論 文

論文 81-18-5-3

한국어의 Machine Translation을 위한 구문 구조 분석
(Syntactic Analysis of Korean Sentence for Machine Translation)

李 柱 根*, 韓 成 國*, 全 炳 大**

(Lee, Joo Keun. Han, Sung Kook. and Jeon, Beong Tae)

要 約

이 논문은 기계번역을 위한 한국어의 構文分析 algorithm과 system 구성에 관한 것이다.

종래의 言語學的 文章構造를 재검토하여 品詞와 成分을 통일된 관점에서 형태론적으로 분석 다음, 효과적인 品詞分類 algorithm을 제안하고, 逆移動變形 algorithm을 적용한 成分構造를 attribute개념을 도입하여 phrase structure rule로 처리하였다.

또한 한국어 組合文字의 조직개념을 lexicon구성에 도입하고 breadth-first searching에 의하여 문장의 深層構造가 포함된 parsing table을 생성하는 構文分析 system을 구성하였으며, system program에 의해 입력문장을 深層構造로 분석한 결과를 보였다.

Abstract

This paper deals with the syntactic analysis algorithms of Korean sentence and system for machine translation.

The parts of speech and constituients are syntactically analized at unified view-points and then an effective classification algorithm is proposed. The constituients which are applied an inverse movement transformation algorithm are processed with the concept of attribute.

Syntactic analysis system is constructed to generate parsing table including the deep structure of sentence by lexicon proper to the combinational property of Korean and breadth-first searching method. The results obtained from the system program are shown as the parsing table of source sentences.

1. 序 論

自然語를 공학적인 측면에서 연구하기 시작한 것은 1960년 이후 컴퓨터의 발달과 더불어 기계번역(machine translation) 문제가 대두된 이후이다. 초기의 연구는 單語 單位의 변환과 약간의 例外處理만으로 족할 것으로 기대하였다.[6,7] 그러나 그 보조적인 처리가 방대해져서 system의 한계가 들어났으며, 1966년에 ALPAC 보고서가 제출되어 기계번역을 포함한 自然言語의 처리는 불가능하다고 선언되어 기계번역 연구는 한때 중단되기도 하였다. 70년대에 와서 言語理論 및 컴퓨터의 발달과 더불어 새로운 각도로 이 문제를 재조명하게[10] 되었으며, 일부는 실용화에 이르렀다.[7] 그러나 한국어의 기계번역에 대한 연구는 아직 발표된 바 없을 뿐만 아니라, 기존 文法構造도 통일되어 있지 않고 종래의 文法을 그대로 컴퓨터로 처리하기에는 많은 공학적인 문제가 파생된다.

논문에서는 컴퓨터에 의한 한국어의 構文構造分析(syntactic analysis)을 위하여 品詞의 認定과 분류를 재고찰하여 文法規則을 形式化하고, 構文分析을 위한 algorithm과 parsing 방법을 제안하였다.

*正會員, **準會員, 仁荷大學校 工科大學 電子工學科
(Dept. of Electronics Engineering, Inha Univ.)

接受日字 : 1981年 4月 28日

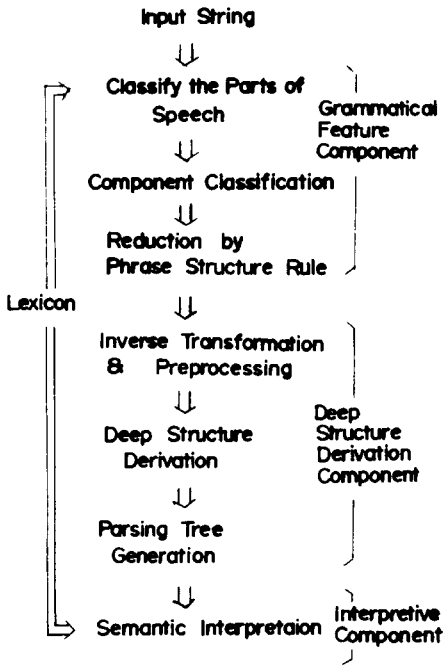


그림 1. 구문 분석 순서

Fig.1. Syntactic analysis procedures.

