

자질 집합 표현에 의한 자연언어 문법 규칙 기술

박성숙 한성국
원광대학교 컴퓨터공학과

A Feature Set Description of Grammar Rules
for Natural Languages

Sung-Suk Park Sung-Kook Han
Dept. of Computer Engineering, Won Kwang University

요 약

문법체계내의 문법규칙은 규칙의 기술양식에 의해 언어특성이 결정된다. 본 논문에서는 문법 체계의 규칙기술을 위한 새로운 자질 집합 기술 (feature set description)을 제안하고, 이를 기반으로 한 파라미터화된 문맥자유문법 (parametrized context-free grammar : PCFG)을 정의하여, 자연언어의 문법규칙을 구성하는 방법에 대하여 기술한다. 자질 집합 기술은 간결한 규칙체계를 유지하면서 강력한 생성능력을 갖는 문법체계를 구현할 수 있어, 자연언어 처리 시스템에 효과적으로 적용할 수 있음을 보였다.

1. 서 론

일반적으로 자연언어처리는 형태소 분석, 구문 구조분석, 의미해석과 담화 처리등 여러단계를 거치게 된다. 이중에서 구문구조 분석(syntactic analysis)은 문장의 형성구조를 가시적의 형태로 표출하는 것으로 자연언어 처리를 위한 기반구조를 제공하게 된다. 구문구조는 자연언어의 전개양상을 정의하는 문법체계(grammar formalism)에 의해 결정되어 진다. 때문에 자연언어 처리의 많은 연구가 문법체계의 규명에 중점을 두어 진행되어 왔으며, GPSG, LFG와 HPSG등 수많은 체계가 제시되고 있다[4,8]. 제시된 문법체계들은 자연언어의 구조를 규명하기 위한 다양한 이론적 관점을 제공하고 있다.

그런데 문법체계가 자연언어 구조를 효과적으로 표현하기 위해서는 문법규칙의 기술(description of grammar rules)방식이 무엇보다도 중요해 진다[7]. 문법체계는 정의된 문법규칙을 토대로하여 구성 되기 때문에 문법규칙의 기술방식은 문법 체계의 틀을 규정하기 된다. 본 논문에서는 문법규칙체계의

규칙기술을 위해 새로운 자질집합 기술(feature set description : FSD)을 제안하고 이를 통해 파라미터화된 문맥 자유 문법(parametrized context-free grammar : PCFG)의 구성과 규칙 기술의 연산 특성에 대하여 서술한다.

본 논문에 제시한 자질집합에 기초를 둔 PCFG는 간결한 표현과 강력한 생성능력을 갖고 있으며 전개방식이 단일연산에 의해 실행됨으로 자연언어의 문법규칙기술에 유용하게 활용될 수 있다.

2. 문법규칙의 기술

Chomsky의 변형생성 문법이래로 자연언어의 구문구조를 규명하려는 지속적인 연구가 있어 왔으며, 다양한 관점에서의 많은 문법체계가 제시되고 있다.

ATN(argumented transition network)등의 순차적(procedural) 형식체계를 비롯하여, 최근에는 GPSG, LFG 와 HPSG등의 선언적(declarative) 문법체계가 널리 응용되고 있다. 일반적으로 문법체계는

이론체제와 규칙체제로 구분할 수 있다. 이론체제는 자연언어의 내재적인 특성을 일반화한 범용 원리로 구문구조의 해석과 설명에 원용되어 진다. 반면에 규칙체제는 문법규칙의 실제적인 규칙기술과 규칙의 집합체를 구문구조를 전개하는 생성시스템을 제공하게 된다. 규칙체제는 일반적으로 CFG를 형식을 중심으로 구성되어 왔다. 그러나 자연언어는 CFG보다는 다양한 구성을 하고 있기때문에 ID/LP나 복합 자질등을 도입하여 생성능력을 확장하여 왔다.

