다. Chomsky(1981)에서는 (18c)의 John honest가 하나의 소절을 이룬다고 하였다. 그는 아래의 예문에서 그의 주장을 더욱 정당화하였다.

- (19) a. John considers Bill_i angry at himself_i
 b. *John_i considers Bill angry at himself_i
 c. John_i considers Bill angry at him_i
 d. *John considers Bill_i angry at him_i

위의 a, c는 정문인 반면에 b, d가 비문인 이유는 Bill angry at him/himself가 하나의 소절을 이루듯이 대명사나 재귀대명사의 형태를 취한다는 것 때문이다. 즉 (19a)는 John이 절의 boundary를 넘어선 명사를 재귀대명사화하였기 때문에 비문이 된 것이며, (19d)는 한 소절안에서 두 명사가 동일인을 지칭하는데도 재귀대명사화를 하지 않았기 때문에 비문이 된 것이다.

또한 다음과 같은 예도 있다.

- (20) a. John left the room angry.
 b. John left the room empty.
 c. John ate the meat naked.
 d. John ate the meat raw.

위와 같은 소절은 부가소절이라고 불린다. 이때는 controller가 어떤 것인가가 매우 중요하다. 즉 (20 a,b)의 예문들은 똑 같은 형태의 구문처럼 보이는데도 controller가 각각 다르다. (20a)의 angry의 controller는 주어인 John인 반면에 (20b)의 empty의 controller는 목적어인 the room인 것이다. 마찬가지로 (20c)에서는 주어인 John이 controller이며 (20d)에서는 the meat가 controller이다. 이와 같은 문제를 해결하기 위해서 GB에서는 PRO를 설정하여 control 관계를 설명하려고 하였다. 그러나 이 예문들을 자세히 살펴보면 커다란 차이점이 있다는 것을 알 수 있다. 즉 (20a)와 (20b)에서 쓰이고 있는 동사 left가 의미상으로 차이가 있다는 점이다. (20a)에서는 「떠나다」라는 의미가 있는 반면에 (20b)에서는 「남겨두다」의 의미가 있는 것 같다. 이는 곧 left가 2개의 subcategorization을 가지고 있다고 볼 수 있다. 그래서 GPSG에서는 (20a)와 (20b)의 left를 자기 다른 SUBCAT번호를 갖는 것으로 해결

하고자 할 수 있다. 그리하여 (20a)의 left 에 SUBCAT 번호가 부여되면 angry 의 controller 는 주어인 John 이 되도록 하게되며, (20b)의 left 에 다른 SUBCAT 번호를 부여하여 empty 의 controller 가 the room 이 되도록 하면된다. (20c,d)도 이와같이 하여서 (20c)에서는 naked 의 controller 를 John 이 되도록 하며, (20d)에서는 raw 의 controller 를 the meat 로 하면 된다.⁷

또한 아래와 같이 소절을 분류하기도 한다.⁸⁾

- (21) a. I consider [_A John [_A stupid]]
 b. I consider [_N John [_N a fool]]
 c. I expect [_P the sailor [_P off my ship]]
 d. I heard [_V Jack [_V come into the kitchen]]
 e. I feared [_{+V} Jack [_{+V} killed by enemy]]

이상과 같은 분석들은 결국 각 예문들의 constituent structure 를 어떻게 보느냐에 차이가 있음을 알 수 있으며, 또 각각의 category 의 status 를 명시하여 의미해석을 하려고 함을 알 수 있다. 그러나 아직까지의 설명들은 명쾌하지 못한 것 같다. 특히 표층구조를 중시하는 GPSG에서 볼 때 더욱 설명력이 떨어진다. 왜냐하면 상상키 어려운 심층구조를 설정하여 설명하려 하기 때문이다. 그러나 GPSG도 아직은 이러한 문제를 다루지 않고 있다. 그래서 필자는 이러한 예문들은 GPSG framework 으로 해결하여 보고자 한다.

모든 문법 이론이 그러하듯이 GPSG도 constituent structure 를 중시한다. 왜냐하면 syntactic tree 가 곧 의미 해석의 기초가 되기 때문이다. 그래서 필자는 다음과 같은 ID rules 을 제시하여 문제를 해결하고자 한다.