2. 構文分析

自然言語의 構文分析은 Chomsky의 變形生成文法에 기초를 두는 것이 일반적이며 그 순서는 그림 1과 같다.[11] 變形生成文法은 言語科學으로서 목표와 방법이 객관적이고 과학적이며[5] 또한 phrase structure rule에 의해 top-down 적으로 文章을 생성하므로 생성의 관점에서 컴퓨터화는 비교적 간단하다. 그러나 構文分析 과정은 생성의 역과정이므로 parsing이 실패될때 회복이 불가능 하는 등의 문제가 파생한다.[10]

이 논문에서는 變形生成文法으로 구성된 phrase structure rule의 처리효율을 높이기 위해 attribute의 개념을 도입하고, top-down parsing의 backtrace를 제거하기 위해 LR(K) parsing과 precedence parsing을 이용한 botton-up parsing으로 構文分析을 한다.

2—1. 品詞의 認定과 分類

단어는 문장을 형성하는 최소 의미의 단위이기때문에 단어의 분류문제는 구문 연구의 출발점이 된다. 종래에는 7품사로부터 12품사까지 10여 종류의 분류 방법이 있으며, 학자마다 상당한 견해차로 통일되어 있지 않다.[1,12] 이들 분류는 단어의 의미에 치중하고 문법적 기능을 도외시 하였기 때문에 컴퓨터와 같은 기계처리에는 그 구조가 적합하지 않다. 본 논문에서는 언어학적 유용성과 컴퓨터 처리의 효율성을 고려하여 形態論的인 관점에서 체계화하여 표 1과 같이 분류한다. 이렇게 함으로써 品詞分類가 명확해지고 成分構造에 통일된 개념이 적용될 수 있어 phrase structure rule이 간단하게 형성된다.

표 1. 품사 분류

Table 1. Classification of parts of speech.

형태분류	의미분류	
체언사 (N)	명사 (No)	
	대명사 (Ns)	인칭 (Nsm)
		지시 (Nsp)
	수사 (Nn)	
서술사 (V)	동사 (Vb)	
	형용사 (Va)	
	체언류+이다 (Vnc)	
관형사 (D)	성상 (Da)	
	수 (Dn)	
	지시 (Dp)	
부사 (A)	시간 (At)	
	양태 (Aa)	
	장소 (Ap)	
	수량 (An)	
	정도 (Ah)	
	상태 (As)	
조사 (P)	격조사 (Pc)	주격 (Pcs)
		관형격 (Pcd)
		목적어 (Pco)
		보격 (Pcc)
		부사격 (Pca)
	보조사 (Pch)	

2—2. 品詞分類 Algorithm

한글 맞춤법 통일안에 의해 띄어 쓴 구문단위를 중심으로 처리할 때 助詞와 어미의 분리를 위해 중요부(head)와 부속부(tail)로 나누어 처리하는 것이 편리하다. 중요부란 體言과 用言의 語幹처럼 의미를 결정하며 변형되지 않는 부분을 의미하고 부속부는 助詞나 어미처럼 변형하여 문법적 기능을 결정하는 부분을 의

미한다. 한국어는 서술지표가 발달하여[3] go-went-gone과 같은 구조를 갖고 있지 않기 때문에 head 와 tail의 개념을 도입하여 "먹고, 먹지, 먹을…"등을 효과적으로 처리할 수 있다.

품사분류는 성분을 결정하기 위한 전단계이므로 성분결정을 위한 정보를 추출할 필요가 있다. 추출할 정보는 조사와 어미정보가 있으며, 조사정보는 표 2와 같이 하고 어미정보는 서술사의 성격을 고려하여 다음과 같이 세부분류 한다.

1) 종지활용 (Vse_1)
   - 평서형 (Vse_{11})
   - 의문형 (Vse_{12})
   - 감탄형 (Vse_{13})
   - 명령형 (Vse_{14})

2) 전성활용 (Vse_2)
   - 체언사형 (Vse_{21})
   - 관형사형 (Vse_{22})
   - 부사형 (Vse_{23})

3) 접속활용 (Vse_3)
   - 대등형 (Vse_{31})
   - 종속형 (Vse_{32})

3항의 접속활용의 분류는 여러가지 설이 있으나, 본 논문은 대등형과 종속형 분류방법(2)을 도입한다.

