数理解析研究所講究録
943巻 1996年 106-113

Variants of alternating grammars

早大教育　守屋悦朗　(Etsuro Moriya)

交代性文脈自由文法 (alternating CFG : ACFG) は交代性プッシュダウンオートマトン (alternating PDA : APDA) を文法的に特徴付けるために [1, 2] で初めて導入され、その後、継続的な研究がいくつかある [3, 4, 5, 6]。ここでは、それらを簡単にサーベイするとともに、ACFG のいくつかの変種について考察する。

1 Alternating CFG

交代性文脈自由文法 (ACFG) は

$$G = (N, U, \Sigma, P, S)$$

で表わされるシステムである。ここに、

(1) $\underline{G} = (N, \Sigma, P, S)$ は CFG (underlying CFG という) であり、N は非終端記号の集合、Σ は終端記号の集合、P はプロダクションの集合、$S \in N$ は開始記号である。

(2) U は N の部分集合である。U の元は**全称的** (universal) であるといい、$E = N - U$ の元は**存在的** (existential) であるという。

左辺が A であるプロダクションが全部で $A \to \alpha_1, \cdots, A \to \alpha_k$ であるとき、$A \in U$ ならば $A \to \alpha_1, \cdots, \alpha_k$ と書き、$A \in E$ ならば $A \to \alpha_1 \mid \cdots \mid \alpha_k$ と書く。G のどのプロダクションの右辺も ε (空語) でないとき、G は **ε-free** であるという。また、$\underline{G}$ が線形あるいは右線形のとき、G は**線形** (linear) あるいは**右線形** (right-linear) であるという。

語 $w \in \Sigma^*$ の G における**導出木** (derivation tree) とは、各ノードに $(N \cup \Sigma)^*$ の元がラベル付けされた有限の根付き木 T で、次の条件を満たすものである (このとき、T は w を**生成する**という)。

(i) T の根のラベルは S である。

(ii) T の葉のラベルはすべて w である。

(iii) π を T の内部ノード、$\alpha A \beta$ $(\alpha, \beta \in (N \cup \Sigma)^*, A \in N)$ をそのラベルとする。

(a) $A \to \gamma_1, \cdots, \gamma_k$ ならば、π はラベルがそれぞれ $\alpha\gamma_i\beta$ $(1 \leq i \leq k)$ である丁度 k 個の子供を持つ。このとき、π を**全称的ノード**という。

(b) $A \to \gamma_1 \mid \cdots \mid \gamma_k$ ならば、π はラベルが $\alpha\gamma_i\beta$ $(1 \leq i \leq k)$ のいずれかである丁度1個の子供を持つ。このとき、π を**存在的ノード**という。

とくに、T のすべてのノードにおいて、$\alpha \in \Sigma^*$ であるときだけ (iii) が適用されているならば、T は**最左** (leftmost) であるという。

T における根を始点とする道を G による**導出** (derivation) といい、T の各ノードのラベルを G の**文形式** (sentential form) という。

G は言語

$$L(G) = \{\, w \in \Sigma^* \mid G \text{における } w\text{の導出木が存在する} \,\}$$

を生成する。ACFG によって生成された言語を**交代性文脈自由言語** (alternating CFL : **ACFL**) という。とくに、G の導出木を最左に限定したとき、G を **left-ACFG** といい、

$$L_{left}(G) = \{\, w \in \Sigma^* \mid G \text{における } w\text{の最左導出木が存在する} \,\}$$

を **left-ACFL** という。タイプ X の文法で生成される言語のクラスを $\mathcal{L}$(X) で表わす。

ACFG はもともと APDA を文法で特徴付けることを目的に導入されたものであるが、その最初の試み [2] には誤りがあった。根本的な難しさは、ACFG を (1) 最左導出に制限できるか、(2) ε-free にできるかにある。この問題は現在も解決されていないが、ACFG にこれらの制限の両方あるいは一方を課すことによって、いくつかの言語クラスを特徴付けることができる。