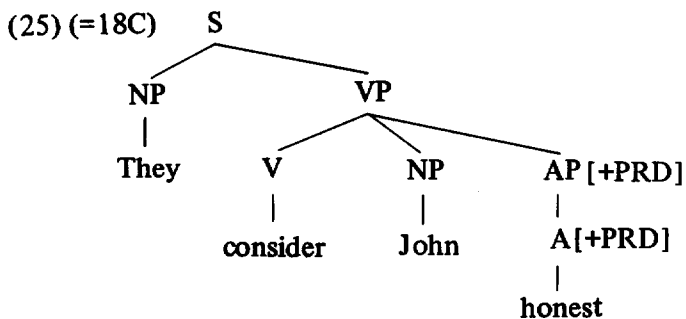
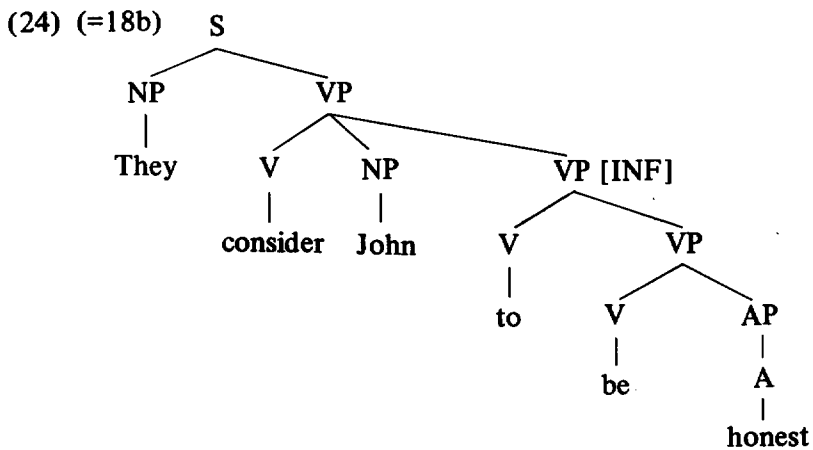
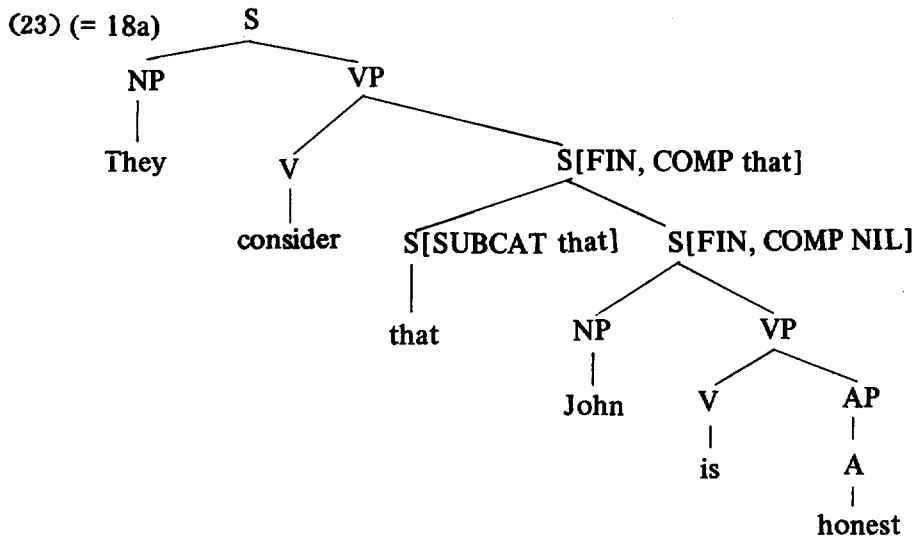
- (22) a. VP → H[], S[FIN] ... consider (18a)
 b. VP → H[], NP, VP[INF] ... consider (18b)
 c. VP → H[], NP, XP [+PRD] ... consider (18c)

위에서 볼 수 있듯이 Small clause 의 구문은 [+PRD]⁹ 를 갖는다고 하여서 해결하고자 한다. 그러면 (22a,b,c)의 각각의 tree 는 아래와 같다.

⁷ SUBCAT 번호를 달리하여 semantic type 에 의해 controller 가 어떤 것인지를 밝혀 준다.

⁸ Stowell, Tim (1981) 참조.

⁹ [+PRD] 는 (copula 다음에 나타나는) predicative complement phrases 에 사용되며, [-PRD] 는 non-predicative phrases 에 사용된다.



위의 각각의 의미 해석은 아래의 예문에서와 비슷하게 할 수 있겠다.

(26) a. Terry believes that Lou is insane.

b. Terry believes Lou to be insane.

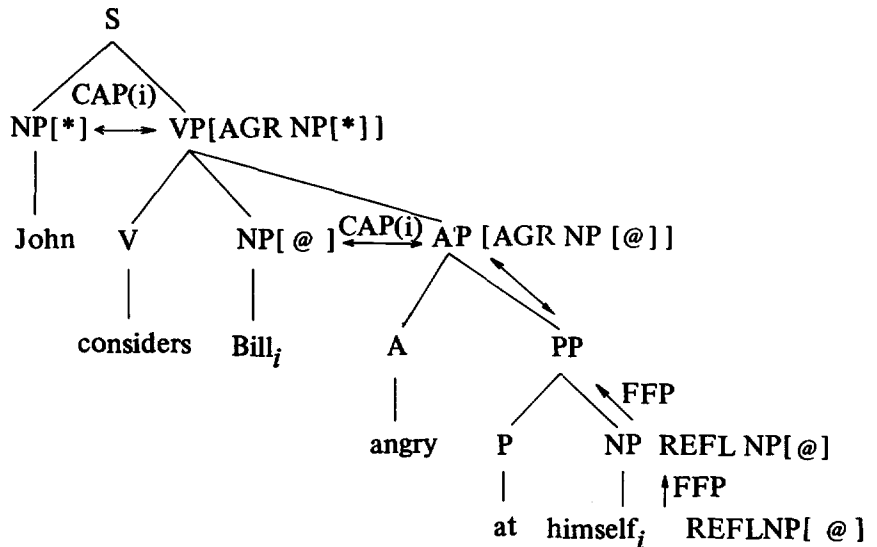
위의 예문들의 동사는 각기 다른 SUBCAT 번호를 갖는다.

- (27) a. VP → H[40], S[FIN] ... believe (←26a)
 b. VP → H[17], NP, VP[INF] ... believe (←26b)

그래서 H[40]의 translation은 believe'이며 semantic type은 <S, <NP, S>>이고 H[17]의 translation은 $f_R(\text{believe})$ 이며 semantic type은 <VP, <NP, <NP, S>>>가 된다.¹⁰

위에서와 마찬가지로 (18a)에서의 consider 동사는 <S, <NP, S>>의 semantic type을 가지며, 전체의 의미는 consider'(is honest'(Jack*))(they*)가 된다. 또 (18b)의 consider는 <VP, <NP, <NP, S>>>의 semantic type을 가지며, 전체 문장의 의미는 $f_R(\text{consider})$ (to-be-honest')(John*)(they*) ↔ consider'(to-be-honest'(John*))(they*)가 된다. 또 (18c)의 consider는 <AP, <NP, <NP, S>>>의 semantic type을 가지며 전체 문장의 번역은 $f_R(\text{consider})$ (honest')(John*)(they*) ↔ consider'(honest'(John*))(they*)가 된다.¹¹

(28)¹²



¹⁰ H[17]의 전체 문장의 의미 해석은 $f_R(\text{believe})$ (to-be-honest')(Lou*)(Terry*) = believe'(to-be-honest')(Lou*)(Terry*).

