

## Some Restrictions on CFGs With Memory

東京女子大学 (文理) 守屋悦朗 (Etsuro Moriya)

**Abstract.** Context-free grammars with memory was first introduced by the author in [1]. In this note some restricted versions of them are considered. Among them are (left-/right-/meta-)linear context-free grammars with various types of memory. The memory types considered include stack, pushdown stack, stack counter, counter, queue, and double-ended queue.

## § 1 諸定義

著者は先に”メモリ付き”文脈自由文法 (CFG M) と呼ぶ、文脈自由文法の 1 つの拡張を定義し、様々のタイプの CFG M について考察した。また、それらの特徴付けを分割オートマトンと呼ぶオートマトンによって与えた [1]。ここではその CFG M を、基になる文脈自由文法を制限するという観点から考察する。

はじめに、メモリを持つ文脈自由文法 (CFG M) の一般的定義を述べる。CFG M (context-free grammar with memory) は、 $G = (N, \Sigma, \Gamma, P, S, \phi, \$)$  なる 5 項組によって指定される。ここに、 $N$  は非終端記号の有限集合、 $\Sigma$  は終端記号の有限集合、 $\Gamma$  はメモリ記号の有限集合、 $S \in N$  は文記号、 $\phi, \$ \in N \cup \Sigma \cup \Gamma$  はメモリのエンドマークであり、 $P$  は次のいずれかの形のプロダクションよりなる有限集合である：

- ①  $A \rightarrow \alpha$  ( $A \in N; \alpha \in (N \cup \Sigma)^*$ )
- ② (i)  $A \rightarrow Bf$  (ii)  $A \rightarrow fB$  ( $A, B \in N; f \in \Gamma$ )
- ③ (i)  $Af \rightarrow \alpha$  (ii)  $fA \rightarrow \alpha$  ( $A \in N; \alpha \in (N \cup \Sigma)^*$ )
- ④  $Af \rightarrow Bg, Af \rightarrow gB, fA \rightarrow gB, fA \rightarrow Bg$  ( $A, B \in N; f, g \in \Gamma$ )

直観的には、CFG M とは各非終端記号がメモリを持つ CFG のことであり、 $\phi \Gamma^* N \Gamma^* \$$  の元  $\phi \alpha A \beta \$$  は非終端記号  $A$  がメモリ内容  $\alpha \beta$  を持ち、読み書きポインタが  $\alpha$  と  $\beta$  の境目のところを指していることを表す。 $(\phi \Gamma^* N \Gamma^* \$ \cup \Sigma)^*$  の上の 2 項関係  $\Rightarrow$  を次のように定義する。 $\beta, \gamma \in (\phi \Gamma^* N \Gamma^* \$ \cup \Sigma)^*$ ;  $\delta, \varepsilon \in \Gamma^*$ ;  $A_1, \dots, A_k \in N \cup \Sigma$  に対して、

(1) メモリ内容の継承  $A \rightarrow A_1 \dots A_k$  が①型プロダクションのとき、

$$\beta \phi \delta A \varepsilon \$ \gamma \Rightarrow \beta \delta_1 A_1 \varepsilon_1 \dots \delta_k A_k \varepsilon_k \gamma$$

ここに、 $A_1 \in N$  のとき  $\delta_1 = \phi \delta, \varepsilon_1 = \varepsilon \$$ ;  $A_1 \in \Sigma$  のとき  $\delta_1 = \varepsilon_1 = \lambda$

(2) メモリへの書き込み  $A \rightarrow Bf$  (or  $A \rightarrow fB$ ) が②型のプロダクションのとき、

$$\beta \neq \delta A \varepsilon \$ \gamma \Rightarrow \beta \neq \delta Bf \varepsilon \$ \gamma \quad (\text{or } \beta \neq \delta A \varepsilon \$ \gamma \Rightarrow \beta \neq \delta fB \varepsilon \$ \gamma)$$

(3) メモリ内容の部分的消去  $Af \rightarrow A_1 \dots A_k$  (or  $fA \rightarrow A_1 \dots A_k$ ) が③型のプロダクションのとき、

$$\begin{aligned} \beta \neq \delta Af \varepsilon \$ \gamma &\Rightarrow \beta \delta_1 A_1 \varepsilon_1 \dots \delta_k A_k \varepsilon_k \gamma \\ (\text{or } \beta \neq \delta fA \varepsilon \$ \gamma &\Rightarrow \beta \delta_1 A_1 \varepsilon_1 \dots \delta_k A_k \varepsilon_k \gamma) \end{aligned}$$

ここに、 $A_i \in N$  のとき  $\delta_i = \neq \delta$ ,  $\varepsilon_i = \varepsilon \$$ ;  $A_i \in \Sigma$  のとき  $\delta_i = \varepsilon_i = \lambda$