이상의 고찰을 기초로 品詞分類 algorithm class를 다음과 같이 정의한다.

algorithm [class] P(M)은 조사가 기억된 memory 영역을 의미하며, E(M)은 어미가 기억된 memory 영역이다.

c_1 : 最長一致 원칙을 사용하여 searching 과정에서 일치한 PTR을 모두 stack에 push한다 (참고 3—1)

c_2 : Top(stack)을 pop하여 입력단어를 head와 tail로 분리한다.

c_3 : Tail이 존재하면 PTR에 의해 품사를 찾고, 존재하지 않으면 step c_{12}를 실시한다.

c_4 : N을 포함한 경우 短音節 test를 한다.

c_5 : Test에 성공하면 P(M)을 tail을 가지고 searching하여 조사를 찾고, 실패하면 step c_7을 실시한다.

c_6 : 조사가 존재하면 조사를 분류하여 table을 작성한다.

c_7 : 조사가 존재하지 않으면 tail을 E(M)으로 searching 한다.

c_8 : 어미가 존재하면 어미를 분류하여 table를 작성한다.

c_9 : 어미가 존재하지 않으면 step c_2 부터 다시 한다.

c_{10} : PTR에 의해 V가 포함되어 있으면 tail을 가지고 E(M)을 searching하고, 포함되어 있지 않으면 다음 단어를 처리한다.

c_{11} : 어미가 존재하면 table을 작성하고, 어미가 존재하지 않으면 다음 단어 처리를 한다.

c_{12} : PTR에 의해지시된 품사가 D 또는 A만인 경우, table을 작성한다.

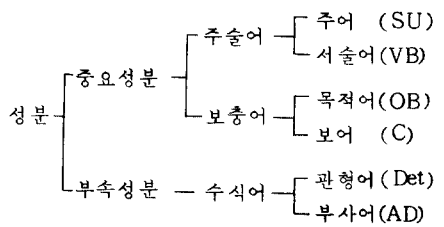
c_{13} : N또는 V가 포함된 경우 (Det,Nt) 또는 (Det, Vt)로 각각 판정한다. (참고 2—4)

2—3. 成分決定

문장을 形態論的인 관점에서 고찰하면 성분은 문법형식의 構成要素인 동시에 사고내용의 단위가 된다. 일반적으로 한국어는 그 특징상 성분을 중심으로 構文分析하는 것이 가장 효율적이라고 지적되어 있다.[4] 본 논문에서는 한국어의 본질적인 구성체계를 다루기 위하여 종래의 構文要素를 정비하고 문법적 작용을 중시하여 표 2와 같이 분류한다.

표 2. 성분 분류

Table 2. Constituient classification



위 보어(C)의 정의는 학자마다 다르고 形態論的으로는 주여와 같은 형식을 취하므로 補語와 主語를 기계적으로 선별하는 것은 매우 곤란하다. 이 논문에서는 논문(13)을 참고로 하다.

品詞構造는 成分構造와 통일된 개념을 적용하여 컴퓨터처리가 효과적이다.

품사분석 과정에서 유도한 정보에 의해 표3과 같은 phrase construction rule에 의해 성분을 결정한다.

품사분류에서 추출한 정보만을 사용하여도 성분이 결정되므로 효과적인 처리를 할 수 있고, 전체 system을 통일적으로 구성할 수 있다.

표 3. 구성 규칙
Table 3. Phrase construction rule.

SU ⟵ Ni + Pcs
OB ⟵ Ni + Pco
C ⟵ Ni + Pcc
Ni ⟵ Ni | Vs + Vse21
Ad ⟵ A | Ni + Pca | Vs + Vse23
Det ⟵ D | Ni + Pcd | Vs + Vse22
VB ⟵ Vs + Vse1

2-4. 逆移動變形 Algorithm

한국어에서는 어순이 일정하지 않기 때문에 변형된 表面構造(surface structure)로 부터 深層構造(deep structure)를 유도하고 修飾語의 非交叉性을 유지하기 위해서 역변형을 해야한다. 이동변형을 대상으로 逆移動變形을 위해 다음을 정의한다.