¹¹ 결국 (18a), (18b) 그리고 (18c)의 의미가 동일하다는 것은 meaning postulate에 의해서도 확인된다.

¹² GKPS (1985)에 의하면, FCR12: [AGR] ⊃ [-N, +V] 때문에 AP는 AGR 값을 가질 수 없으나, 한국어 등에서는 AP도 AGR 값을 가질 수 있기 때문에 필자는 FCR12를 수정하여 [AGR] ⊃ [-N]으로 취급한다. 즉 AP도 AGR 값을 갖는다.

이러한 방법에 의하면 Chomsky(1981)에서 제시되었던 예문들도 잘 해결된다. (19a)의 tree를 그려서 생각하기로 하자

즉 위의 문장의 동사 consider의 semantic type은 $\langle AP, \langle NP, \langle NP, S \rangle \rangle \rangle$ 로서 persuade동사와 semantic type이 같다. 그래서 AP의 AGR의 값은 NP[@]인 Bill이 된다. 이는 CAP(i)에 의한 것이다. 그리고 himself의 REFL 값이 FFP에 의해서 삼투되어 AP의 AGR 값과 같게 되어서 정문이 된 것이다. 그러나 (19b)의 문장은 himself의 REFL 값인 NP[@]와 VP의 AGR 값인 NP[*]와 틀리게 되어 비문이 된다.¹³

마지막으로 (20a, b)의 문장들에 대한 의미해석규칙을 알아 보기로 한다. 먼저 우리는 아래와 같은 예를 잘 알고 있다.

- (29) a. Kim persuades Lee to go.
b. Kim promises Lee to go.

위의 두 문장은 구조상으로는 동일하나 persuade와 promise의 semantic type은 각기 다르다. 즉 persuade의 semantic type은 $\langle VP, \langle NP, \langle NP, S \rangle \rangle \rangle$ 이고 promise의 semantic type은 $\langle NP, \langle VP, \langle NP, S \rangle \rangle \rangle$ 이다. 그리고 각각의 translation은 $f_E(\text{persuade}')$ 와 $\lambda \mathcal{S}[f_E(\text{promise}'(\mathcal{S}))]$ 이 된다. 다시 말하면 (29b)에서는 to go의 controller가 주어인 Kim인 반면에, (29a)에서는 to go의 controller가 목적어인 Lee이다는 것이다. 이와 같은 방법에 의하면 앞에서 언급하였듯이 (20a, b)의 left의 semantic type도 각기 다르게 된다. 즉 (20a)의 left는 promise의 semantic type과 같은 $\langle NP, \langle VP, \langle NP, S \rangle \rangle \rangle$ 가 되며 이 때의 번역은 $\lambda \mathcal{S}[f_E(\text{left}'(\mathcal{S}))]$ 가 된다. 그리하여 전체 문장의 번역은 $f_E(\text{left}'(\text{the room}'(\text{angry}'(\text{John}^*))) \Leftrightarrow \text{John}^*(\lambda x[\text{left}'(\text{the room}'(\text{angry}'(x)))(x^*)])$ 로서 '존은 화를 내며 방을 떠났다'의 의미가 된다. 반면에 (20a)의 left는 persuade의 semantic type과 같은 $\langle VP, \langle NP, \langle NP, S \rangle \rangle \rangle$ 가 되며 번역은 $f_E(\text{left}')$ 이 된다. 그리하여 전체 문장의 번역은 $f_E(\text{left}'(\text{empty}'(\text{the room}'(\text{John}^*))) \Leftrightarrow (\text{the-room}') \lambda x[\text{left}'(\text{empty}'(x^*))](x^*) (\text{John}^*)$ 로서 '존은 방을 비워두었다'의 뜻이 된다. 다시 말하면 (20a)의 angry의 controller는 주어인 John이며 (20b)의 empty의 controller는 목적어인 the room이다. 이와 마찬가지로 (20c)에서 naked의 controller는 주어인 John이 된다. 왜냐하면 ate의 semantic type은 $\langle NP, \langle VP, \langle NP, S \rangle \rangle \rangle$ 이며 번역은 $\lambda \mathcal{S}f_E(\text{ate}'(\mathcal{S}))$ 가 되기 때문이다. 또 (20d)에서 raw의 controller는 목적어인 the meat가 된다. 왜냐하면 이때의 ate의 semantic type은 $\langle VP, \langle NP, \langle NP, S \rangle \rangle \rangle$ 가 되며 번역은 $f_E(\text{ate}')$ 이 되기 때문이다.

¹³편의상 NP[*]와 NP[@]을 사용한다.

위에서 보았듯이 소절의 구문은 VP → H[], NP, XP[+PRD]의 ID rule을 갖는 것으로 소절이라는 말이 맞지 않게 된다. 이때의 번역은 각각의 동사가 갖는 semantic type에 의해서 결정된다. 즉 XP[+PRD]의 controller가 무엇인지를 알 수 있다. 이는 심층구조에 의해서 소절을 이룬다고 하여 문제를 더욱 야기시키는 것보다, 표면구조에 충실하면서 동사의 어휘의 속성에 따라 의미 해석을 체계적으로 하는 GPSG의 framework가 더욱 잘 기술하며 설명한다는 것을 보여준 것이다.

다음 장에서는 영어의 명령 구문에 대하여 살펴 본다.