문법구조를 결정하는 규칙체제에서는 무엇보다도 규칙의 기술방식이 중요해 진다. 규칙의 기술방식에 의하여 규칙체제의 언어구조 표현의 적합성과 효율성이 결정되기 때문이다. 문법규칙의 기술방식이 자연언어 처리등에 적극적으로 활용되기 위해서는 다음의 여러 요건들을 만족할 수 있어야 한다[1].

(1) 기술의 명료성과 생성능력

문법규칙을 명료하게 기술할 수 있는 체제이어야 한다. 일반적으로 CFG 형식의 규칙 기술이 이해하기 쉽고 다루기 쉬운 장점이 있다. 그러나 CFG는 자연언어의 여러 현상을 표현할 수 만큼의 충분한 생성능력을 갖고 있지 못하다. 따라서 강력한 생성능력의 CFG형식의 명료한 기술체제가 필요해 진다. 유사문맥의존문법(midly context-sensitive grammar)인 TAG(tree adjoining grammar)가 대표적인 예이다[6].

(2) 규칙 표현성

자연언어의 다양한 구조와 문맥의존성을 효과적으로 표현할 수 있는 여러 방법이 제시되고 있다. GPSG의 ID/LP나 LFG의 자질변수(feature variable) 등은 규칙이 표현하는 의미를 보다 정확하게 제한하려는 기술방식이다. 규칙기술을 위해서는 규칙구성의 원시적 요소가 되는 문법요소들의 표현방식도 자질구조를 기반으로 표현하는 것이 일반적이며, 이 경우에 규칙전개를 위해 단일화(unification)연산을 활용할 수 있게 된다[1,7].

즉, 규칙의 표현성을 위해서는 문법요소의 자질구조와 제한된 규칙의 단일화 전개가 요구되어 진다.

(3) 처리 효율성

컴퓨터 체제를 위해서는 기술된 규칙을 파싱할 수 있는 실용적인 알고리즘이 존재하여야 한다. CFG경우 $O(N^3)$ 의 알고리즘이 존재한다[5]. 일반적으로 문법의 생성능력이 강력할수록 복잡도가 높아진다. 문법체제의 규칙체제에 대한 실용적 알고리즘은 문법체제의 실제 응용에 관건이 된다.

3. 문맥자유문법의 파라메터화

자연언어의 구문구조는 문법체제(grammar formalism)을 형식화한 규칙의 서술 양식에 의해 전개되어 진다. 문법규칙의 서술 양식은 문법체제의 이론적 원리를 명시적으로 표현하는 문법체제의 근간이다. 최근 자연언어 처리에 널리 사용되고 있는 문법체제들의 규칙 서술 양식을 고찰하여, 언어 현상을 보다 명료하게 기술할 수 있으며, 시스템 구현이 용이한 CFG기반의 자질집합기술(feature set description:FSD) 방식을 정의하고 이를 중심으로 한 문법규칙 기술에 관하여 서술한다.

3.1 문법체제의 자질 집합 기술

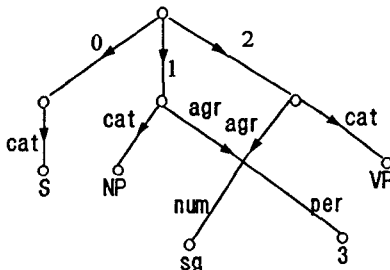
문법이란 언어의 내적 구조에 관한 지식을 기술하기 위한 지식표현 체계의 일종이다. 따라서 문법의 규칙표현은 수학적으로 잘 정의가 된(well-formed)체제를 형성하여야 하며, 언어요소와 언어 현상을 문법원리에 따라 일관성있게 설명할 수 있어야 한다.