정의 1.[Attribute] ; 품사분류 과정에서 결정된 성분의 성격을 attribute라 한다.

$A = \{ att \mid att \in org, Nt, Vt \}$

여기서 org ; 품사분류 lexicon에 정의된 성분과 같은 경우
Nt ; Vse21인 경우
Vt ; Vse22 또는 Vse23인 경우

정의 2.[성분구조] : 성분구조는 ordered pair로 다음과 같이 정의한다.

$Pharce = \{ (ph, att) \mid ph \in Su, OB, C, Det, Ad, VB, att \in A \}$

정의 3.[성분관계]

$R(phrase_1, phrase_2) = phrase_2$

단, i) $ph_1 = Det$이면 ph_2 Su, OB, C 또는 $phrase_2$(Vb, Nt), (Ad, Nt)
ii) $phrase_1 \in$ (Ad, org), (Ad, Vt)이면 $ph_2 \in$ Vb 또는 $phrase_2 \in$ (Det, Vt)
iii) $ph_1 \in$ad이면 $phrase_2 \in$ (Su, Vt)
iv) $phrase_1 \in$ (Ad, org)이며 $phrase_2$ (Ad, Vt)일 경우에만 성립한다.

이상은 본 논문에서 품사분류를 성분 구조에 통일된 개념으로 설정하였기 때문에 가능하다.

정의 3을 이용하면 單一文의 逆移動變形은 용이하나, 復合文의 경우는 먼저 主述관계를 명확히 해야 한다. 그러므로 정의 2를 이용하여 主述관계를 결정한 다음 逆移動變形을 한다. 복합문에 대한 逆移動變形 algorithm Move는 다음과 같다.

Algorithm [**move**] Si는 가장 안쪽에 위치한 主語(the inmost subject)이고 So는 가장 밖에 위치한 主語(the outermost subject)이다. 主述관계가 성립한 서술사는 다음 처리에 영향을 받지 않는다.

M_1 : Si를 찾는다.
M_2 : Si 다음에 (Ob, Vt) 또는 (C, Vt)가 존재하면 주술관계를 성립시키고 step M_5를 처리한다.
M_3 : att가 Vt인 것을 찾아 주술관계를 성립시키고 step M_5를 처리한다.
M_4 : Vb와 주술관계를 성립시킨다.
M_5 : 모든 주어에 대하여 주술관계가 성립되었으면 step M_6을 하고, 성립되어 있지 않으면 Si_{-1}에 대해 step M_2부터 실행한다.
M_6 : VB가 주술관계를 성립시켰으면, step M_7을 실행하고 아니면 So의 주술관계를 끊고 step M_2 부터 실행한다.
M_7 : 주어앞에 있는 Ob 또는 C는 phrase realation을 적용하여 주술관계 사이로 이동한다.
M_8 : 주술관계가 성립한 가장 안쪽부터 정의 3의 성분관계를 적용한다.
M_9 : 단일문의 역이동 변형규칙을 적용한다.

2-5. Phrase structure rule

逆移動變形이된 구조에서 성분의 문법적 관계를 규명하고, 문장의 深層構造를 유도하기 위해 표 4와 같은 phrase structure rule을 정의한다.

표 4. 구조규칙
Table 4. Phrase structure rule.

Su ⟵ (Det) + SU
OB ⟵ (Det) + OP
C ⟵ (Det) + C
VB ⟵ (Ad) + VB
NP ⟵ SU | OB | C
VP ⟵ VB
VP ⟵ NP(OB,C) + VP
S ⟵ (NP) + VP

이와 같이 phrase structure rule이 간단명료하게 유도된 이유는 품사분류를 성분구조에 형태론적으로 일치시키고, attribute의 개념을 도입함으로써 성분결정이 용이해졌기 때문이다.