(4) メモリポインタの移動  $\zeta \rightarrow \theta$  が④型のプロダクションのとき、

$$\beta \neq \delta \zeta \varepsilon \$ \gamma \Rightarrow \beta \neq \delta \theta \varepsilon \$ \gamma$$

$\Rightarrow^*$  を  $\Rightarrow$  の反射推移閉包とすると、

$$L(G) = \{ w \in \Sigma^* \mid \neq S \$ \Rightarrow^* w \}$$

を、 $G$  の生成する言語 (CFLM) という。

さて、[1] ではCFG Mの持つメモリ構造にいくつかの制限を加えたものを考えた。

(a) ④型のプロダクションを含まないとき、CFG 2 P (context-free grammar with two pushdown stores) という。

(b) CFG 2 Pで、かつ、 $\Gamma = \{\$, Z\}$ 、かつ、任意の文形式が  $(\neq YZ^*NZ^*\$ \cup \neq NZ^*\$ \cup \neq YZ^*N\$ \cup \neq N\$ \cup \Sigma)^*$  の形をしているようなCFG MをCFG 2 C (context-free grammar with two counters) という。 $\$$ は、カウンタの底を表す。

(c) 次に、②(ii)、③(ii)型のプロダクションを許さず、かつ、④型のプロダクションは次の形のものだけに制限する：

$$Af \rightarrow Bf, Af \rightarrow fB, fA \rightarrow fB, fA \rightarrow Bf$$

さらに、②(i)型プロダクション  $A \rightarrow Bf$ 、③(i)型プロダクション  $Af \rightarrow \alpha$  が適用出来るのは、それぞれ次の形の文形式に限定する：

$$\beta \neq A \delta \$ \gamma \quad \text{または} \quad \beta \neq Af \delta \$ \gamma$$

$$(A, B \in N; f \in \Gamma; \delta \in \Gamma^*; \beta, \gamma \in (\neq \Gamma^* N \Gamma^* \$ \cup \Sigma)^*)$$

このようなCFG MをCFG S (context-free grammar with a stack) という。

(d) CFG Sで、かつ、③型のプロダクションを含まないものをCFG NES (context-free grammar with a nonerasing stack) という。

(e) CFG Sで、かつ、 $\Gamma = \{\$, Z\}$ 、かつ、任意の文形式が  $(\neq NZ^*\$ \cup \neq N\$ \cup \Sigma)^*$  の形をしているようなものをCFG SC (context-free grammar with a stack counter) という。

(f) ④型のプロダクションを含まないCFG SをCFG P (context-free grammar with a pushdown store) という。これは、インデックス文法[3]に他ならない。

(g) CFG Pで、かつ、 $\Gamma = \{\$, Z\}$ 、かつ、任意の文形式が  $(\neq NZ^*\$ \cup \neq N\$ \cup \Sigma)^*$

の形をしているようなものを CFG C (context-free grammar with a counter) という。

X型の文法によって生成される言語のクラスを  $\mathcal{L}_X$  で表すことにする。[1] [2] で次のことが示されている。

- 定理 1 (1)  $\mathcal{L}(\text{CFG2C}) = \mathcal{L}(\text{CFG2P}) = \mathcal{L}(\text{CFGM}) = \text{帰納的可算言語のクラス}$   
 (2)  $\mathcal{L}(\text{CFG}) \subsetneq \mathcal{L}(\text{CFG C}) \subsetneq \mathcal{L}(\text{CFG P}) \subsetneq \mathcal{L}(\text{CFG S}) \subset \mathcal{L}(\text{CSG})$

## § 2 制限付きCFG M

ここでは、CFG Mの基礎になるCFGに制限を加えることを考える。 $G = (N, \Sigma, \Gamma, P, S, \phi, \$)$  をCFG Mとする。 $h: (N \cup \Sigma \cup \Gamma)^* \rightarrow (N \cup \Sigma)^*$  を、 $A \in N \cup \Sigma$  のとき  $h(A) = A$ 、 $f \in \Gamma$  のとき  $h(f) = \lambda$  で定義された準同型写像とし、

$$\underline{P} = \{h(x) \rightarrow h(y) \mid x \rightarrow y \in P\}$$

と定義する。 $\underline{G} = (N, \Sigma, \underline{P}, S)$  を  $G$  の基礎CFG (underlying context-free grammar) という。 $G$  が右線形 (左線形、線形) 文法であるとき、 $G$  は右線形 (左線形、線形) CFG M であるという。