변형문법등에서는 언어현상의 설명에 중점을 두어 문법규칙에 대한 수학적 체계형성이 미흡 하였지만, 전산언어학의 토대를 제공한 ATN의 출현으로 문법규칙을 위한 형식체제에 대한 연구에 관심을 갖게 되었다. 많은 연구의 결과로, 최근에는 주로 자연언어의 문맥 자유성(context freeness)과 선언적(declarative) 기술 방식에 기초한 자질 기반

문법체계(feature-based grammar formalism)가 일반화 되었다.

S → NP VP
 <1 agr> = <2 agr>
 VP → V NP
 <0 agr> = <1 agr>
 NP → 'he'
 <0 agr num> = 'sg'
 <0 agr per> = '3'
 NP → 'apples'
 <0 agr num> = 'pl'
 VP → 'eats'
 <0 agr num> = 'sg'
 <0 agr per> = '3'
 <0 tense> = PAST

(a)



(b)

그림 3-1. 자질 기반 문법 기술과 단일화

이런 문법체계들은 문법정보나 지식을 자질과 자질값(feature value)을 쌍으로 하는 자질 구조(feature structure)로 형식화하고, 문법 규칙의 운용에는 단일화(unification) 연산을 사용한다. 그림 3-1은 PATR 등에서 사용한 문법규칙의 표현과 단일화에 의한 구문구조 전개의 대표적인 일례를 보인 것이다.

자질 기반 문법에서 언어 정보는 레코드(record) 형태의 자질들로 구조화하고, 언어현상은 그림 3-1처럼 경로수식(path equation)을 단일화 연산으로 해결하고 있다. 자질 기반의 문법체계인 PATR,

FUG, HPSG, GPSG와 LFG등은 세부 문법 기술 양식에는 차이가 보이지만, 자질 중심의 CFG로써 적격문 생성을 위한 규칙 제한(constraint)의 부가 방식과 제한 조건의 단일화에 의한 검사를 채택하고 있다.

자질 기반 문법에서는 어휘의 정보체계가 상대적으로 중요한 역할을 하게 되며, 문법규칙은 극히 추상적인 형태로 나타난다.

(3-1) [SUBCAT <] → H[LEX -, C

HPSG의 약식 표현 (3-1)은 언어체계의 개념 구조를 추상화하고 있으나, 이를 위한 하부구조의 형성이 실제적으로는 용이하지 않으며 정의된 문법체계의 무결성(integrity)을 파악하고 보증하는 것이 어렵다. 본 논문에서는 기존의 자질 기반 문법체계를 부분적으로 포괄하고 그 이론 체계를 원용할 수 있으면서, 문법규칙의 명료한 표현과 시스템 구현이 용이한 자질 집합 기술(feature set description:FSD)을 정의한다. 이러한 규칙 기술 양식은 문법 체계의 일관성(consistency)을 검사하고 자연 언어 처리시스템 개발환경으로 적절하게 응용할 수 있다. FSD는 문법규칙의 기술을 위한 방식을 제공하며, 문법 구조체의 해석(interpretation)은 문법 이론의 원리 내에서 행하게 된다.

정의 : [FSD] FSD는 문법규칙 기술양식으로써 (T, S, F, V, P)의 5-tuple로 구성된다. 여기서 T는 F:V로 구성되는 자질항(feature term), S는 시작 기호(starting symbol), F는 자질 집합, V는 자질값을 그리고, P는 자질항으로 구성되는 $A \rightarrow \alpha$ 의 생성 규칙($A \in T, \alpha \subset T$)의 집합을 각각 의미한다.