$t(n)$ 時間限定、$s(n)$ 領域限定の DTM によって受理される言語のクラスをそれぞれ **DTIME**$(t(n))$, **DSPACE**$(s(n))$ で表わす。NTM に関して **NTIME**$(t(n))$ および **NSPACE**$(s(n))$ が、また、alternating TM に関して **ATIME**$(t(n))$ および **ASPACE**$(s(n))$ が同様に定義される。とくに、

Reg=**DSPACE**(1) = 正則集合のクラス、$\mathcal{L}$(CSG) = **NSPACE**(n),

P= $\cup_{i\geq 1}$**DTIME**(n^i), **PSPACE**= $\cup_{i\geq 1}$**DSPACE**(n^i)

である。CSG は文脈依存文法を表わす。また、言語のクラス $\mathcal{L}$ に対して

$$LOG(\mathcal{L}) = \{\, L \mid \exists L' \in \mathcal{L} \text{ s.t. } L \leq_{log} L' \,\}$$

と定義する。ここに、$L \leq_{log} L'$ は L が L' に (many-one reduction のもとで) 対数領域還元可能であることを表わす。

定理 1.1 $\mathcal{L}$(right-linear ACFG) =**Reg**.

定理 1.2 [4] $LOG(\mathcal{L}(\text{linear ACFG})) = \mathbf{P}$.

定理 1.3 [4] $LOG(\mathcal{L}(\varepsilon\text{-free left-ACFG})) = \mathbf{PSPACE}$.

したがって、$\mathcal{L}$(linear ACFG) $\subseteq$ **P**, $\mathcal{L}$(ε-free left-ACFG) $\subseteq$ **PSPACE** であるが、等号が成り立つかどうかは知られていない (おそらく成り立たないであろう)。

次に、ACFG の定義を次のように少し修正する。\$ を Σ の元でない記号とする。\$ は終端語の endmarker としての役割を持つ。**marker 付き ACFG** とは

$$G = (N, U, \Sigma \cup \{\$\}, P, S)$$

のことである。marker 付き ACFG においては、プロダクションの右辺および文形式は $(N \cup \Sigma \cup \{\$\})^*$ の元である。導出木の定義において、(ii) を

(ii′) T の葉のラベルはすべて $\$w\$$ である。

に変更し、$L(G)$ の定義を

$$L(G) = \{\ w \in \Sigma^* \mid G \text{における } \$w\$ \text{ の導出木が存在する}\ \}$$

に変更する。次の条件を満たす定数 c が存在するとき、G は**線形消去的** (lenear erasing) であるという:

$$\forall w \in L(G)\ \exists T\ (T\text{は } G \text{における導出木})\ \forall \alpha\ (\alpha\text{は } T\text{のノードのラベル})\ |\alpha| \leq c|w|.$$

定理 1.4 [5] $\mathcal{L}$(linear erasing ACFG) = $\mathcal{L}$(APDA) = $\cup_{c>0}$**DTIME**(c^n). ここに、$\mathcal{L}$(APDA) は APDA によって受理される言語のクラスである。

2 Alternating CSG

ACSG(alternating CSG, **交代性文脈依存文法**) とは $G = (N, U, \Sigma, P, S)$ のことである。ここに、P 以外は ACFG のそれとまったく同じである。P の元は

$$\alpha X \beta \to \alpha x \beta, \quad \alpha, \beta \in N^*,\ X \in N,\ x \in (N \cup \Sigma)^+$$

なる形のプロダクションである ($\alpha, \beta \in (N \cup \Sigma)^*$ としても、あるいは、プロダクションの形を

$$\alpha \to \beta, \quad \alpha, \beta \in (N \cup \Sigma)^+, |\alpha| \leq |\beta|$$

と定義しても、以下の結果には影響しない)。

ACSG における導出木および最左導出の定義は ACFG の場合と同様である。ただし、ACSG の場合には、プロダクションの左辺の最も左の非終端記号が全称的であるか存在的であるかによって、プロダクションが全称的であるか存在的であるかを定義する。

定理 2.1 [6] $\mathcal{L}$(left-ACSG)=$\mathcal{L}$(APDA).