3. 構文分析 System

일반적으로 構文分析 system에 지향하는 방향은 (i) 문법표현의 이해가 쉽고 (ii) 문법규칙을 적용하는 제어구조에 유연성이 있어야 하며, (iii) 효율적이어야 한다는데 있다.

본 논문에서는 한국어의 lexicon 구성과 앞서 제시한 algorithm에 의하여 유연성 있는 構文分析 system을 구성하였다.

3—1. Lexicon 구성과 Searching 방법

自然言語의 대형사전을 그대로 기억시킨다는 것은 어려운 문제의 하나이며, 또 기억된 어휘를 효과적으로 처리하는 것은 더욱 어려운 일이다.

본 논문에서는 앞서 제시한 5품사에 의해 단어를 분류하고 變形하지 않는 중요부 head만을 기억시킨다. 어휘의 data format는 그림 2와 같다.

A는 기본자모이고, B는 동일계열의 다음 pointer,

A	B	C	PTR

그림 2. Lexicon의 data format

Fig. 2. Data format for lexicon.

C는 하위계열의 pointer로서 작용한다. PTR은 단어의 정보에 대한 pointer로 사용한다. pointer로서 각 어휘를 연결시키므로써 삭제, 첨가와 수정이 자유로우며 체계에 얽매이지 않고 처리할 수 있다.

lexicon으로부터 어휘를 효과적으로 searching하기 위해 한국어의 組合文字 개념을 활용한다.[8] 한국어 단음절의 group 변환 특징은 breadth—first searching에 적합한 구조이다. 또한 data format이 breadth-first searching에 적합하므로 searching 할 수 있다. head와 tail을 분리하기 위해 最長一致의 원칙 (the principle of the longest matching)[9] 을 적용하여 searching 과정에서 일치된 pointer를 stack operation에 의해 하나씩 타당성을 고찰한다.

(참고:그림 3)

3—2. 구문분석순서

입력문장을 컴퓨터에 의하여 構成을 分析하는 순서는 그림 4와 같다. 構文分析은 앞서 언급한 각 algorithm과 phrase structure rule등에 의해 처리한다.

辭典分析(lexical analysis)에 의한 symbol table은 다음과 같은 구조로 작성된다.

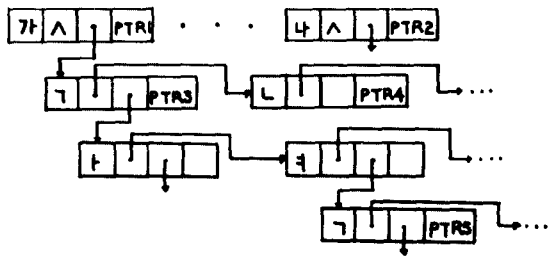


그림 3. Lexicon 구성과 searching 방법

Fig. 3. Lexicon and searching method.

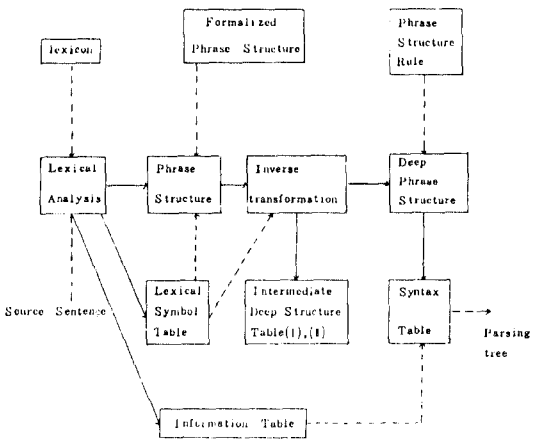


그림 4. 구문분석 흐름도

Fig. 4. Flow diagram of syntactic analysis.

index	source phrase	pos	par	end	phs	atts

index는 table의 일련번호, source phrase는 입력문장의 성분, pos는 품사분류이고, par는 조사의 세부분류이며 end는 서술지표의 세부분류이다. Phs와 atts는 phrase construction rule에 의해 phrase structure 단계에서 채워진다.

역변형(inverse transform) 단계에서 입력문이 單一文일 경우에는 intermediate deep structure table (I)을, 復合文일 경우에는 table(II)를 구성한다.

