数理解析研究所講究録
第666巻 1988年 215-224

語い機能文法の生成能力の上界について

東京電機大　理工　　西野哲朗 （ Tetsuro NISHINO ）

概略： 語い機能文法（ LFG ）は，自然言語の構文を記述するために，Kaplan と Bresnan によって導入された形式的システムである．本論では，LFG によって生成される言語のクラスが，NP に含まれることを示す．これは，言語学における未解決問題への解答である．主定理の証明は次の２段階から成る．まず LFG を強多項式属性文法として表現する．そして強多項式属性文法について得られている理論的結果を，LFG に適用する．

１．はじめに

生成文法理論は 1950 年代に Chomsky [1] によって提唱されて以来，形式言語理論と自然言語理論に分かれ，それぞれ独立に発展してきた．Chomsky の自然言語理論である変形文法理論 [2] は，変形操作が可能なために帰納的可算集合を生成してしまうので，人間の言語運用のモデルとしては能力が強すぎるという批判があった．最近，Gazdar らが一般化句構造文法（ GPSG ）を提唱し [4]，自然言語の文法は文脈自由文法で記述できると主張してから，その批判は一層強まっている [5]．

そのような状況において，変形論者の言語学者 Bresnanと計算機科学者の Kaplan は，文脈自由文法を拡張して，語い機能文法（ LFG ）を導入し，変形操作を用いずに文法機能の整合性が検査できる枠組を与えた [6]．さらに Kaplan らは [6] において，LFG が生成する言語は文脈依存言語であると主張した．しかし [6] には，その証明の概略しか与えられていなかったし，またその証明の概略が，疑わしい仮定に基づいているとの批判が Gazdar らによってなされた [5]．

本論ではこの言語学における未解決問題に対し，以下のような解答を与える．「 LFG が生成する言語のクラスは，非決定性多項式時間言語のクラス NP に属する．」その証明は次のような２つの段階に分けて行う．(1) LFG が強多項式属性文法 [3] であることを示す．(2) 強多項式属性文法に関して得られている結果を LFG に適用する．

ここに属性文法とは，文脈自由文法を，その各プロダクションに意味規則を付随させることにより拡張した文法であり，従来，コンパイラの自動生成や構造エディタ等に幅広く応用されてきた文法である．また強多項式属性文法は，意味規則の計算に必要な時間量が，ある制約を満たす属性文法である [3]．

2．語い機能文法

語い機能文法 [6]（Lexical Functional Grammars，以下 LFG と略す．）においては，c 構造（constituent structure）と f 構造（functional structure）という2つの構造を用いて，文法的に適格な文の集合を記述する．c 構造は，文における単語や句などの表層的な並びを表現する木構造であり，f 構造は，主に文における文法機能の関係を表現するある種のリスト構造である．

LFG は，c 構造規則の集合と，語い項目リストにより記述される．c 構造規則は，文脈自由型プロダクションに，f 構造に関する等式（テンプレートと呼ばれる）が付随した形をしている．また，各語い項目にも，テンプレートが記入されている．図 2.1 に簡単な LFG の例を示す．

```
SE →     NP        VP
     (↑ SUBJ)=↓  ↑=↓

NP → DET    N
     ↑=↓   ↑=↓

VP →  V        NP           NP
     ↑=↓  (↑ OBJ)=↓  (↑ OBJ2)=↓
```

(a) C 構造規則

```
a:       DET,  (↑ SPEC)=A
                (↑ NUM)=SG
girl:    N,    (↑ NUM)=SG
                (↑ PRED)='GIRL'
handed:  V,    (↑ TENSE)=PAST
                (↑ PRED)
                  ='HAND <(↑ SUBJ)(↑ OBJ)(↑ OBJ2)>'
the:     DET,  (↑ SPEC)=THE
baby:    N,    (↑ NUM)=SG
                (↑ PRED)='BABY'
toy:     N,    (↑ NUM)=SG
                (↑ PRED)='TOY'
```