Aho [3] は、CFG Sにおいて①型のプロダクションと③型のプロダクションのタイプ分けを行った。①が $\alpha$ 型、かつ、③が $\beta$ 型であるCFG Sを $\alpha\beta$ 型文法ということにする。従って、右線形CFG P = 右線形右線形CFG Pである。次の結果が既知である [2]。

- 定理 2 (1)  $\mathcal{L}(\text{CFG}) = \mathcal{L}(\text{右線形右線形CFG P}) = \mathcal{L}(\text{左線形左線形CFG P})$   
 (2)  $\mathcal{L}(\text{CFG}) \subsetneq \mathcal{L}(\text{右(左)線形左(右)線形CFG P})$

実際、定理 2 (1) の左側の等式はもっと一般に成り立つ。すなわち、メモリタイプ  $\alpha$  の右線形右線形CFG Sはメモリタイプ  $\alpha$  の非決定性1方向オートマトンと能力が等しい。以下では、非決定性1方向スタックオートマトン [5] を "SA"、非決定性1方向スタック非消去型スタックオートマトン [5] を "NESA"、非決定性1方向1スタックカウンタマシン [4] を "SCA"、非決定性1方向1カウンタマシン [6] を "CA" と略記し、これらが受理する言語のクラスをそれぞれ  $\mathcal{L}(\text{SA})$ ,  $\mathcal{L}(\text{NESA})$ ,  $\mathcal{L}(\text{SCA})$ ,  $\mathcal{L}(\text{CA})$  と書く。

命題 1  $X \in \{S, \text{NES}, \text{SC}, \text{C}\}$  に対して、 $\mathcal{L}(\text{右線形右線形CFG X}) = \mathcal{L}(X\text{A})$ 。

証明  $X=S$  の場合について考える。 $\mathcal{L}(\text{SA}) \subset \mathcal{L}(\text{右線形右線形CFG S})$  の証明は、SAの状

態をCFG Sの非終端記号とし、入力記号の読み込みと状態変化を右線形プロダクションでシミュレートする。SAのスタック読みモード(状態Aでスタック内容 $\beta\gamma$ の $\beta$ と $\gamma$ の境目にポインタがあり、入力記号 $a$ を読んでいるとする)に対応して、CFG Sではスタック内にある非終端記号( $\phi\beta A\gamma\$$ のAに対応する)に①型の右線形プロダクション $A \rightarrow aA$ とそれらに引き続いてスタックポインタの移動に対応する④型のプロダクションを適用すればよい。

$\mathcal{L}$ (右線形右線形CFG S) $\subset \mathcal{L}$ (SA)の証明も同様であるが、CFG Sにおけるスタック読みモード(④型プロダクションの適用)に対しては、SAにおいては $\lambda$ -入力動作が対応する。

$X \in \{NES, SC, C\}$ の場合も全く同様である。

どのプロダクションの右辺にも終端記号が含まれていないようなCFG Sを実時間CFG Sという。realtime CAが $R$ で閉じていない[6]ことから次を得る。

系  $\mathcal{L}$ (実時間右線形右線形CFG S)は $R$ で閉じていない。

補題 1  $X \in \{CFG S, CFGNES, CFGSC, CFGC\}$ に対して、

$$L \in \mathcal{L}(\text{右線形右線形}X) \Leftrightarrow L^R \in \mathcal{L}(\text{左線形左線形}X),$$

$$L \in \mathcal{L}(\text{右線形左線形}X) \Leftrightarrow L^R \in \mathcal{L}(\text{左線形右線形}X).$$

証明 容易。

命題 2  $X \in \{CFG S, CFGP\}$ に対して、 $\mathcal{L}(\text{右線形右線形}X) = \mathcal{L}(\text{左線形左線形}X)$ 。

証明  $L \in \mathcal{L}(\text{右線形右線形}X) \Leftrightarrow L^R \in \mathcal{L}(\text{左線形左線形}X)$

$$\Leftrightarrow L \in \mathcal{L}(\text{左線形左線形}X)$$

前半の $\Leftrightarrow$ は補題1より、後半の $\Leftrightarrow$ はCFL, SAが $R$ で閉じていることによる。

その他のクラスについても、 $R$ で閉じていることが言えれば同様のことが成り立つ。命題2の証明から、

$$\mathcal{L}(\text{左線形右線形}X) \text{ が } R \text{ で閉 } \Leftrightarrow \mathcal{L}(\text{左線形右線形}X) = \mathcal{L}(\text{右線形左線形}X)$$

$$\Leftrightarrow \mathcal{L}(\text{右線形左線形}X) \text{ が } R \text{ で閉}$$

が成り立つことに注意したい。

今のところ、 $X \in \{S, A\}$ について、 $\mathcal{L}(XA)$ と $\mathcal{L}(\text{左線形右線形}CFGX)$ 、 $\mathcal{L}(\text{右線形左線形}CFGX)$ との包含関係については不明であるが、次のことは容易に示すことが出来る。正則集合のクラスを $\mathcal{R}$ で表す。