No	front index	back index	ph	att

위 table (I)의 구조에서 No는 일련번호 이고 front index와 back index는 성분관계R이 발생한 전후 성분의 index이며, ph와 att를 결정한다. 복합문

일 경우에는 역이동변형 algorithm에 의해 table (Ⅲ)를 다음과 같이 작성한다.

table(Ⅲ) ;

No	index

문장은 최종적으로 深層構造 단계에서 구문분석을하며, syntax table 또는 parsing tree를 작성한다. 본 논문은 두 성분간의 관계를 중심으로 하고 있기 때문에 parsing tree는 본질적으로 binary tree가 된다. parsing tree를 위한 syntax table의 구조는 다음과 같다.

Line	left	right	node	att

Line은 syntax table의 일련번호이고 left와 right는 phrase structure rule이 적용되어 parsing tree의 node를 구성할 수 있는 intermediate deep structure table 또는 syntax table의 일련번호이다.

3-3. Simulation 결과

앞서 제시한 algorithm과 構文分析 순서에 따라 program을 작성하여 入力文을 深層構造가 포함된 syntax table로 분석된 결과를 표 5, 6에 보인다. 이밖에 가능한 복잡한 문장에 대해서도 simulation한 결과 만족한 결과를 얻었다.

표 5. 단문의 예에 대한 parsing tree
Table 5. Parsing table for a simple sentence example.

LEXICAL SYMBOL TABLE

INDEX	POS	PAR	END	PHS	ATTS
I1	A	--	--	AD	ORG
I2	V	--	VSE22	DET	VT
I3	N	PCO	--	OB	ORG
I4	D	--	--	DET	ORG
I5	N	PCS	--	SU	ORG
I6	A	--	--	AD	ORG
I7	V	--	VSE11	VB	ORG

INTERMEDIATE DEEP STRUCTURE TABLE(1)

NO	FRONT	BACK	PH	ATT
N1	I4	I5	SU	ORG
N2	I1	I2	DET	ORG
N3	N2	I3	OB	ORG
N4	I6	I7	VB	ORG

SYNTAX TABLE

LINE	LEFT	RIGHT	NODE	ATT
L1	--	N1	NP	--
L2	--	N3	NP	OB
L3	--	N4	VP	--
L4	L2	L3	VP	--
L5	L1	L5	S	--

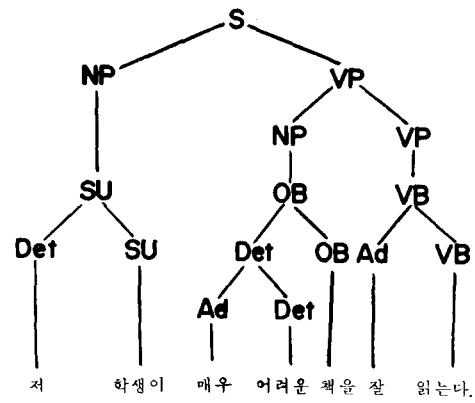


그림 5. 표 5의 parsing tree
Fig. 5. Parsing tree representation of Table 5.

(1) 單文의 경우

저 학생이 매우 어려운 책을 잘 읽는다.

이 예의 syntax table을 parsing tree로 표시하면 그림 5와 같다.

표 6. 복합문의 예에 대한 parsing table
Table 6. Parsing table for a complex sentence example.

LEXICAL SYMBOL TABLE

INDEX	POS	PAR	END	PHS	ATT
I1	N	PCS	--	SU	ORG
I2	N	PCS	--	SU	ORG
I3	A	--	--	AD	ORG
I4	V	--	VSE22	DET	VT
I5	N	PCC	--	OB	ORG
I6	V	--	VSE11	VB	ORG

INTERMEDIATE DEEP STRUCTURE TABLE

NO	INDEX
N1	I1
N2	I2
N3	I3
N4	I4
N5	I5
N6	I6

SYNTAX TABLE

LINE	LEFT	RIGHT	NODE	ATT
L1	--	N1	NP	--
L2	--	N2	NP	--
L3	N3	N4	VP	--
L4	--	N5	NP	OB
L5	--	N6	VP	--
L6	L2	L3	S	DET
L7	L6	L4	NP	OB
L8	L7	L5	VP	--
L9	L1	L8	S	--