(b) 語い項目リスト

図 2.1. LFG の例 [6]

ここで，LFG の形式的定義を行う．図 2.1 の SE, NP, VP 等を文法範ちゅうと呼び，a, girl, handed 等を単語と呼ぶ．また，SPEC, NUM, PRED 等を機能名といい，A, SG, PAST 等を記号という．さらに，'GIRL', 'HAND<(↑ SUBJ)(↑ OBJ)(↑ OBJ2)>' 等 ' で囲まれた表現を意味形式という．また，↑，↓ をメタ変数という（その意味は後述）．

注．本稿では簡単のために，メタ変数 ↑↑，↓↓ の説明を省略している．

定義 2.1　指示子を，以下のように再帰的に定義する．

(1) a を機能名とするとき，(↑ a)，(↓ a) はそれぞれ指示子である．

(2) d を指示子，a を機能名とするとき，(↑ d)，(↓ d)，(d a) はそれぞれ指示子である．

(3) 上で定義されたもののみが，指示子である．　■

d_1, d_2 を指示子またはメタ変数とする．$d_1 = d_2$ または $d_1 \in d_2$ なる形の式をタイプ I のテンプレートという（簡単のために，後者のテンプレートについては，本稿では説明を省略する）．

d を指示子またはメタ変数，s を記号または意味形式とするとき，$d = s$ なる形の式をタイプ II のテンプレートという．タイプ II のテンプレートのうち，特に
$(\uparrow a) = s$ なる形の式（a は機能名）を，語い項目型テンプレートという．

定義 2.2 語い機能文法（Lexical Functional Grammar,LFG）とは，以下の３条件を満たす５項組 G =（C, W, R, L, SE）のことをいう．

(1) C は文法範ちゅうの集合，W は単語の集合，R は c 構造規則の集合，L は語い項目リストを表す．また，$SE \in C$ は，文を表す文法範ちゅうである．

(2) R の c 構造規則は次の形をしている．

$$X_0 \rightarrow X_1\ X_2\ \ldots\ X_m$$
$$E_1\ E_2\ \ldots\ E_m$$

ただし，$X_i \in C$, $1 \leq i \leq m$ とし，各 E_i, $1 \leq i \leq m$ はタイプ I または II のテンプレートの集合とする．

(3) L の語い項目は次の形をしている．

$$w\ :\ X,\ E$$

ただし，$w \in W$, $X \in C$ とし，E は語い項目型テンプレートの集合とする． ■

c 構造は，c 構造規則によって生成された木に，語い項目を挿入することにより得られる．図 2.1 の LFG は例えば図 2.2 のような c 構造を生成する（x_1 等の変数は，今は無視する）．

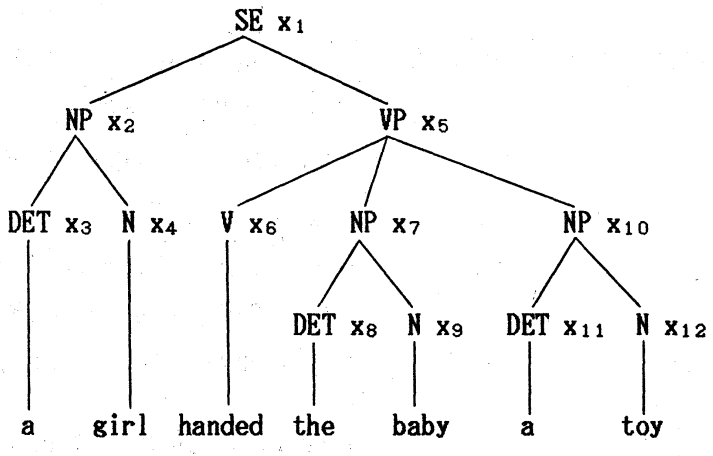

図 2.2. 図2.1 の LFG が生成する c 構造の例 [6]

c 構造内の各ノードは，それぞれ対応する f 構造をもつ．図 2.1 の規則にみられるようなメタ変数 ↓ は，そのノードに付随した f 構造を表す．また，メタ変数 ↑ は，そのノードの親に付随した f 構造を表している．したがって，ノード n に付随した等式 ↑ = ↓ は，n に付随した f 構造と，n の親に付随した f 構造が等しいことを表現している．また，ノード n に付随した等式(↑ OBJ) = ↓ は，n の親の OBJ 属性の値が，n に付随した f 構造と等しいことを表現している．