CSG と等価である線形有界オートマトン (LBA) に alternation の機能を加えたとき $\mathcal{L}$(alternating LBA)=$\mathcal{L}$(APDA) であることを考えると、この結果は自然な帰結といえる。

このように ACSG に alternation の機能を持たせると CSG よりも真に能力が上がるわけであるが、ACSG における alternation の回数を有限回に制限すると能力は上がらず CSL の範囲にとどまることも分かっている (3 節参照)。

3 Alternation-bounded ACFG

ここでは alternation の回数を限定した ACFG や ACSG について考察する。

ACFG や ACSG による導出 (導出木の根から葉への道) において、存在的ノード (あるいは全称的ノード) から全称的ノード (存在的ノード) へ移行することを **alternation** という。導出木 T における alternation の回数の最大値を T の alternation 回数と定義する。ACFG(ACSG) G のどの導出木も alternation の回数が高々 k であるとき、G は **k-ALT 有界** (k-ALT-bounded) であるという。とくに、k-ALT 有界かつ S が存在的であるような ACFG(linear ACFG, ACSG) によって生成される言語のクラスを **Σ_k-ACFL** (**Σ_k-LACFL**, **Σ_k-ACSL**) で表わす。Σ_1-ACFL = $\mathcal{L}$(CFG) $\subsetneq$ Σ_2-ACFL であることは容易に示すことができる。

一般の ACFG を ε-free にできるかどうかは現在のところ未解決であるが、次の定理に述べるように、ALT 有界 left-ACFG は ε-free にすることができる。

定理 3.1 [3] $L \subseteq \Sigma^*$, $L \in \mathcal{L}$(ALT-bounded left-ACFG) とする。\$ を Σ に属さない記号とすると $L\$ \in \mathcal{L}$(ALT-bounded ε-free left-ACFG) である。

この定理により、$\mathcal{L}$(ALT-bounded left-ACFG) $\subseteq$ **DSPACE**(n) であることが示せるが、最左導出という制限をなくすと ACFG を ε-free にできるかどうかが分かっていないので、$\mathcal{L}$(bounded ACFG) が **ASPACE**(n) = $\mathcal{L}$(APDA) = $\cup_{c>0}$**DTIME**(c^n) = $\mathcal{L}$(linear erasing ACFG) に含まれるかどうかは明らかでない。

初期状態が存在的で、高々 k 回の alternation しか起こさないような APDA が受理する言語のクラスを **Σ_k-APDA** で表わす。Σ_k-ACFL と Σ_k-APDA の間に単純な関係はなさそうである。次のことが知られている。

命題 3.1 [7] $\mathcal{L}$(CSG) = **NSPACE**(n) $\subseteq$ Σ_k-PDA $\subseteq$ **DSPACE**(n^2), $k \geq 2$.

定理 3.2 [6] $\mathcal{L}$(ALT-bounded left-ACSG) = $\mathcal{L}$(CSG).

この定理の証明では Σ_k-APDA $\subseteq$ Σ_k-left-ACSL であることを示しているので、次の系を得る。これは有界 APDA を特徴づけるものであるが、ここで考えている APDA は one-way モデルであり、(two-way APDA は one-way APDA で模倣できるが) 有界

two-way APDA を有界 one-way APDA で模倣できるかどうかは不明なので、残念ながら命題 3.1の refinement にはなっていない。。

系 3.1 $k \geq 2$ なる任意の k に対して、Σ_k-APDA $=$ **NSPACE**$(n) = \mathcal{L}$(CSG) である。