IV. 명령문 (Imperative Construction)

우리는 영어 문장 가운데서 아래와 같이 어떤 동사들은 주어로서 특정한 NP만을 요구하는 것을 알고 있다. 그래서 category theory도 서로 다른 종류를 구분하는 것이 필요하다. 그 중에서 NP category의 종류들을 아래의 예문을 통해서 알아 보자

- (30) a. It appears that Kim is unhappy.
 b. It bothers us that Sandy can't come.
 c. It would be unlikely for Pat to be called upon.

- (31) a. There is nothing in the box.
 b. There is a pig roasted.
 c. There were two women teaching the class.

위의 예문들에서 expletive pronouns *it*와 *there*는 지극히 제한된 분포를 갖는다. 구체적인 예를 들어 보면 아래의 (32)와 (33)처럼 어떤 expletive pronouns은 어떤 VPs의 주어는 될 수 없다.

- (32) a. *It put the book on the table.
 b. *It said that Kim was happy.
- (33) a. *There put the book on the table.
 b. *There preferred for Sandy to get the job.

반면에 아래의 (34)와 (35)에서 볼 수 있듯이 어떤 'ordinary' NPs는 expletive subjects를 취하는 VPs의 주어로 나타나지 않는다.

- (34) a. *Sandy appears that Kim is happy.
 b. *Danna would be likely for Pat to be called upon.
- (35) a. *Robin is nothing in the box.
 b. *He was a pig roasted.

이와 같은 여러 종류의 NP 를 구분하기 위해서 GPSG 에서는 NFORM 이라는 feature 를 사용한다. 그리고 그 값으로 아래와 같은 것들을 갖는다.

(36) { *it*, *there*, NORM }

위의 값 중에서 2개의 expletive pronouns *it* 와 *there* 는 아래와 같은 lexical entries 를 포함하는 categories 가 된다.

(37) a. <*it*, NP[PRO, -PLU, NFORM *it*], { }, Δ>
 b. <*there*, NP[PRO, NFORM *there*], { }, Δ>

그리고 'ordinary' NPs 는 아래의 (38)처럼 되며, NFORM 의 default value 는 NORM 이라는 것을 확인해주는 (39)와 같은 FSD 8이 있다.

(38) [NFORM NORM] ← 'ordinary' NPs
 (39) FSD 8: [NFORM] ⊃ [NFORM NORM]

또한 아래의 예문에서 보듯이, 영어의 어떤 동사는 주어로서 expletive pronoun *it* 를 가지며 'extraposed' 위치에 sentential constituent 를 취한다.

(40) a. It bothered Lou that Robin was chosen.
 b. It should be obvious that they will resign.
 c. It would bother Lou (for Chris) to do that.
 d. It seems that Pat was unhappy.
 e. It appears that Leslie will be late.

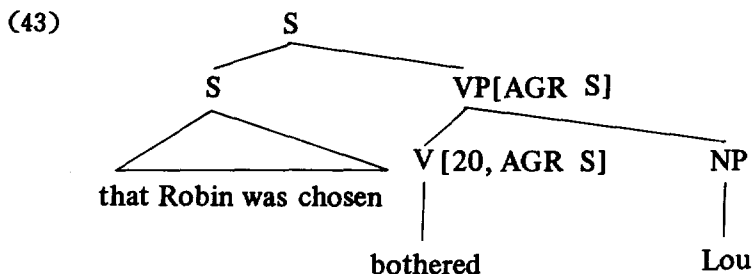
그러나 아래의 예문처럼, 전부는 아니지만 그 중 몇 개는 sentential subject 를 허용한다.

(41) a. That Robin was chosen bothered Lou.
 b. That they will resign should be obvious.
 c. (For Chris) to do that would bother Lou.
 d. *That Pat was unhappy seems.
 e. *That Leslie will be late appears.

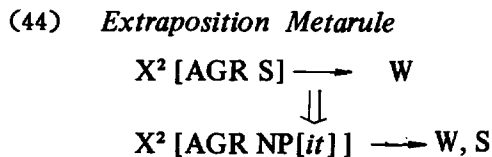
이러한 예문들은 해결하기 위해선 아래와 같은 ID rule 이 필요하다.

(42) a. VP [AGR S] → H[20], NP
 b. bother, amuse, ...
 c. bothered Lou.

그래서 위의 (41a)의 구문은 아래와 같은 tree가 된다.

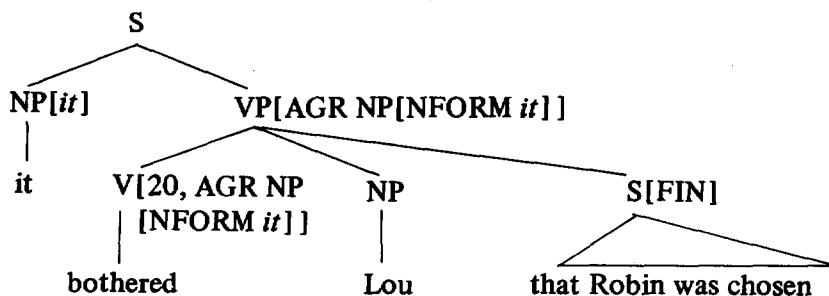


그리고 (40a)의 구문과 (41a)의 구문은 아래의 metarule에 의해서 연결된다.



즉, (44)의 metarule에 의해서 (45)와 같은 ID rule이 나오며 (46)과 같은 tree가 그려진다.

- (45) a. $VP[AGR NP[it]] \rightarrow H[20], NP, S$
 b. bother, amuse, ...