1つの c 構造 t が与えられたときに，t 全体に対応する f 構造を決定するには，以下のような手続きに従えばよい．

(1) t の各ノードに対応した f 構造を変数で表す（図 2.2 参照）．

(2) (1) の変数を用いて，t のノードに付随したすべてのテンプレートを具現化（instantiate）する．このようにして得られた等式の集まりを t の f 記述という（図 2.3 に，図 2.2 の c 構造から得られた f 記述を示す）．

(3) (2) で得られた f 記述を，連立方程式として代数的に解き，SE に対応した変数の値を求めると，それが目的の f 構造である（図 2.3 の f 記述から x_1 の値を求めると，図 2.4 に示した f 構造を得る．これが図 2.2 の c 構造に対応した f 構造である）．

上記 (3) の f 記述解法アルゴリズムに関する詳細は，Kaplan & Bresnan [6] の補遺を参照されたい．

```
[ SUBJ   [ SPEC  A
         | NUM   SG
         | PRED  'GIRL' ]

  TENSE  PAST

  PRED   'HAND <(↑ SUBJ)(↑ OBJ)(↑ OBJ2)>'

  OBJ    [ SPEC  THE
         | NUM   SG
         | PRED  'BABY' ]

  OBJ2   [ SPEC  A
         | NUM   SG
         | PRED  'TOY' ]
]
```

図 2.4. 図 2.2 の c 構造に対応する f 構造 [6]

$(x_1\ \mathrm{SUBJ}) = x_2,\ x_1 = x_5,$
$x_5 = x_6,\ (x_5\ \mathrm{OBJ}) = x_7,\ (x_5\ \mathrm{OBJ2}) = x_{10},$
$x_2 = x_3,\ x_2 = x_4,$
$x_7 = x_8,\ x_7 = x_9,$
$x_{10} = x_{11},\ x_{10} = x_{12},$
$(x_3\ \mathrm{SPEC}) = \mathrm{A},\ (x_3\ \mathrm{NUM}) = \mathrm{SG},$
$(x_4\ \mathrm{NUM}) = \mathrm{SG},\ (x_4\ \mathrm{PRED}) = \text{'GIRL'},$
$(x_6\ \mathrm{TENSE}) = \mathrm{PAST},$
$(x_6\ \mathrm{PRED}) = \text{'HAND<(↑ SUBJ)(↑ OBJ)(↑ OBJ2)>'},$
$(x_8\ \mathrm{SPEC}) = \mathrm{THE},$
$(x_9\ \mathrm{NUM}) = \mathrm{SG},\ (x_9\ \mathrm{PRED}) = \text{'BABY'},$
$(x_{11}\ \mathrm{SPEC}) = \mathrm{A},\ (x_{11}\ \mathrm{NUM}) = \mathrm{SG},$
$(x_{12}\ \mathrm{NUM}) = \mathrm{SG},\ (x_{12}\ \mathrm{PRED}) = \text{'TOY'}$

