GPSG를 이용한 구문 해석기의 설계에 관한 연구

°우 요섭* 김 영섭* 김 한우* 최 병욱*
*한양 대학교

A STUDY ON THE CONSTRUCTION OF NATURAL LANGUAGE PARSER USING GPSG

Y. S. WOO* Y. S. KIM* H. W. KIM* B. U. CHOI*
*HANYANG UNIVERSITY

ABSTRACT

This paper designs parser using GPSG for syntactic and semantic analysis of English input sentences. By use of a number of unification-based principles and Tomita's algorithm, syntactic anaysis is described. Also in semantic analysis, Montague semantics is used.

1. 서론

자연 언어 처리를 위한 formalism으로 unification-based approach가 많이 제안되고 있다. GPSG(Generalized Phrase Structure Grammar), LFG와 같은 theory-oriented formalism과 FUG, DCG, PATR-II등의 tool-oriented formalism이 그러한 것들이다.

GPSG는 1980년 전후에 Gerald Gazdar에 의해 제안된 언어학과 논리학을 결합시킨 문법 이론으로 아직 상당한 이론적 유동성을 내포하고 있으나, 현상의 이론적 성과로도 계산기 공학적 측면, 즉 implement 영역에서는 많은 주목을 받고 있다고 보여진다.

GPSG의 개략적 모델은 Fig1.과 같다. 입력문은 구구조 문법에 의해 구문 해석을 하고, 다시 변환 규칙에 의해 의미 표현을 하며, 의미 표현에 관해서는 Montague semantics를 기반으로 하는 논리식으로 표현한다.

GPSG의 언어학상의 가장 큰 특징은, Chomsky의 변형문법에서와 같이 변형 조작에 의해 언어 현상을 설명하지 않고, CFG를 기초로 하여 문의 구문적인 해석을 행한다는 것이다.

또한 근래의 변형문법이나 구구조 문법에서는 rule의 수가 줄어드는 추세인데, 이것은 rule을 대신하여 principle을 중시하기 때문이다. GPSG는 그 대표적으로 ID/LP rule, Metarule, principles, conventions등을 사용한다는 점에서 Metagrammar라 불린다.

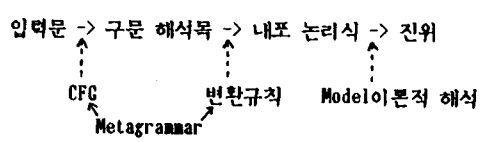


Fig1. GPSG의 개략적 모델

본 논문에서 ID rule의 처리에는 Tomita's algorithm을 도입, modify하여 효율성을 제고하였고, 입력문의 의미 기술은 syntactic, semantic feature들과 operator들을 사용하는 modified Montague grammar로서 Semantic Interpretation Schema를 채용하여 표현하였다.

2. GPSG에 관한 고찰

2.1 구문 해석

GPSG에서는 문의 각 categories에 관한 information을 feature를 통해 표시한다. feature는 단어의 형태소 정보(품사, 수, 시제, 인칭등)를 구로 확장한 것으로, 구의 품사, 의미 category, 수, 시제, 격, 인칭, 생략된 내부 구의 문법범주, 관계대명사, 재귀대명사등에 관한 정보의 집합체이고, principles들의 control에 따라 parsing tree의 nodes 사이를 전이하면서 information을 전달한다. 다음에 GPSG에서 사용되는 feature들의 일례를 보인다.

(1) CAT = {CASE, COMP, CONJ, GER, NEG, NFORM, NULL, POSS, REMOR, WHMOR}
HEAD = {ARG, ADV, AUX, BAR, INV, LOC, N, PAST, PER, PFORM, PLU, PRD, SLASH SUBCAT, SUBJ, V, VFORM}
FOOT = {RE, SLASH, WH}

feature의 표현은 [feature-name value]의 형태로 하는데, value는 또다른 feature의 표현을 가질 수 있다. feature AGR, SLASH, RE, WH는 CAT으로 구분된 feature를 그 value로 갖으며, 상기의 feature는 각기 그 구분에 따라 대응하는 principles의 적용을 받게 된다.