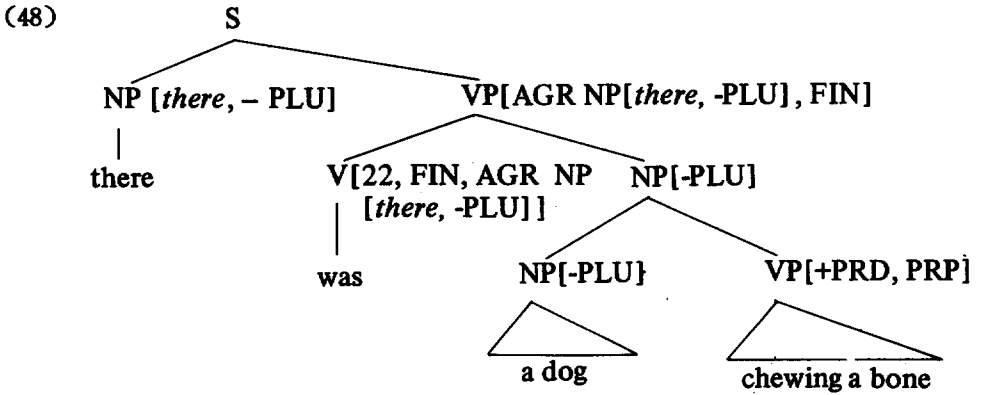
위에서 다루었던 expletive pronoun *it*와 비슷하게 *there*도 설명된다.

- (47) a. There was a lion in the zoo.
 b. There were three wolves sick that day.
 c. There was a dog chewing a bone.

위의 예문들을 해결하기 위해선 아래와 같은 ID rule이 필요하다.

- (48) a. VP[AGR NP [*there*, α PLU]] → H[22], NP[αPLU]
- b. be, ...
- c. was a lion in the zoo.

위의 (47c)를 tree 로 나타내면 아래와 같다.



우리는 아래와 같은 구문을 볼 수 있다.¹⁴

- (49) a. In the garden is a fountain.
- b. Near the fountain sat a large purple gorilla.
- c. Behind the trees appeared to stand a large building of some kind.

위의 예문들은 해결하기 위해선 아래와 같은 규칙을 받아들여야 한다.

- (50) S → PP, VP [+there] / PP

위와 같은 규칙을 받아들여야 하는 증거는 아래와 같은 문장 때문이다.

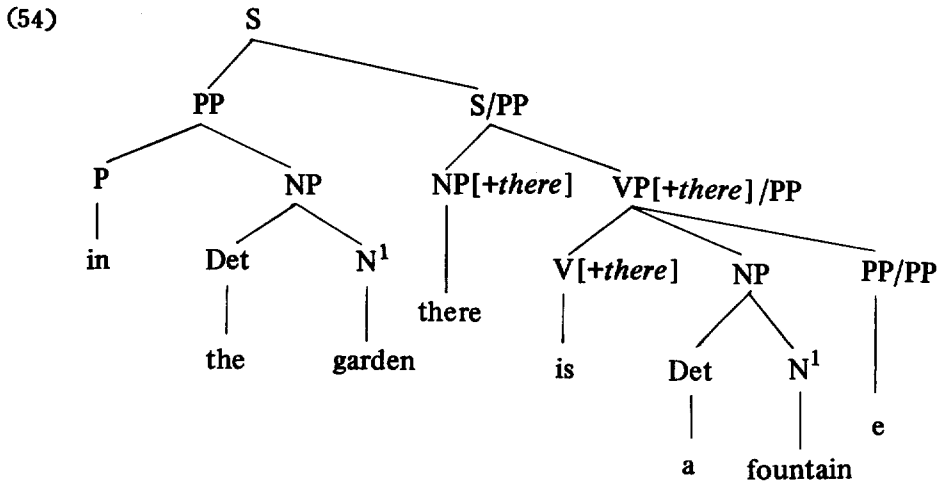
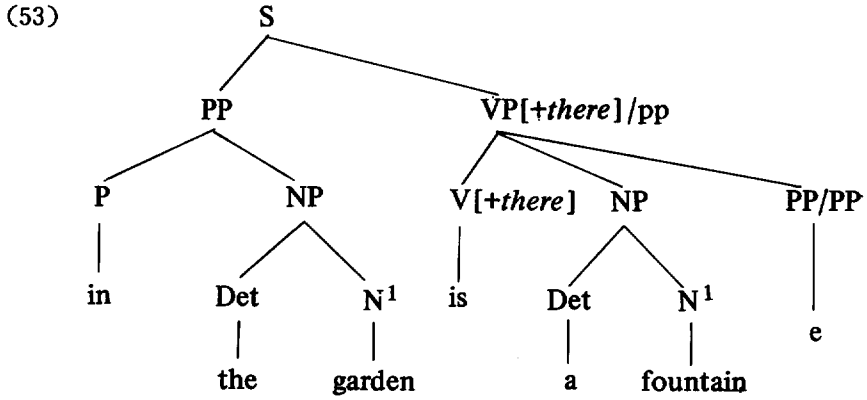
- (51) Near the fountain sat a large purple gorilla, didn't there?

즉 위의 예문은 tag question 구문으로서 (49b)의 주어는 there 이라는 것을 알려주기 때문이다. 또한 아래의 예문에서 볼 수 있듯이 Topicalization 구문의 규칙과는 상이하기 때문이다.

- (52) a. In the garden, there is a fountain.
- b. Near the fountain, there sat a large purple gorilla.
- c. Behind the trees, there appeared to stand a large building of some kind.

¹⁴ Postal, Paul (1977) *Linguistic Inquiry* 8, 141-154 참조.

위의 예문들 중에서 (49a)와 (52a)의 tree는 각각 아래의 (53)과 (54)가 된다.



위에서 중요한 점은 (49a)와 (52a)의 대응되는 문장의 의미가 동일하다는 것이다. 필자는 이러한 점에 착안하여 영어의 명령문을 GPSG로서 해결하고자 한다.

영어의 명령문은 아래와 같다.

- (55) a. Come here.
 b. Leave the room.
 c. Drink your milk.
 d. Wash your hand.