図 2.3. 図 2.2 の c 構造から得られる f 記述 [6

f 構造は厳密には，機能名とその値の対から成るリストであるが，通常，図 2.4 に示したような行列形式で表現される．1つの c 構造には，それに対応する f 構造が一意に定まる．f 構造に対しては，その適格性を決定するための適格性条件が以下のように定められている．適格な f 構造に対応する c 構造は，適格であると定義する．

f 構造の適格性条件

1. (唯一性) すべての属性が1つ，そしてただ1つの値を持つ．
2. (完全性) PRED 属性の値の下位範ちゅう化で現れる文法機能は，同じ f 構造内の属性になっている．(動詞や形容詞を，それが現れる文脈によって，より詳細な文法範ちゅうに分類することを下位範ちゅう化という．)
3. (一貫性) f 構造内の下位範ちゅう化できる文法機能は，PRED 属性の値のなかに現れる． ■

主張 2.1 D を，ある1つの c 構造に付随した f 記述とし，D に現れる等式の個数を n とする．そのとき，D から一意に定まる最大の f 構造 F が適格であるか否かは，n の多項式時間で決定可能である．

証明 : Kaplan & Bresnan [6] の補遺に示されているアルゴリズムから，次の2つが導かれる : (1) f 記述 D から，それが記述している f 構造 F を決定することは n の多項式時間でできる．(2) F のサイズは n の多項式で押えられる．さらに F の適格性は，上の4条件からわかるように，F のサイズの多項式時間で決定できる． ■

LFG G の生成言語とは，G の適格な C 構造が導出する終端記号列全体の集合のことである．LFG の定義から，LFG の生成言語のクラスが，文脈自由言語のクラスより大きいことは明らかである．しかし，そのクラスがどの位大きいのかは，冒頭にも述べたように，未だに評価されておらず，言語学の未解決問題となっている [5]．

3．属性文法の定義

この節では，属性文法の一般的定義と強多項式属性文法の定義を述べる．文脈自由文法を $Gr = (\Sigma, \Gamma, P, Z)$ で表す．ただし，各成分はそれぞれ，非終端記号，終端記号およびプロダクションの有限集合と，開始記号を表している．P のプロダクション p を

$$p : X_0 \rightarrow w_0\ X_1\ w_1\ \ldots\ w_{n-1}\ X_n\ w_n$$

と書く．ただし，$X_j \in \Sigma$，$w_j \in \Gamma^*$，$0 \leq j \leq n$，$n \geq 0$ とする．Gr は通常の意味で既約であるとし，また，いかなる非終端記号 X に対しても，$X \rightarrow^+ X$ なる形の導出を持たないと仮定する．

属性文法は，特定の意味領域上で定義される．意味領域Dom とは２項組 (Val, Fun) のこととする．ここに，Val は各属性の値域から成るクラスである．また，Fun は $f : V_1 \times V_2 \times \cdots \times V_m \rightarrow V_0$ なる形をした関数の集合とする．ただし $m \geq 0$ とし，さらに $0 \leq i \leq m$ に対し，$V_i \in Val$ とする．Fun に属する関数を意味関数と呼ぶ．$m = 0$ の場合は，f は V_0 の要素を表す．

定義 3.1 属性文法 (Attribute Grammar，以下 AG と略す) とは，以下の３条件を満たす４項組 $G = \langle Gr, Attr, Rule, Dom \rangle$ のことをいう．

(1) $Gr = (\Sigma, \Gamma, P, Z)$ は文脈自由文法であり， G の基底文脈自由文法と呼ばれる．

(2) Σ の各非終端記号 X に対し，属性の有限集合 A(X) が付随している．X の属性 a を a(X) と表す．$Attr = \bigcup_{X \in \Sigma} A(X)$ とする．X の属性の集合 A(X) は，継承属性の集合 I(X) と合成属性の集合 S(X) に分割される．すなわち，$A(X) = I(X) \cup S(X)$，$I(X) \cap S(X) = \phi$．次のことを仮定する：$A(X) \neq \phi$，$I(Z) = \phi$ かつ $|S(Z)| = 1$．すなわち，開始記号 Z は１つの合成属性しか持たない．各属性 a について，Val は a の値域 V(a) を含んでいるものとする．