GPSG에서 사용되는 rule은 단순히 PS rule이 아니라 ID/LP rule이다. ID(immediate dominance) rule은 PS rule에서의 RHS의 각 category의 존재 여부만을 표시하고, LP(linear precedence) rule은 ID rule의 RHS의 각 category의 순서를 정해준다. GPSG는 N[+N, -V], V[-N, +V], A[+N, +V], P[-N, -V]의 4개의 major category와 Det(determiner)등의 몇 가지 minor category를 가진다.

한편 basic ID rule set로 부터 새로운 ID rule set를 얻는 function으로 Metarule을 도입한다. subject-aux inversion, passive voice, extraposition과 같은 언어 현상은 이러한 Metarule에 의해 표현되고 있고, 이러한 점에서는 Metarule이 변형문법의 변형조작과 유사하나, Metarule의 적용에 의해 최종적으로 ID rule의 형태를 얻는다는 점에서 차이가 있다. (2)는 Metarule의 일례이다.

(2) Passive Metarule

VP -> W, NP

⇓

VP[PAS] -> W, (PP[PFORM by])

feature [PAS]는 [VFORM PAS]의 약어로 passive를, meta-variable W는 임의의 문법 category들을 말한다.

Metagrammar를 이루는 또다른 요소가 conventions과 principles들이다. GPSG에서는 FCR, FSD의 convention과, 분 해석 tree의 각 node 사이의 feature 전파에 관한 HFP, FFP, CAP의 principle들을 사용한다.

FCR(Feature Cooccurrence Restrictions)은 [-low] -> [+high]와 같이 feature들 사이의 어떠한 dependencies를 표시하는 absolute condition으로 'category의 legal extention이라 할 수 있다. FSD(Feature Specification Defaults)는 specified되지 않은 feature들에 대한, 또 feature들 간에 언급되지 않은 관계들에 대한 default를 주는 convention이다.

그리고 HFP은 LHS의 HEAD feature(PAS 등)의 value는 RHS에서 H(LHS와 같은 [N, V]를 갖는 syntactic category)의 대응하는 HEAD feature의 value와 일치한다는 것을, FFP는 LHS의 FOOT feature(SLASH 등)의 value는 RHS에서는 각 category의 대응하는 FOOT feature의 value를 합한 것과 일치한다는 것을, CAP는 주어와 동사의 수.성 일치등과 같이 RHS category들 간에 control관계가 있을 때 control feature들의 일치를 표시한다.

이상의 ID/LP rule, Metarule, conventions, principles 들에 의한 GPSG의 구조를 Fig2에 보였다. 이 그림에서와 같이 basic ID rule set로 부터 instantiate된 rule set는 PS rule set와 같게 된다.

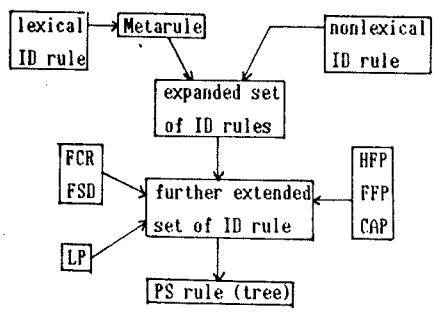


Fig2. GPSG의 구조

2.2 의미 해석

GPSG의 의미론은 Montague semantics와 거의 유사하다고 할 수 있다. 그러나 동사가 rasing-to-object verb, equi verb등으로 쓰일때 의미기술에 각기 f_R, f_E의 operator를 사용하며, passive verbs는 fp의 operator를 사용한다. f_R, f_E, fp를 interpretation하기 위해서는 이하 (4)와 같은 Meaning Postulate를 사용한다.

(3) a. loved => fp(love')

b. believed => fp(believe')

c. believed => fp(f_R(believe'))

d. persuaded => fp(f_E(persuade'))

(4) Meaning Postulate

a. $\forall v \forall P_1 \cdots \forall Pn \square[f_R(\zeta)(v)(P_1)\cdots(Pn) \leftrightarrow \zeta(v(P_1))\cdots(Pn)]$

b. $\forall v \forall P_1 \cdots \forall Pn \square[f_E(\zeta)(v)(P_1)\cdots(Pn) \leftrightarrow P_1\{\lambda x[\zeta(v(x^*))(x^*)(P_2)\cdots(Pn)]\}]$

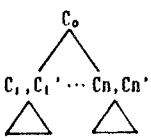
c. $\forall P_1 \forall P_2 \forall v_1 \cdots \forall vn \square[fp(\zeta)(P_1)(v_1)\cdots(vn)(P_2) \leftrightarrow \zeta(v_1)\cdots(vn)(P_2)(P_1)]$

(단, P_1, P_2는 NP-type)

이외에도 postnominal modifier일때는 MOD라는 semantic feature, topicalization일때는 EXT라는 operator를 정의하여 의미를 기술한다.