위에서처럼 영어의 명령문은 표면적인 주어가 없으며 동사는 원형이다. 그러나 아래의

예문들을 통해서 알 수 있듯이 주어인 You와 조동사 will이 생략되었다는 것을 알 수 있다.

- (56) a. Come here, won't you / *can't he?
 b. Leave the room, won't you / *won't they?
 c. Drink your milk, won't you / *must you?
 d. Wash your hand, won't you / *may I?

즉, Tag Question 구문에서는, 서술문의 조동사의 부정형 (만약 부정형이면 긍정형) 과 주어가 문미에 첨가되는데 이는 본 문장의 조동사와 주어가 무엇인지를 알게 해준다. 위에서 각 문장의 주어는 You이고 조동사는 will이라는 것을 알 수 있다.

또 아래의 예문을 통해서도 알 수 있다.

- (57) a. Kick *you / yourself.
 b. Kick *you / yourselves.
 c. Kick me / *myself.
 d. Kick him / *himself.
 e. Kick us / *ourselves.
 f. Kick them / *themselves.

즉 재귀대명사를 통해서 명령문의 주어는 You이라는 것을 알 수 있다. 왜냐하면 재귀 대명사란 앞에 나온 명사가 한 clause boundary 내에서 반복될 경우에는 재귀대명사의 형태를 취하기 때문이다. 더욱 위의 예문 가운데 정문과 비문의 차이는 아래의 예문을 통한 재귀 대명사화에서 더욱 확실하게 구별된다.

- (58) a. You kicked yourself / *you.
 b. You kicked yourselves / *you.
 c. You kicked me / *myself.
 d. You kicked him / *himself.
 e. You kicked us / *ourselves.
 f. You kicked them / *themselves.

이처럼 명령문은 통사적인 형태로서 주어는 나타나지 않지만 여러 다른 현상과 비교할 때 주어는 You이라는 것을 알 수 있다. 필자는 앞에서 보이지 않는 주어 there를 설정하여 구문을 해결하였듯이, 명령문도 그와 비슷한 방법으로 해결하고자 한다.

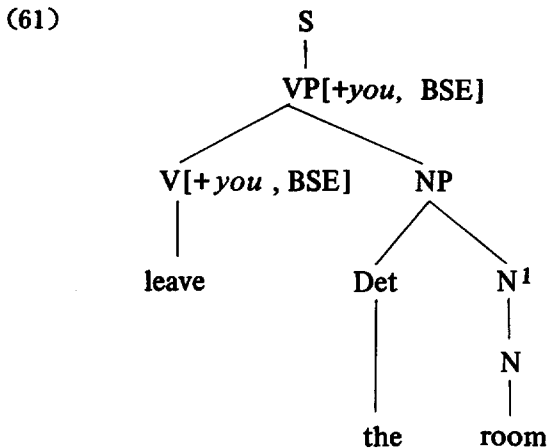
먼저 명령문에서 보이지 않는 주어를 understood 'You'라고 부르며 이 understood you를 NORM의 값으로 받아들인다. 즉 NORM의 값을 아래처럼 확장하는 것이다.

(59) { *it, there, you, NORM* }

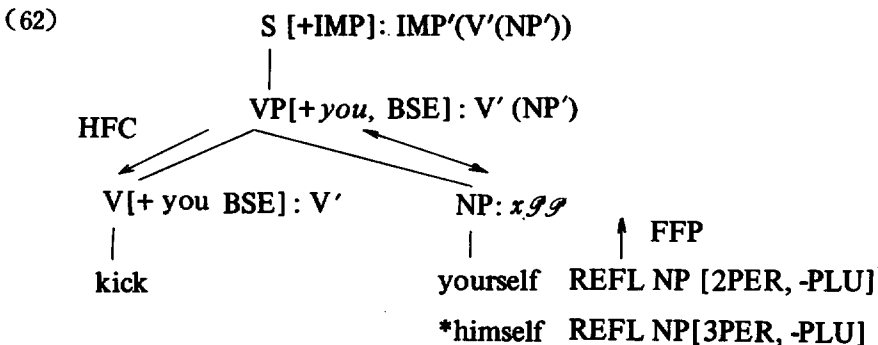
또한 아래와 같은 규칙을 설정한다.

(60) $S \rightarrow VP [+you, BSE]$ ¹⁵

이때 (55b) 구문은 아래와 같은 tree 가 된다.



또 (57a,d)의 interpreted terminated tree 는 (62)가 된다.

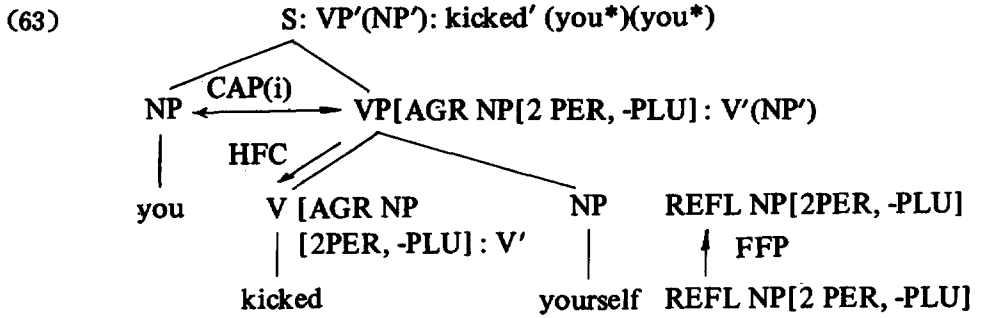


위에서 FOOT feature 인 REFL이 값으로 NP[2PER, -PLU] 일 경우에는 FFP 를 준수하며, 삼투하여 VP의 AGR의 값과 동일하여 정문이 되나, REFL의 값이 NP[3PER, -PLU] 일 경우에는 FFP에 따라 삼투되거나 VP의 AGR 값인 NP[2PER, -PLU]와 틀리게 되어 비문이 된다. 이때의 의미 해석은 Pollard 와 Sag (1983)

¹⁵ +you 는 AGR NP[NFORM you]를 대신한 것이다. 또한 명명문의 동사의 형태는 BASE 또는 INF로 취급하는 경향이 있으나 여기서는 BASE로 취급한다.