(3) P のプロダクション p に対し，意味規則の有限集合 R_p が付随している．$Rule = \bigcup_{p \in P} R_p$ とする．プロダクション p において，$A(X_i)$, $0 \leq i \leq n$ の属性 a を，p における属性 a の生起といい，a(i) で表す．p の属性生起全体の集合を Occ(p) と書く．Occ(p) は，次のように分割される：

$In(p) = \{ a(i) \mid a \in I(X_0)$ または $a \in S(X_i), i \geq 1 \}$,

$Out(p) = \{ a(i) \mid a \in S(X_0)$ または $a \in I(X_i), i \geq 1 \}$

とすれば，$Occ(p) = In(p) \cup Out(p)$，$In(p) \cap Out(p) = \phi$．P の各プロダクション p に対し，$R_p$ は Out(p) の各属性を定義する意味規則を１つ含んでいる．属性 a(k)，$0 \leq k \leq n$ を定義する意味規則は次の 1)，2) により定義される．

1) $f : V_1 \times V_2 \times \cdots \times V_m \rightarrow V_0$，$V_0 = V(a(k))$ なる形をした Fun の関数 f．

2) m 個の属性 $a_j(i_j)$，$1 \leq j \leq m$，$0 \leq i_j \leq n$．ただし，すべての j について，$V_j = V(a_j(i_j))$ が満たされているものとする．

1)，2) により定義される意味規則を $a(k) := f (a_1(i_1), \cdots, a_m(i_m))$ と書く．また R_p は，１つの条件式 C_p を含んでいる．ただし，C_p に現れる属性は In(p) の要素である． ■

属性文法 G の属性付き木とは，G の導出木で，その各内部ノードに付随したすべての属性値が，意味規則に従って計算されており，かつ，各内部ノードですべての述語が真と評価されているものをいう．

以下では，計算量理論の用語を用いるので，それらの定義をここで述べておく．自然数上の関数 f は，決定性チューリング機械を用いて，入力の長さの多項式回のステップ数で計算できるときに，多項式時間計算可能であるという．m をあるチューリング機械とし，L をアルファベットX 上の言語とする．任意の $x \in X^*$ に対し，$x \in L$ か否かを，m を用いて $|x|$ の多項式時間で判定できるときに，言語 L は m により多項式時間認識可能であるという．非決定性チューリング機械により多項式時間認識可能な言語のクラスを NP という．

定義 3.2 [3] 強多項式属性文法 (Strongly PolynomialAttribute Grammar, 以下 SPAG と略す) とは，以下の４条件を満たす AG = < Gr, Attr, Rule, Dom > のことをいう．

(1) X をあるアルファベットとするとき，Val は X^* のみを含む．

(2) Rule に現れるすべての関数 f および述語 C_p は，多項式時間計算可能である．

(3) Rule に現れるすべての関数 f に対し，ある定数 c が存在して，任意の $x_1, \ldots, x_m \in X^*$ に対し，$|f(x_1, \ldots, x_m)| \leq c + \sum_{i=1}^{m} |x_i|$ が成立する．

(4) G の任意の属性付き木において，すべての属性値が，木のノード数の多項式時間で計算できる． ■

属性文法 G の生成言語とは，G の属性付き木の yield (葉のラベルになっている終端記号を左から順に並べて得られる文字列) 全体からなる集合をいう．L_{SPAG} で SPAGの生成言語のクラスを表す．次の定理が得られている．

定理 3.1 [3] $unpad(L_{SPAG}) = NP$ ■

言語のクラス W に対し，unpad(W) は次のように定義される：p を多項式とするとき，言語 L に対し，$pad_p(L) = \{ \#^{p(|x|)}x \mid x \in L \}$ と定義する．# は無意味な記号であり，$pad_p(L)$ は言語 L の語 x の前に，# を $p(|x|)$ 個連接して得られる列全体の集合である．このとき，$unpad(W) = \{ L \mid$ ある多項式 p が存在して $pad_p(L) \in W \}$ と定義する．