さて、定理 1.2を考慮し、linear ACFG がどのくらいの生成能力を持つかを考察したい。$\mathcal{L}$(linear ACFG) $\subseteq$ **P**, $\mathcal{L}$(CFG) $\subseteq$ **P**, $\mathcal{L}$(linear ACFG)$-\mathcal{L}$(CFG) $\neq \emptyset$ であるが、$\mathcal{L}$(CFG) $\subseteq \mathcal{L}$(linear ACFG) であろうか？現在わかっているのは次のことだけである。

命題 3.2 $k \geq 2$ ならば、Σ_k-LACFL と $\mathcal{L}$(CFG) とは比較不能である。

4 その他の alternating grammar

1970 年代、CFG の様々な拡張が導入された。それらは CFG をベースにして定義されているため、容易に alternation の概念を導入することができる。ここでは、状態文法 [9] と制御集合付き文法 [11] を取り上げる。なかでも状態文法は alternation の概念が最も自然に導入出来る。

交代性状態文法 (**ASG**, alternating state grammar) とは

$$G = (K, U, N, \Sigma, P, S, s_0, F)$$

で表わされるシステムである。ここで、

(1) K は**状態**の有限集合、U は K の部分集合である。U の元は**全称状態**と呼ばれ、$E = K - U$ の元は**存在状態**と呼ばれる。s_0 は K の特別な元で、**初期状態**と呼ばれる。$F \subseteq K$ は**受理状態**の集合である。

(2) N は非終端記号の有限集合、Σ は終端記号の有限集合、$S \in N$ は開始記号である。

(3) P は次の形をしたプロダクションの有限集合である:

$$(p, A) \to (q, \alpha), \quad p, q \in K,\ A \in N,\ \alpha \in (N \cup \Sigma)^*$$

左辺が (p, A) であるプロダクションが全部で $(p, A) \to (q_1, \alpha_1), \cdots, (p, A) \to (q_k, \alpha_k)$ であるとき、$A \in U$ ならば $(p, A) \to (q_1, \alpha_1), \cdots, (q_k, \alpha_k)$ と書き、$A \in E$ ならば $(p, A) \to (q_1, \alpha_1) \mid \cdots \mid (q_k, \alpha_k)$ と書く。

(4) $\underline{P} = \{\ A \to \alpha \mid (p, A) \to (q, \alpha) \in P\ \}$ とするとき、$\underline{G} = (N, \Sigma, \underline{P}, S)$ を G の underlying CFG という。$\underline{G}$ が ε-free であるとき、G は **ε-free** であるという。

G における (最左) 導出 (木) は ACFG の場合と同様に定義されるが、プロダクションの適用にあたっては次に述べるような制約がある。

G における文形式とは $K \times (N \cup \Sigma)^*$ の元のことをいう。プロダクション $(p, X) \to (q, x)$ が文形式 (r, α) に適用可能であるとは $r = p$ かつ $\alpha = \beta X \gamma$ となる $\beta, \gamma \in (N \cup \Sigma)^*$ が存在することである。また、このとき、X は (p, α) において書き換え可能であるという。(p, α) において書き換え可能な非終端記号のうち α の中で最も左側に位置するものを**可能最左**な非終端記号という。X が (p, α) において可能最左な非終端記号で、かつ X が α の中の左から k 番目以内の非終端記号であるとき、X は**k可能最左**であるという。

語 $w \in \Sigma^*$ の G における導出木とは、各ノードに $K \times (N \cup \Sigma)^*$ の元がラベル付けされた有限の根付き木 T で、次の条件を満たすものである。

(i) T の根のラベルは (s_0, S) である。

(ii) T の葉のラベルはすべて $F \times \{w\}$ の元である。

(iii) π を T の内部ノードとし、$(p, \alpha A \beta)$ をそのラベルとする。ただし、$p \in K$, $A \in N$, $\alpha, \beta \in (N \cup \Sigma)^*$ で、A は $(p, \alpha A \beta)$ において可能最左な非終端記号とする。