을 받아들인다. 그래서 VP의 의미는 $\lambda \nu \nu (\lambda x \text{kick}'(x^*)(x^*))$ 가 된다. 문제는 문장S의 의미이다. 그래서 필자는 IMP 라는 semantical feature 를 imperative sentence 에 받아들이고자 한다. 그리고 번역은 IMP' 이 되도록 한다.¹⁶ 이때 semantical feature란 동사 자질 중 의미 해석에서 독특한 기능을 하는 것으로 operator의 역할을 한다. 그래서 IMP는 pragmatics에서 흔히 사용하는 performative prefixes¹⁷ (또는 phrases)와 같은 'I order you to'의 의미를 갖도록 한다. 즉 speaker가 imperative mood로서 understood you에게 말하는 의미를 갖게 한다. 그러면 (62)의 번역은 IMP'(V'(NP'))로서 speaker (I)가 understood you에게 imperative mood (ORDER)로서 무슨 일을 하도록 말하는 의미가 된다.

그러나 명령문이 아닌 (58)의 구문은 다음과 같은 interpreted terminated 와 의미 해석을 갖는다.



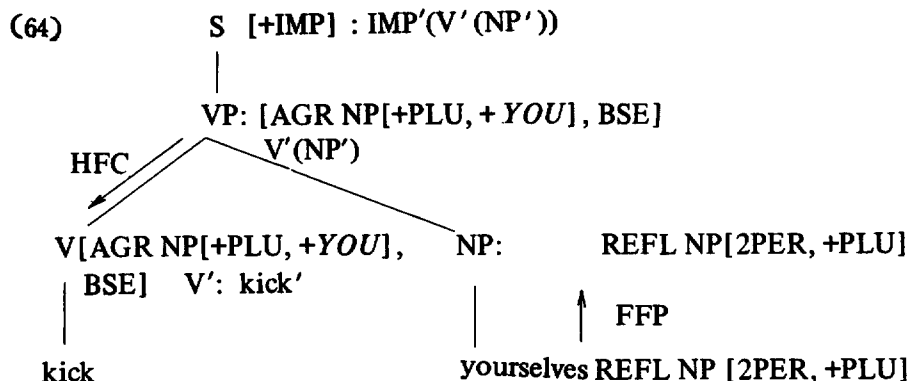
위에서의 주어 you는 NFORM의 값에서 NORM 중의 하나이다. 즉, 명령문에서의 understood you는 아니다. 그래서 의미 해석도 imperative mood가 전혀 없다.

다른 예 하나 더 들어 보자. 즉, (57b)의 tree와 의미 해석을 알아 보자.

¹⁶ IMP' = $\lambda \nu \nu VP[(I(x)-ORDER-YOU(y), \nu VP(y^*))]$ 로 생각하며, I와 You는 context-dependent function으로 어떤 발화 상황에서 누가 명령자이고 누가 피명령자인가를 찾아주는 역할을 한다. 또한 ORDER는 이 문장이 명령문임을 나타내는 것으로 가정한다.

¹⁷ Performative prefixes는 3종류가 있는데, 명령문에는 'I order you to'이, 의문문에는 'I ask you whether'이, 그리고 declarative sentence에는 'I state to you that'이다.

Levinson, Stephen C. .1983. pp. 242-51 참조.



위의 (64)에서는 HFC에 의해서 VP의 [AGR NP[+PLU, +you], BSE] 값이 V에도 전달되며, FFP에 의해서 yourselves의 REFL 값이 삼투되어 VP의 AGR 값과 일치하여 정문이 되며, 번역은 VP에서 $(\lambda \mathcal{P} \{ \lambda x \text{kick}'(x^*)(x^*) \})$ 가 된다. 그리하여 전체 문장 S의 번역은 $\text{IMP}'(\lambda x \text{kick}'(x^*)(x^*))$ 이 된다.¹⁸ 즉 speaker (I)가 understood you에게 imperative mood (ORDER)로서 understood you가 자신을 차는 개체들의 집합에 속하면 진이 되는 의미를 갖는다. 그러나 IMP'의 형식화가 문제점으로 남는다. 필자는 주에서 개인적인 제안을 하였다.

이상에서 본 바와 같이 명령구문은 표층구조에 주어 없으나 Tag Question이나 Reflexivization을 통해서 보이지 않는 주어, understood you가 있는 것 같다. 또한 명령구문의 의미는 IMP'에 의해서 나올 수 있다. 이는 종래의 performative hypothesis에 의한 performative prefixes인 'I order you to'가 명령구문에 나타난다는 것을 semantic feature인 IMP로 받아들인 것이다. 이러한 분석은 표층구조만 갖고 있는 현행 GPSG framework에 더욱 잘 맞는다.