실제로 semantic interpretation에서, 입력된 syntactic parsing tree로부터 logical semantic forms을 얻어내는 schema는 다음의 Semantic Interpretation Schema이다.

(5) Semantic Interpretation Schema



$C_0'=Tr_0(Tr_1(FR((C_0\text{-FOOT}), (Tr_2(C_1'),\cdots Tr_2(Cn'))m)))$

where

a. if $\langle f,\alpha\rangle \in C_0$인 어떤 FOOT feature f가 있으면
then $Tr_0(\phi)=\lambda v^{\alpha}\phi$
else $Tr_0(\phi)=\phi$

b. if [CONJ α]가 C_0에는 없고 daughter node에 있으면
then $Tr_1(\phi)=\alpha'(\phi)$
else $Tr_1(\phi)=\phi$

c. if $\langle f,\alpha\rangle \in C_0$인 어떤 FOOT feature가 있으면
then $Tr_2(Ci')=Tr_3(Ci')(v^{\alpha})$ or $Tr_2(Ci')=Tr_3(Ci')$
else $Tr_2(Ci')=Tr_3(Ci')$

d. if feature σ가 C_0에는 없고 daughter node에 있으면
then $Tr_3(Ci')=\sigma'(Ci')$
else $Tr_3(Ci')=Ci'$

(5)에서 FR은 daughter node의 semantic representation들을 combination하는 operator인 Functional Realization을, $\langle f,\alpha\rangle$는 feature name f와 value α의 pair를 의미한다.

3. GPSG Parser의 설계

3.1 ID rule의 적용

GPGS parser를 위한 형태소 해석은 그 결과를 모두 feature로서 lexical item에 부여한다. 즉 syntactic category와, VFORM, PLU, PAST, WH등의 feature가 결정되어 진다.

ID rule의 표현에 있어서는 생략이 가능한 category와 반복이 가능한 category를 활용함으로서 ID rule의 기술을 좀 더 generalize하게 하였다. 즉, 생략 가능한 category에는 *를, 반복이 가능한 category에 @를 붙여 표기하여 일반화하고, rule에 부가된 feature들을 적정하게 줄임으로서 ID rule의 기술을 간략하게 한다. 실제 본 논문에서 구성한 시스템은 40 개 정도의 ID rule만을 갖는다. 한편, category의 구분에 있어서는 N, V, A, P의 major category 이외에 Det(determiner), Deg(degree modifier), Cnj(conjunction), Cmp(complementizer)의 minor category만을 사용하였다 (6)에 ID rule의 일례를 보인다.

(6)
```
(VP -> H ((SUBCAT 10)) PP* ((PFORM TO)) VP ((VFORM INF)))
(N2 -> DET* ((SUBCAT 23)) H1)
(A1 -> H ((SUBCAT 38)) V2 ((VFORM INF) (SLASH NP)))
```

ID rule의 처리 과정, 즉 interpreter는 Tomita's algorithm을 근간으로 구현한다. Tomita's algorithm은 stack을 direct acyclic graph로 기술한 Graph-Structured Stack을 이용하여 parsing을 수행한다. 또한 parsing 과정에서 rule 적용에 ambiguity가 있을때는 가능한 parse tree들을 병렬적으로 모두 산출하며, 이때 efficient한 representation을 제공하기 위해 sub-tree sharing technique과 local ambiguity packing technique을 채용하고 있다.