４．属性文法による LFG の特徴付けとその応用

任意の LFG は，下のアルゴリズム 4.1 に示す方法で，それと等価な (同じ言語を生成する) SPAG に変換することができる．以下では，"abc" で文字列 abc を表し，・ で文字列の連接を表す．また，$x_1, \ldots, x_m$ を文字列とするとき，関数 CONCAT を次のように定義する：$CONCAT(x_i) = x_1 \cdot "\#" \cdot x_2 \cdot "\#" \cdot \ldots \cdot "\#" \cdot x_m$ ($1 \leq i \leq m$)

アルゴリズム 4.1 では，属性生起 a(i) を $[a,X_i]$ と表している．

アルゴリズム 4.1　LFG を SPAG に変換する

入力：LFG $G_0 = (C, W, R, L, SE)$

出力：G_0 と等価な SPAG $G = \langle Gr, Attr, Rule, Dom \rangle$，ただし，$Gr = (\Sigma, \Gamma, P, Z)$ とする．

手続き：[1] Z を C には含まれない記号とするとき，$\Sigma = C \cup \{ Z \}$ とし，$\Gamma = W$ とする．また，$\Sigma - \{ Z, SE \}$ の各非終端記号 X に対し，$I(X) = \{ b \}$, $S(X) = \{ c, f \}$ とする．さらに，$I(Z) = \phi$, $S(Z) = \{ f \}$, $I(SE) = \{ b \}$, $S(SE) = \{ f \}$ と定める．

[2] (1) の形をした R の各 c 構造規則から，以下のようにしてプロダクション $p \in P$ および，Rule の意味規則集合 R_p を構成する．

$$X_0 \rightarrow X_1\ X_2\ \ldots\ X_m \quad (1)$$
$$E_1\ E_2\ \ldots\ E_m$$

ただし，E_i はテンプレートの集合とし，$E_i = \{ e_{i1}, e_{i2}, \ldots, e_{ik(i)} \}$ とする．

1. 文脈自由規則 $p : X_0 \rightarrow X_1\ X_2\ \ldots\ X_m$ を P に加える．
2. (b および c に関する意味規則)

$$[b,X_1] := [b,X_0] + 1;$$
$$[b,X_j] := [c,X_{j-1}] + 1, \text{ for } 2 \leq j \leq m;$$
$$[c,X_0] := [c,X_m]$$

を R_p に加える．

3. (f に関する意味規則)

各テンプレート $e_{ij} \in E_i$, $1 \leq i \leq m$, $1 \leq j \leq k(i)$ 内のメタ変数 ↑ を文字列 "x[b,X_0]" で，メタ変数 ↓ を文字列 "x[b,X_i]" でそれぞれ置き換えたものを d_{ij} とする．　$A = \underset{\substack{1 \leq i \leq m \\ 1 \leq j \leq k(i)}}{\mathrm{CONCAT}}\ (d_{ij})$,　$B = "[f,X_1]\#[f,X_2]\#\ldots\#[f,X_m]"$

とするとき，$[f,X_0] := A \cdot "\#" \cdot B$ を R_p に加える．

4. $C_p = \text{true}$ とする．

[3] (2) の形をした L の各語い項目から，以下のようにしてプロダクション $p \in P$ および，Rule の意味規則集合 R_p を構成する．

$$w :\ X, E \quad (2)$$

ここに，$E = \{ e_1, \ldots, e_m \}$ とする．

1. 文脈自由規則 $p : X \rightarrow w$ を P に加える．
2. (b および c に関する意味規則)

$[c,X] := [b,X]$ を R_p に加える．

3. (f に関する意味規則)

各テンプレート e_i, $1 \leq i \leq m$ の左辺に現れる ↑ を文字列 "x[b,X]" で置き換えたものを d_i とする. $[f,X] := \mathrm{CONCAT}_{1 \leq i \leq m}(d_i)$

を R_p に加える.