IV. 결 론

필자는 영어 구문의 구성 성분의 구조가 현행 GPSG framework 상에서 매우 중요한 역할을 한다는 점에 착안하여, 아직은 불충분한 영역중에서 소절과 명령문에 대한 연구를 시도하였다. 특히 여러 학자들에 의해 연구되어온 소절에 대해서 현행 GPSG의 접근 방법을 모색하였는 바, 의미적인 문제를 통사적으로 해결하려는 또는 통사적인 문제를 의미적으로 해결하려는 경향이 있음을 발견하고, 각각의 문제를 각각의 분야가 맡아서 동시에 해결하도록 시도하였다. 그래서 종래의 소절에 대한 구성 성분의 구조에 반대하여, VP →

¹⁸ $\text{IMP}'(\lambda \mathcal{P} \{ \lambda x \text{kick}'(x^*)(x^*) \}) = \text{I}(x) - \text{ORDER} - \text{YOU}(y), \text{kick}'(y^*)(y^*)$

H [], NP, XP [+ PRD] 라는 ID rule 을 받아들였고, XP [+ PRD] 의 controller가 무엇인지는 동사의 의미에 의존한다고 생각하여 서로 다른 의미를 갖는 형의 동사는 SUBCAT 번호로 구별하였다. 그리고 각 문장의 semantic type 이 동사의 SUBCAT 번호에 따라 달라지도록 하여서 좀더 설명력있게 되었다. 또 구성 성분의 구조가 특이한 영어 명령문에 대해서 연구한 결과, NFORM 에 understood you 를 값으로 첨가시켜 reflexivization 현상 등을 설명할 수 있게 되었다. 또한 semantical feature IMP 를 명령문S에 도입하여 speaker (I) 가 understood you 에게 imperative mood 로서 말하는 의미를 갖도록 하였다.

이상의 분석은 표층구조만 갖고 있는 GPSG 의 정신에 부합하는 것이며 구성 성분 구조에 대한 올바른 분석이다. 또 통사적인 문제는 통사부가 해결하고 의미적인 문제는 의미부가 해결하는 올바른 언어 분석이 된다.

참 고 문 헌

- Akimajian, Adrian, and Frank Henry. 1975. *An Introduction to the Principles of Transformational Syntax*. Cambridge: MIT Press.
- Chomsky, Noam. 1981. *Lectures on Government and Binding*. Dordrecht: Foris.
- Dowty, David R. 1982b. Gramatical Relations and Montague Grammar. *The Nature of syntactic Representation*, ed. by P. Jacobson, and G. Pullum, 79-130. Dordrecht: Reidel.
- Flickinger, Daniel. 1983. Lexical Heads and Phrasal Gaps. *Proceedings of WCCFL 1*, ed. by D. Flickinger & M. Wescoat, 89-101. Stanford: Stanford Linguistics Association.
- Gazdar, Gerald. 1981. Unbounded Dependencies and Coordinate Structure, *Linguistic Inquiry 12* :155-84. Stanford: Stanford Linguistic Association.
- _____. 1982. Phrase Structure Grammar. *The Nature of Syntactic Representation*, ed. by P. Jacobson & G. Pullum, 131-86. Dordrecht: Reidel.
- Gazdar, Gerald, and Geoffrey Pullum. 1981. Subcategorization, Constituent Order, and the Notion Head. *The Scope of Lexical Rules*, ed. by M. Moortgart, H. van der Hulst, and T. Hoekstra, 107-23. Dordrecht: Foris.
- Gazdar, Gerald, Ewan Klein, Geoffrey Pullum, and Ivan Sag. 1982. Coordinate structure and unbounded dependencies. *Developments in Generalized Phrase Structure Grammar: Stanford Working Papers in Grammatical Theory*, ed. by Michael Barlow, Daniel Flickinger, and Ivan A. Sag, 38-68. Bloomington: Indiana University Linguistics Club.
- _____. 1985. *Generalized Phrase Structure Grammar*. Cambridge: Harvard University Press.
- Kaplan, Ronald M., and Joan Bresnan. 1982. *Lexical-Functional Grammar: A Formal System*

- for Grammatical Representation. *The Mental Representation of Grammatical relations*, ed. by Joan Bresnan, 173-281. Cambridge: The MIT Press.
- Livinson, Stephen C. 1983. *Pragmatics*. Cambridge University Press.
- Newmeyer, Frederick J. 1980. *Linguistic Theory in America: The First Quarter-Century of Transformational Generative Grammar*. New York: Academic Press.
- Pollard, Carl, and Ivan Sag. 1983. Reflexives and Reciprocals in English: An Alternative to the Binding Theory. *Proceedings of WCCFL 2*, ed. by M. Flickinger & M. Wescoat, 189-203. Stanford Linguistics Association.
- Postal, Paul. 1977. About a "nonargument" for raising. *Linguistic Inquiry* 8, 141-54.
- Pullum, Geoffrey K., and Gerald Gazdar. 1982. Natural languages and context free languages. *Linguistics and Philosophy* 4, 471-504.
- Sag, Ivan A., and Ewan H. Klein. 1982. The syntax and semantics of English expletive pronoun constructions. *Developments in Generalized Phrase Structure Grammar: Stanford Working Papers in Grammatical Theory, Volume 2*, ed. by M. Barlow, Daniel Flickinger, and Ivan A. Sag, 92-136. Bloomington: Indiana University Linguistics Club.
- Sells, Peter. 1984. *Syntax and Semantics of Resumptive Pronouns*. Doctoral Dissertation. University of Massachusetts.
- _____. 1985. *Lectures of Contemporary Syntactic Theories*, 77-134. Center for the Study of Language and Information, Leland Stanford Junior University.
- Soams, Scott and David M. Perlmutter. 1979. *Syntactic Argumentation and the Structures of English*. California, Berkeley and Los Angeles: University of California Press.
- Stowell, Tim. 1981. *Origins of Phrase Structure*. Doctoral Dissertation. MIT Press.
- Williams, Edwin. 1975. Small clause in English. *Syntax and Semantics* 4, 249-273.
- _____. 1980. Predication. *Linguistic Inquiry* 2, 203-238.
- _____. 1983. Against Small Clause. *Linguistic Inquiry* 14, 187-308.
- _____. 1984. There-insertion. *Linguistic Inquiry* 15, 131-54.
- 박병수. 1984. '통제 일치 원리와 한국어 존칭 어미' 「언어 연구」 Vol.5, 33-43.
경희대 언어 교육 연구소.