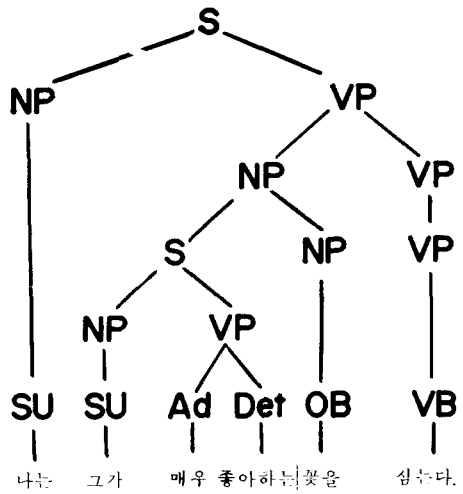


그림 6. 표 6의 parsing tree

Fig.6. Parsing tree representation of Table 6.

(2) 復合文의 경우

나는 그가 매우 좋아하는 꽃을 심는다.

이 예의 syntax table을 parsing table로 나타내면 그림 6과 같다.

4. 結論

自然言語의 構文分析문제는 광범위하고 대단히 어려운 문제에 속한다. 한국어의 경우는 文法構造가 다양하고 통일되어 있지 않으므로 더욱 그러하다. 본 논문은 한국어 文章의 成分간의 관계와 文法規則을 形式化하고 이것을 기초로 문장의 構文分析 algorithm과 분석 system을 보였다. 검토된 중요한 성과를 총괄하면 다음과 같다.

(1) 品詞와 成分構造를 통일된 개념으로 유도함으로써 phrase structure rule이 효과적으로 이루어져 parsing이 간결하게 이루어졌다. 또 분석 system의 체통은 變形生成文法의 구조와 일치한다.

(2) 品詞分類를 컴퓨터처리에 알맞게 형태론적으로 구성하였고 분류 algorithm을 제시하였다.

(3) 한국어 문장의 결합관계를 재고찰하고 成分관계 R과 attribute의 개념을 도입하여 構文分析을 유연성 있게 하였다.

(4) 構文分析이 단계별로 이루어져 program 처리가 용이하다. 그러나 曖昧文章(ambiguous sentence)은 포함치 않고, 또 한국어의 기계번역을 위한 최초의 시도이므로 매우광범위에 걸쳐 언급하였다. 부분적인 깊은 부분은 추가로 발표하겠으며, 이 분야에 관심을 가진 독자들에게 한국어에 대한 전체파악에 참고되면 다행이겠다.

參 考 文 獻

1. 최현배, "우리말본", 정음사, 1962.
2. 남광우, "개정현대국어국자의 제문제",일조각, 1973, PP.83－105.
3. 김민수, "국어문법론", 일조각, 1977.
4. 유목상, "국어의 문장구성 단위에 대한 고찰", 중대논문집 제 14집, PP.39－63.
5. 서정수, "변형 생성문법의 이론과 국어 V-류어의 하위분류", 아한 1－1, 1968 PP. 57－110.
6. A. Hill, "Linguistics," Voice of America Forum Lectures, 1969, pp. 213-234.
7. Victor Ynguve, "COMIT" CACM Vol. 6, No. 3, 1963. pp. 83-84.
8. J.K. Lee, "A Method for the Recognition of Printed Korean Characters" J. KIEE, Vol. 7, No. 4, pp. 198-209.
9. D. Knuth, "The art of Computer Programming; Vol. 3 Sorting and Searching," Addison-Wesley, 1973.
10. T. Winograd, "Understanding Natural Language," Academic Press, 1972.
11. N. Chowsky, "Aspect of the theory of Syntax," 1965, MIT Press.
12. W.A. Woods, "Transitional Network Grammar for Natural Language Analysis." CACM Vol. 13, No. 10, 1970.
13. 서병국, "국어문법논고", 형성출판사, 1973.
14. 정인승, "우리말의 씨가름에 대하여",한글, Vol.125, PP. 32－43.