생성규칙 P는 자질항을 하나의 비종단 기호 또는 종단기호로 간주한 CFG의 규칙체계와 동일하다. 자질항의 자질값은 집합 이론적 구조(set-theoretic

structure)가 아닌 단순값(atomic value)만을 갖는다. 자질항에서 자질과 자질값을 대표하는 자질변수(feature variable)와 자질값 변수(value variable)의 개념을 FSD에 도입한다. 이러한 변수에 대한 개념은 LFG의 ↑ 또는 ↓, CCG의 범주변수(categorial variable)등과 같은 개념으로 다른 여러 문법 체계에서 널리 이용되고 있는 개념이다. 본 논문에서는 경로 방정식처럼 문법규칙의 제한성을 부과하기 위해 변수를 사용한다. 변수의 사용은 다른 문법 기술체계의 도입없이 자질 구조 내에 표시되어지므로 규칙의 기술의 간명해지고, 그 문법적 기능도 명확하게 표현되어진다. 변수의 instantiation은 단일화 연산처럼 변수간의 동가관계(equality)로 결정한다. FSD로 그림 3-1의 규칙체계를 기술하면 다음과 같이 된다.

(3-2)

$$\left[\begin{array}{l} \text{CAT:S} \\ \text{ARG:x} \end{array} \right] \rightarrow \left[\begin{array}{l} \text{CAT:NP} \\ \text{ARG:x} \end{array} \right] \left[\begin{array}{l} \text{CAT:VP} \\ \text{ARG:x} \end{array} \right]$$

(3-2)의 FSD 표현은 언어요소를 직접 사용하여 그들간의 결합관계를 보이고 있다. 문법규칙의 기술이 언어요소의 대표기호를 사용해서 별도로 기술되는 것이 아니라, 언어요소의 구조체 안에 함께 기술되어 진다. 즉, 언어요소는 정보체계 내부에 결합가(valency)정보를 포함하게 되며, 이것은 일반적으로 변수로 실현된다.

FSD는 규칙기술 측면의 장점뿐만 아니라, 교차의존구조(cross-serial dependency)등 자연언어의 특이 구조를 표현할 수 있는 생성능력을 갖는다. 문맥의존언어(context-sensitive language)로 알려진 $L=\{a^n b^n c^n \mid n \geq 1\}$ 은 (3-3)의 FSD로 생성되어 진다.

'aaabbccc'와 같은 경우에는 SI:x의 변수값이 등가성을 갖지 않으므로 생성에서 배제된다. FSD는 이처럼 CFG형태를 유지하면서도 CSG의 생성능력을 갖기때문에 자연언어의 구문체계 기술에 유용하다.

(3-3)

$$\left[\begin{array}{l} \text{VAR:S} \\ \text{SI:x} \end{array} \right] \rightarrow \left[\begin{array}{l} \text{VAR:A} \\ \text{SI:x} \end{array} \right] \left[\begin{array}{l} \text{VAR:B} \\ \text{SI:x} \end{array} \right] \left[\begin{array}{l} \text{VAR:C} \\ \text{SI:x} \end{array} \right]$$

$$\left[\begin{array}{l} \text{VAR:A} \\ \text{SI:x} \end{array} \right] \rightarrow \left[\begin{array}{l} \text{TER:a} \\ \text{SI:x} \end{array} \right] \left[\begin{array}{l} \text{VAR:A} \\ \text{SI:x-1} \end{array} \right] \left[\begin{array}{l} \text{TER:a} \\ \text{SI:1} \end{array} \right]$$

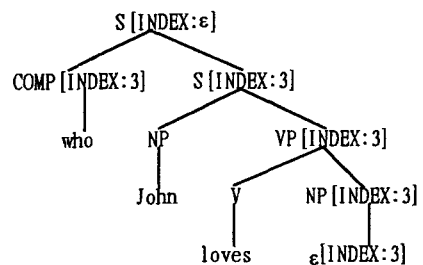
$$\left[\begin{array}{l} \text{VAR:B} \\ \text{SI:x} \end{array} \right] \rightarrow \left[\begin{array}{l} \text{TER:b} \\ \text{SI:x} \end{array} \right] \left[\begin{array}{l} \text{VAR:B} \\ \text{SI:x-1} \end{array} \right] \left[\begin{array}{l} \text{TER:a} \\ \text{SI:1} \end{array} \right]$$

$$\left[\begin{array}{l} \text{VAR:C} \\ \text{SI:x} \end{array} \right] \rightarrow \left[\begin{array}{l} \text{TER:c} \\ \text{SI:x} \end{array} \right] \left[\begin{array}{l} \text{VAR:C} \\ \text{SI:x-1} \end{array} \right] \left[\begin{array}{l} \text{TER:a} \\ \text{SI:1} \end{array} \right]$$