4. C_p = true とする.

[4] 以下の規則を追加する.

1. 文脈自由規則 p : Z → SE を P に加える.
2. [f,Z] := [f,SE], [b,SE] := 1 を R_p に加える.
3. C_p = VALID([f,SE]) とする. ただし VALID(D) は, f 記述 D から一意に定まる最大の f 構造が適格なとき, かつそのときにかぎり真となる. ■

注. 上の [2] および [3] で, 長距離依存関係の記述で用いるメタ変数 ⇑, $\Uparrow_{NP}$, $\Downarrow_{Pro}$ 等は, d_{ij} および d_i を構成する際に, 適当な変数で置き換えればよい (本稿では詳しくは述べない).

LFG の生成言語全体のクラスを L_{LFG} であらわすと, 次の定理を得る.

定理 4.1 $L_{LFG} \subseteq L_{SPAG}$

証明 アルゴリズム 4.1 により, LFG G_0 から構成される属性文法を G とする. 最初に, G が定義 3.2 の4条件を満たしていることを示す. まず (1) については明らか. 次に, G の意味規則に現れる関数は, 後者関数 (1 を加える関数), 恒等関数および, 文字列の連接関数の3つであり, これらはすべて条件 (2), (3) を満たす.

G の基底文脈自由文法 Gr の1つの導出木を t とし, t から根 Z を取り去った, SE を根とする t の部分木を t' で表す. G の意味規則により, t' を LFG G_0 の c 構造とみたときの, それに対応する f 記述が, t の SE ノードに付随した属性 [f,SE] の値として求まる. したがって, 主張 2.1 により, G のプロダクション Z → SE に付随した述語 VALID([f,SE]) の真偽は, [f,SE] のサイズの多項式時間で判定できる. G の Rule に現れる他の述語はすべて true なので, Rule に現れるすべての述語は, 条件 (2)を満たす.

また, G における属性評価の過程は, t の各ノードを左から右への深さ優先順序で1度ずつ訪れて, それぞれのノードで f 記述の断片を文字列として連接しながら記録して行くものなので, 条件 (4) は満たされる. 最後に, G の意味規則の構成の仕方から, G と G_0 が等価であることは明らかである. ■

系 4.1 $L_{LFG} \subseteq NP$

証明 定理 4.1 および定理 3.1 より，

$$L_{LFG} \subseteq unpad(L_{LFG}) \subseteq unpad(L_{SPAG}) = NP.$$ ■

5．おわりに

本論で行った LFG の属性文法による特徴付けは，言語学や自然言語処理と属性文法の関係を明確にする１つの試みであり，将来，自然言語処理の分野にコンパイラ構成法を適用して行く際の基礎になるものと期待される．同時に本研究は，Chomsky [1] 以来，自然言語理論とは独立に発展してきた形式言語理論の，自然言語理論へのフィードバックを目指すものでもある．

参考文献

1. Chomsky, N. : On Certain Formal Properties of Grammars, Information and Control, 2 : 2, 137-167.
2. Chomsky, N. : Lectures on Government and Binding : The Pisa Lectures, Dordrecht, Foris (1986).
3. Ephremidis, S., Papadimitriou, C. H. and Sideris, M. : Complexity Characterizations of Attribute Grammar Languages, Proc. Structure in Complexity Theory (1987).
4. Gazdar, G., Klein, E., Pullum, G.K., and Sag, I.A. : Generalized Phrase Structure Grammar, Oxford, Basil Blackwell (1985).
5. Gazdar, G. and Pullum, G. K. : Computationally Relevant Properties of Natural Languages and Their Grammars, New Generation Computing 3, pp.273-306 (1985).
6. Kaplan, R. and Bresnan, J. : Lexical-Functional Grammar, In Bresnan, J. (ed.) The Mental Representation of Grammatical Relations, Cambridge, Mass., MIT Press (1982).