(a) $(p, A) \to (q_1, \gamma_1), \cdots, (q_k, \gamma_k)$ ならば、π はラベルがそれぞれ $(q_i, \alpha \gamma_i \beta)$, $1 \leq i \leq k$, である丁度 k 個の子供を持つ。

(b) $(p, A) \to (q_1, \gamma_1) \mid \cdots \mid (q_k, \gamma_k)$ ならば、π はラベルが $(q_i, \alpha \gamma_i \beta)$, $1 \leq i \leq k$, のいずれかである丁度 1 個の子供を持つ。

とくに、(iii) において A が n 可能最左であるとき、T を **left(n) 導出**という。また、(iii) において「可能最左」という条件がないとき、G は**自由** (free interpreted) であるという。

$$L(G) = \{\, w \in \Sigma^* \mid G \text{における } w \text{の導出木が存在する} \,\},$$

$$L(G; n) = \{\, w \in \Sigma^* \mid G \text{における } w \text{の left}(n) \text{ 導出木が存在する} \,\}$$

と定義し、$L(G) = L(G; n)$ であるとき G および $L(G)$ をそれぞれ **left(n)-ASG**, **left(n)-ASL** ($U = \emptyset$ のときには **left(n)-SG**, **left(n)-SL**) という。

状態文法 (SG) とは $U = \emptyset$ である ASG のことであるが、その生成能力については次のことが知られている [9, 10, 11]:

命題 4.1 (1) $\mathcal{L}(\varepsilon\text{-free CFG}) \subsetneq \mathcal{L}(\varepsilon\text{-free free interpreted SG}) \subsetneq \mathcal{L}(\text{CSG})$.

(2) $\mathcal{L}(\text{CFG}) = \mathcal{L}(\varepsilon\text{-free left(1)-SG}) \subsetneq \cdots \subsetneq \mathcal{L}(\varepsilon\text{-free left}(n)\text{-SG})$
$\subsetneq \mathcal{L}(\varepsilon\text{-free left}(n+1)\text{-SG}) \subsetneq \cdots \subsetneq \mathcal{L}(\varepsilon\text{-free SG}) = \mathcal{L}(\text{CSG})$.

(3) $\mathcal{L}(\text{SG})$=帰納的可算言語のクラス.

SG に alternation を導入すると自然に APDA に対応する:

定理 4.1 (1) $\mathcal{L}$((ε-free) ACFG) $\subseteq$ $\mathcal{L}$((ε-free) free interpreted ASG).

(2) $\mathcal{L}$(ε-free left-ACFG) $\subseteq$ $\mathcal{L}$(ε-free left(1)-ASG) = $\mathcal{L}$(APDA).

実は、ACFG には多少不自然な点がないわけではない。例えば、全称的プロダクションだけからなる ACFG は有限集合しか生成できない。しかし、ASG ではこのようなことがなく、その他の点でも ASG と APDA はごく自然に対応するように思われる。

次に、制御集合付き文法を考える。ここでは、プロダクションの適用順を制御集合と呼ばれるグラフ言語によって制限した CFG を考える。

$\underline{G}$=(N, Σ, P, S) を CFG とする。P の部分集合をラベルとするラベル付き根付き木を $\underline{G}$の**制御木**と呼ぶ。$\underline{G}$の制御木すべての集合を $\mathcal{C}_{\underline{G}}$で表わす。$\mathcal{C}_{\underline{G}}$の部分集合 C を $\underline{G}$の制御集合という。C を認識する木オートマトン (root-to-frontier tree automaton) [12] が存在するとき、C は**正則**であるという。CFG $\underline{G}$ と C の順序対 G=($\underline{G}$,C) を**正則制御集合付き ACFG(ACFG with a regular control set)** という。