FSD가 자연언어의 문법체계 기술에 사용되기 위해서는 언어현상을 설명할 수 있는 장치(mechanism)가 있어야 한다. 교차의존구조등의 특이구조를 제외하고는 언어현상은 국소특성(locality)을 갖고 있다. 언어 현상에서의 국소성은 직접지배(immediate dominance)에 의해 해결이 가능하므로 CFG 기반형식으로 나타내어질 수 있다. 반면에 FSD에서 비국소적 현상은 변수의 전달(passing)이나 instantiation에 의해서 실현될 수 있다. 그림 3-2에 관형화 구문에서 변수전달에 의한 공기관계 형성과정을 보였다.

$$S[\text{INDEX:}\epsilon] \rightarrow \text{COMP}[\text{INDEX:}x] S[\text{INDEX:}x]$$

(a)



(b)

그림3-2 변수전달에 의한 공기 관계 형성

변수전달은 GPSG의 foot-feature principle이나 LFG의 f-description등과 동일한 기능을 할 수 있다.

3-2 자질 집합 기술의 자질 연산

단일화 기반 문법의 기본 연산은 단일화(unification)이다. 단일화 연산은 규칙체계의 전개를 위한 유일한 방식으로 문법체계의 간결성과 효율성을 더 해 주고 있다. FSD는 단일화 연산과 유사한 자질변수의 등가관계(equality)을 유일한 연산자로 사용하며, 단일화 연산보다도 자연언어의 다양한 현상을 포착할 수 있는 강력한 기능을 갖고 있다. FSD에서는 규칙의 적용뿐만 아니라 구문구조의 처리과정도 규칙 체계내에서 표현 되어서므로 문법규칙의 기술이 용이하고 규칙의 처리도 효과적으로 실현 되어진다. 이러한 표현능력과 처리 효율성 언어 정보의 처리를 보다 유연하게 할 수 있는 기능을 제공함으로써 자연언어의 문법체계 형성을 체계적으로 할 수 있게 한다. FSD의 등가 관계에 의한 자질연산으로 다음 7종류를 정의할 수 있다.

① 전달 (transfer)

자질값 변수를 사용함으로써 규칙내에서 자질값을 다른 문법요소로 전달할 수 있다. 전달특성은 문법 규칙 기술의 간결성을 제공해 주며, 상호 의존적인 정보를 공유할 수 있도록 하여 준다. 예를 들어 다음 (3-4)은 한국어 동사의 성분 특성을 규정하는 일반규칙으로 사용할 수 있다.

$$(3-4) \quad \left[\begin{array}{l} \text{CAT:S} \\ \text{TYPE:x} \end{array} \right] \rightarrow \left[\begin{array}{l} \text{CAT:NP} \\ \text{TYPE:x} \end{array} \right] \left[\begin{array}{l} \text{CAT:VP} \\ \text{TYPE:x} \end{array} \right]$$

어미의 특성에 따라 별도의 규칙들을 설정할 필요가 없게 되어, 규칙이 보다 일반성을 확보하게 되고 언어학적 의미도 명확하게 된다.

② 삭제 (deletion)

자질에 의해 표현되는 언어정보를 제어 할 수 있다. 규칙전개와 직접적인 연관성이 없거나

불필요한 정보를 상위 계층에서 나타나지 않도록 함으로써 처리의 효율성을 가져올 수 있다.