Tomita's algorithm은 LR table과 동일한 SHift, REduce, ACCept, ERRor를 entry를 갖는 action table과, goto table을 가지고 parsing을 수행한다. 그러므로 시스템에서 ID rule은 각 state에 대응하는 table 상의 multiple entry를 구성한다. rule 적용시에 ambiguity가 발생하면 table상의 적용 entry에 해당하는 ID rule 상의 feature를 이용하여 ambiguity를 갖는 parsing 과정을 제어한다.

3.2 feature instantiation

syntactic parsing에 있어서는 principle들과 convention들의 적용 순서가 중요하게 된다. 이것은 이들 모두가 feature에 대한 규제 또는 변환을 의미하기 때문이다. 본 논문에서는 (7)의 순서로 feature instantiation을 행하였다.

(7) FSD < FCR < Metarule < LP < CAP < FFP < HFP

goto table과 action table은 FCR과 LP 사이에서 참조하게 되며, LP 적용 후에 rule에 기술된 feature의 test를 하고 이로써 parsing partial tree들을 완성해 나간다.

principle들의 적용에 있어서는 unification의 개념을 도입하였다. feature A와 B의 unification C라는 것은, 그값의 지정에 있어서 A와도 B와도 모순이 없는 feature의 범위 내에서 최대 한도의 값을 확정한 것이다. 따라서 unification은 '일치한다'를 일반화한 개념이고, 각 principles들은 해당하는 각 syntactic category의 feature간에 unification의 가능성을 말한다고 할 수 있다.

Metarule의 적용은 action table 상에 lexical item이 적용될 수 없는 때에 한한다. 또한 같은 Metarule이 같은 ID rule 상에 2번 이상 적용되지 못하게 하기 위해, Finite Closure(FC)를 도입하여 Metarule의 무한 적용을 막는다. 이것은 Metarule의 set로부터 이미 적용되어진 Metarule을 삭제함으로서 가능하였다.

3.3 Parsing Algorithm

본 논문에서는 GPSG를 이용하여 영어의 단문, 중문, 복문과 passive voice, extraposition, inverted sentence등의 제 언어 현상을 처리하기 위한 parser를 구성하였다.

입력문의 대략적인 처리 방법은 다음과 같다. STEP1 ~ STEP11은 syntactic parsing, STEP12 ~ STEP18은 semantic parsing 과정을 나타낸다.

STEP1: 입력문을 형태소 해석하고 그 결과를 lexical feature에 부여한다.
STEP2: 순차적으로 next lexical item을 입력하여 FSD와 FCR을 적용한다.
STEP3: action table을 참조한다. table entry에 따라 shift이면 STEP2로, error이면 STEP5로, access이면 STEP12로 간다.
STEP4: FC, Metarule을 적용한다.
STEP5: LP rule을 적용한다.
STEP6: ID rule 상의 feature와 match, test를 행한다.
STEP7: partial parsing tree를 생성한다.
STEP8: CAP을 적용한다.
STEP9: FFP, HFP를 적용한다.
STEP10: FSD, FCR을 적용한다.
STEP11: goto table을 참조하여 STEP3로 간다.
STEP12: parsing partial tree를 입력한다.
STEP13: semantic feature를 삽입한다.
STEP14: lexical semantics, semantic operator를 적용한다.
STEP15: Semantic Interpretation Schema를 적용한다.
STEP16: meaning postulate을 적용하고, lambda abstract를 행한다.
STEP17: tree complete? noo이면 STEP12로 간다.
STEP18: sentence의 logical representation 추출. END.

본 system을 이용한 parsing의 실제 예로써 syntactic parsing tree와 semantics 표현을 Fig3.에 보였다.

4. 결론

본 논문에서는 영한번역 시스템 구성의 일환으로 GPSG를 이용하여 parser를 구성하였다.

본 논문에서 구현한 시스템은 VAX-11/750 상에서 Franz Lisp으로 구현되었으며, 시스템은 table과 사전 부분을 제외하고 약 1,500 line 정도의 규모를 갖는다.

parsing 과정에서 ID rule의 처리에 table driven 방식을 도입함으로써 출력 효율을 제고하였으며, 광범위한 언어 현상을 포함하는 syntactic parsing과, 논리식에 근거한 의미 추출을 행하였다.