T を $\underline{G}$の制御木とするとき、$val(T)$ を次のように定義する。π を T の任意のノードとする。π のラベルを次のように $(N \cup \Sigma)^*$ の元で付け替える：

(1) T の根の新しいラベルを S とする。

(2) π の元のラベルが $p = \{X_1 \to x_1, \cdots, X_k \to x_k\}$ で、新しいラベルが α であるとする。p のプロダクションの左辺として現れる非終端記号のうち α において最も左にあるものを X とする。このような p の元を $X \to x'_1, \cdots, X \to x'_\ell$ とし、$\alpha = \beta X \gamma$ とする。π がちょうど ℓ個の子供を持つとき、それら (順序は任意) の新しいラベルをそれぞれ $\beta x'_1 \gamma, \cdots, \beta x'_\ell \gamma$ とする。それ以外の場合、$val(T)$ は定義されない。

(3) T のすべての葉の新しいラベルが $w \in \Sigma^*$ であるならば、$val(T) = w$ と定義する。それ以外の場合、$val(T)$ は定義されない。

G=($\underline{G}$,C) が生成する言語を $L(G) = \{\ val(T) \mid T \in C\ \}$ と定義する。

(注) ACFG の本来の定義を考慮すると、制御集合付き ACFG を次のように定義することも考えられる。

G を ACFG とし、T を G の導出木とする。T のラベルを次のように付け替えたものを $c(T)$ で表わす。π を T の任意のノードとする。

(a) π が全称ノードで π において適用されたプロダクションが $A \to \gamma_1, \cdots, A \to \gamma_k$ ならば、π のラベルを $\{A \to \gamma_1, \cdots, A \to \gamma_k\}$ とする。

(b) π が存在ノードで π において適用されたプロダクションが $A \to \gamma_i$ ならば、π のラベルを $\{A \to \gamma_i\}$ とする。

$G = (\underline{G}, C)$ が生成する言語を

$$L(G) = \{\ w \mid w\text{を生成する}\underline{G}\text{の導出木が存在し、}c(T) \in C\text{である}\ \}$$

で定義する。

制御集合付き ACFG の性質については別の機会に述べる。

References

[1] 守屋悦朗、CFG における並列性–とくに alternation– について、LA シンポジウム、1987.2.

[2] Moriya,E., A grammatical characterization of alternating pushdown automata, *Theor. Comput. Sci.* **67** (1989), 75–85.

[3] 中村愛、Alternating context-free grammar について、信学技報 COMP87-15 (1987), 47–52.

[4] Chen,Z-Z. and S.Toda, Grammatical characterizations of P and PSPACE, *IEICE Trans.* **E73** (1990), 1540–1548.

[5] Ibarra,O.H, T.Jiang and H.Wang, A charcterization of exponential-time languages by alternating context-free grammars, Technical Report No.91-13, McMaster University, 1991.

[6] 中山晋、Grammatical charcterizations of alternating pushdown automata and linear bounded automata, 早稲田大学修士論文、1996.2.

[7] Ladner,R., L.J.Stockmeyer and R.J.Lipton, Alternation bounded auxiliary pushdown automata, *Inform. Contr.* **62** (1984), 93–108.

[8] Chandra,A.K., D.C.Kozen and L.J.Stockmeyer, Alternation, *J. ACM* **28** (1981), 114–133.

[9] Kasai,T., An infinite hierarchy between context-free and context-sensitive languages, *J. Comput. Sys. Sci.* **4** (1970), 492–508.

[10] Moriya,E., Some remarks on state grammars and matrix grammars, *Inform. Contr.* **23** (1973), 48-57.

[11] Salomaa,A., Matrix grammars with a leftmost restriction, *Inform. Contr.* **20** (1972), 143–149.

[12] Thatcher,J.W., Tree automata – an informal survey, in A.V.Aho ed. "Currents in the Theory of Computing", Prentice-Hall, N.J., 1973.