(3-5)

$$[\text{CAT:S}] \rightarrow \left[\begin{array}{l} \text{CAT:NP} \\ \text{ARG:x} \end{array} \right] \left[\begin{array}{l} \text{CAT:VP} \\ \text{ARG:x} \end{array} \right]$$

언어규칙과 연관되는 자질은 극히 제한되어 있다. 때문에 모든 언어정보를 규칙 전개시에 유지할 필요는 없다. 이러한 규칙에 한정적인 정보를 상위계층에서 나타나지 않도록 규칙 내에서 규정할 수 있다. (3-5)에서 ARG는 상위 계층에서는 나타나지 않는다.

③ 첨가 (addition)

규칙의 적용으로 파생될 수 있는 새로운 언어정보를 추가할 수 있다. 이러한 부분적인 정보들은 규칙 체계내에서 수용하기 아주 어렵다.

(3-6)

$$\left[\begin{array}{l} \text{CAT:NP} \\ \text{CON:SUBJ} \end{array} \right] \rightarrow \left[\begin{array}{l} \text{CAT:NP} \\ \text{TYPE:1} \end{array} \right] \left[\begin{array}{l} \text{CAT:PP} \\ \text{TYPE:1} \end{array} \right]$$

첨가는 삭제와는 달리 새로이 파생되는 언어 정보를 상위 계층에 첨가하게 된다.

④ 변경 (modification)

상위계층에서 하위계층의 자질값을 변경할 수 있다. 변경은 규칙체계의 다형성(polymorphism)을 의미하는 것으로 규칙에 유연성을 부가해 주지만, 처리시에 과도한 규칙 적용이 발생할 수도 있다. 변경의 대표적인 예는 다음과 같다.

(3-7)

$$[\text{CAT:S}] \rightarrow [\text{CAT:VP}]$$

하위 계층의 자질 CAT값이 상위 계층에서 S로 변경되어 진다.

⑤ 자질값화

자질과 자질값은 문자열 처리이다. 자질 문자열을 자질값으로 대체하거나, 자질값화는 자질 처리를 문자열 처리로 간주한 것으로 언어학적인 유의성은 없으나, 규칙기술에 융통성을 가져다 준다.

(3-8)

$$[\text{ROLE: } x] \rightarrow \left[\begin{array}{l} \text{ROLE: OBJ} \\ \text{CAT: NP} \end{array} \right] [x: \text{VP}]$$

x:VP에서 자질값 VP을 갖는 자질 x가 상위 계층에서 자질 ROLE의 자질값으로 대체된다. 상위계층에서의 자질값화는 계층구조의 표현에 적합하다.

⑥ 복사(copying)

(3-9)의 문법규칙에 대해서 살펴보자.

(3-9)

$$\left[\begin{array}{l} \text{PER: } y \\ \text{TENSE: } y \end{array} \right] \rightarrow \left[\begin{array}{l} \text{CAT: NP} \\ \text{NUM: } x \end{array} \right] \left[\begin{array}{l} \text{CAT: VP} \\ \text{NUM: } x \\ \text{PER: } y \\ \text{TENSE: } z \end{array} \right]$$

규칙이 적용 가능하면 PER:y 와 TENSE:z 의 자질 구조만이 상위계층으로 복사되어 진다. 이와 같은 연산은 자질 집합 체계에 대한 것으로 언어학적 의미를 갖는 것은 아니다. 자질 집합화에 더불어 복사는 실용적인 목적에서 사용할 수 있을 것이다. 이와 같은 경우 규칙의 간결성을 유지하기 위해 집합 자질 기호 *을 도입하여 (3-10)처럼 간략하게 기술한다. 집합 자질 기호 *는 명시적으로 표현된 자질을 제외한 전체 자질을 의미한다.

(3-10)

$$[*y] \rightarrow \left[\begin{array}{l} \text{CAT: NP} \\ \text{NUM: } x \end{array} \right] \left[\begin{array}{l} \text{CAT: VP} \\ \text{NUM: } x \\ * : y \end{array} \right]$$

집합 자질 기호 *의 도입은 FSD에 별다른 영향을 주지 않는 것으로 규칙기술의 간편성을 위해 사용하게 된다.