앞으로의 연구는 ID rule 의 좀더 명확한 기술과 parsing 과정에서 Metarule의 적용 시점에 관한 고찰, 그리고 ambiguity를 해소하기 위한 feature들 의 효율적인 설정이 필요하다고 생각한다. 아울러 좀 더 일반화된 GPSG의 형태로서 lexical-oriented grammar인 HPSG (Head-driven phrase structure grammar)와 JPSG (Japanese phrase structure grammar)등의 연구도 수반되어야 할 것으로 보여진다.

참고 문헌

[1] G.Gazdar, E.Klein, G.K.Pullum & I.A.Sag "Generalized Phrase Structure Grammar" Basil Blackwell 1985

[2] P.Sells "Lectures on Contemporary Syntactic Theories" CSLI 1985

[3] M.Tomita "Efficient Parsing for Natural Language" Kluwer Academic Publishers 1986

[4] S.M.Shieber "A Simple Reconstruction of GPSG" in COLING 1986

[5] C.J.Pollard "Lecture Notes on Head-Driven Phrase Structure Grammar" IULC 1985

[6] G.Gazdar & G.K.Pullum "Generalized Phrase Structure grammar : a Theoretical synopsis" IULC 1982

[7] G.Gazdar, E.Klein, G.K.Pullum & I.A.Sag "Coordinate Structure and Unbounded Dependencies" IULC 1982

[8] D.R.Dowty "A Guide to Montague's PTQ" IULC 1978

[9] H.Thompson "Handling Metarule in a Parser for GPSG" IULC 1982

[10] S.M.Shieber "A Introduction to Unification-Based Approaches to Grammar" CSLI 1986


```
## the input sentence is ##

MAHCHINE TRANSLATION IS UNLIKELY TO REACH A SIGNIFICANTLY HIGHER LEVEL OF SUCCESS UNTIL COMPUTER PROGRAMS CAN EXHIBIT UNDERST
NDING  IN THE FULL SENSE OF THE WORLD WHERE WE LIVE IN .

## parsing succeed & good luck ##

S---S---S---NP---N1---N---$N == MACHINE
|||---|||---|||---|||---|||---|||---$N == TRANSLATION
|||---|||---|||---VP---$V == IS
|||---|||---|||---|||---AP---A1---$A == UNLIKELY
|||---|||---|||---|||---|||---|||---VP---$V == TO
|||---|||---|||---|||---|||---|||---|||---VP---$V == REACH
|||---|||---|||---|||---|||---|||---|||---|||---|||---NP---$DET == A
|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---N1---AP---AP---A1---$A == SIGNIFICANTLY
|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---A1---$A == HIGHER
|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---N1---$N == LEVEL
|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---PP---$P == OF
|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---NP---NP1---$N == SUCCESS
|||---S---$CNJ == UNTIL
|||---|||---S---NP---NP1---N---$N == COMPUTER
|||---|||---|||---|||---|||---|||---|||---$N == PROGRAMS
|||---|||---|||---VP---$V == CAN
|||---|||---|||---|||---VP---$V == EXHIBIT
|||---|||---|||---|||---|||---NP---NP1---$N == UNDERSTANDING
|||---|||---|||---|||---|||---|||---PP---$P == IN
|||---|||---|||---|||---|||---|||---|||---NP---$DET == THE
|||---|||---|||---|||---|||---|||---|||---|||---N1---AP---A1---$A == FULL
|||---|||---|||---|||---|||---|||---|||---|||---|||---N1---$N == SENSE
|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---PP---$P == OF
|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---NP---$DET == THE
|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---N1---N1---$N == WORLD
|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---S----NP---N1---$N == WHERE
|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---S---NP---N1---$N == WE
|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---VP---$V == LIVE
|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---PP---$P == IN
|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---|||---NP == TRACE

-) (pp sem_rep)
(UNTIL
 ((UNLIKELY (TO (REACH (A (ADV (SIGNIFICANTLY (HIGHER)) (LEVEL (OF (SUCCESS)))))))
   (TRANSLATION (MACHINE))) COMMA
  (CAN (EXHIBIT (IN (THE (FULL (SENSE (OF (THE (LIVE (LOC (IN (WORLD))) (WE))))))))
       (UNDERSTANDING) (PROGRAM (COMPUTER))))))
NIL
```

Fig3. syntactic parsing tree와 semantic representation의 일례