⑦ 선택(filtering)

집합 자질 기호를 사용하면 특정 자질값만 갖는 자질을 선택적으로 추출할 수 있다. (3-10)의 경우에 대해서 (3-11)는 해당 자질 집합중 자질값이 3인 것만 추출하게 된다.

(3-11)

$$[*3] \rightarrow \left[\begin{array}{l} \text{CAT: NP} \\ \text{NUM: } x \end{array} \right] \left[\begin{array}{l} \text{CAT: VP} \\ \text{NUM: } x \\ * : y \end{array} \right]$$

선택기능도 언어학 의미보다는 규칙기술의 실용성을 위해 사용되어 진다.

FSD는 단일화 연산과 같이 유일한 연산방식인 등가성에 기초하고 있지만, 자질변수를 도입함으로써 다양한 기능을 갖게 된다. 이러한 기능은 규칙 기술을 보다 편리하게 하여 주며, 규칙의 언어학적 의미를 분명히 하여 준다. FSD는 이처럼 규칙 기술의 명료성을 갖고 있을뿐만 아니라, 자질 정보를 제어하기 위한 자질 연산도 규칙에 동시에 기술이 됨으로 구문구조 분석 만으로도 자질 정보를 관리해 나갈수 있는 특징이 있다. FSD는 단일화 기반 문법체계보다는 일반화된 문법기술방식을 제공하고 있는 것이다.

4. 결론

자연언어의 처리 시스템등에 응용하기 위한 문법체계에서는 무엇보다도 문법규칙의 기술 양식이 중요한 요건으로 대두된다. 문법규칙은 규칙의 기술양식에 의하여 서술됨으로 규칙

체제는 전적으로 기술양식에 의존하게 된다. 보다 체계적이고 효율적인 문법규칙의 기술을 의해서는 CFG기반으로 자질중심이고, 제약(constraint)기술이 가능하며 단일화전개들을 고려해야 한다. 본 논문에서는 자질집합을 기반으로 하는 FSD표현 방식과 이를 통한 CFG의 파라미터화에 대하여 서술하였고 다양한 자질연산에 관하여 기술하였다. 본 논문에서는 제시된 자질중심의 파라미터화 방식은 표현 간결성뿐만 아니라 우수한 생성능력을 갖고 있어 문법규칙의 기술이 명료하며, 자질 정보를 제어하기 위한 자질 연산도 규칙내에 동시에 기술됨으로써 규칙처리도 효율적으로 수행된다. 특히 CFG의 파싱방법을 그대로 적용할 수 있어 언어처리에도 아주 효과적이다.

참 고 문 헌

1. 박성숙, '자질 집합 체계에 의한 자연언어의 구문구조 분석', 원광대 석사학위 논문, 1995. 2.
2. 손덕진, 최기선, 김길창, '단일화 중심 문법론에서의 단일화 방법', 인지과학 2:1, 1990
3. 이미선, 한성국, '한국어 형태소 분석기 의 정형화' 93 한국정보과학회 논문집20권 1호, 1993
4. Gazdar, G., Klein, E., Pullum, G., and Sag, L., Generalized Phrase Structure Grammar. Harvard Univ. Press, Cambridge, 1985.
5. J. Early, 'An Efficient Context-Free Parsing Algorithm', ACM 13:2, 1968
6. Joshi, A.K., 'An Introduction to Tree Adjoining Grammars', in Mathematics of Lan-

- guage(ed., Manaster-Ramer), John Benjamins, 1987.
7. Karttunen, L., 'D-PATR : a development environment for unification-based grammar', CSLI-86-61, 1986
8. Pollard, C. and I. A. Sag, Head-Driven Phrase Structure Grammar, CSLI, 1993
9. Pullum, Geoffrey and Gerald Gazdar, 'Natural Languages and Context-free Languages', Linguistics and Philosophy, 4:471-504, 1982