補題 2  $L_0 = \{a^n \cdot a^n \mid n \geq 0\} \in \mathcal{L}(CFGP)$ 。

証明 次のプロダクションより成るCFGPG = ({S', S, A, A', B}, {a}, {f, g}, P, S,  $\epsilon$ , \$) を考える。

$$\begin{aligned} S' &\rightarrow Sg \\ S &\rightarrow AB \mid \lambda \\ A &\rightarrow A'f, \quad A' \rightarrow Sf \\ Bf &\rightarrow aB, \quad Bg \rightarrow a \end{aligned}$$

明らかに

$$\begin{aligned} S' &\Rightarrow^* Af^{2n-2}gBf^{2n-2}gBf^{2n-4}g \dots Bf^2gBg \\ &\Rightarrow^* Sf^{2n}gBf^{2n-2}gBf^{2n-4}gg \dots Bf^2gBg \quad (n=0, 1, 2, \dots) \\ &\Rightarrow Bf^{2n-2}gBf^{2n-4}g \dots Bf^2gBg \\ &\Rightarrow^* a^{2n-1}a^{2n-3} \dots a^3a \end{aligned}$$

で、 $\sum_i (2i-1) = n^2$  であることに注意すると  $L(G) = L_0$  である。

- 命題 3 (1) (a)  $\mathcal{R} \subseteq \mathcal{L}(\text{左線形右線形CFG}) \cap \mathcal{L}(\text{右線形左線形CFG})$   
 (b)  $\mathcal{L}(\text{左線形右線形CFG}) \cup \mathcal{L}(\text{右線形左線形CFG}) \subseteq \mathcal{L}(\text{CFGP})$   
 (c)  $\mathcal{L}(\text{左線形右線形CFG}) - \mathcal{L}(\text{CA}) \neq \phi$   
 $\mathcal{L}(\text{右線形左線形CFG}) - \mathcal{L}(\text{CA}) \neq \phi$   
 (2)  $\mathcal{L}(\text{CFG}) \subseteq \mathcal{L}(\text{左線形右線形CFGP}) \subseteq \mathcal{L}(\text{CFGP})$   
 $\mathcal{L}(\text{CFG}) \subseteq \mathcal{L}(\text{右線形左線形CFGP}) \subseteq \mathcal{L}(\text{CFGP})$

証明 (1) (a) 右(左)線形文法をシミュレートすればよいので  $\subset$  は明らか。また、 $\{a^n b^n \mid n \geq 0\} \in \mathcal{L}(\text{左線形右線形CFG}) \cap \mathcal{L}(\text{右線形左線形CFG}) - \mathcal{R}$  は容易に示せる。

(b)  $\subset$  は自明。 $\Psi$  を Parikh 写像とすると、 $L \in \mathcal{L}(\text{左線形右線形CFG}) \cup \mathcal{L}(\text{右線形左線形CFG})$  ならば  $\Psi(L)$  は準線形集合(semilinear set)である。なぜなら、例えば  $L \in \mathcal{L}(\text{左線形右線形CFG})$  とするとき、 $L = L(G)$  となる左線形右線形CFGに対して、Gの③型のプロダクション  $Af \rightarrow Ba$  を  $Af \rightarrow aB$  に置き換えた文法  $G'$  を考えると  $G'$  は右線形右線形CFGであり、しかも  $\Psi(L(G)) = \Psi(L(G'))$  である。命題1より  $L(G') \in \mathcal{L}(\text{CA}) \subset \mathcal{L}(\text{CFG})$  であるから、 $\Psi(L(G'))$  は準線形集合である。

よって、補題2より  $\subseteq$  である。

(c)  $L = \{a^i b^j a^i b^j \mid i, j \geq 1\}$  は次のプロダクションを持つ左線形左線形CFGによって生成される。 $S$  は文記号、 $A, A', B, B', C$  は非終端記号、 $f, g, h$  はメモリ記号である。

$$\begin{aligned} S &\rightarrow Ah \\ A &\rightarrow aA' \mid B, \quad A' \rightarrow Af \\ B &\rightarrow bB' \mid C, \quad B' \rightarrow Bg \\ Cf &\rightarrow Ca, \quad Cg \rightarrow Cb, \quad Ch \rightarrow \lambda \end{aligned}$$

補題1より  $L^R \in \mathcal{L}(\text{右線形左線形CFG})$  であり、 $L, L^R$  とともに CFL でないから  $\mathcal{L}(\text{CA})$  の

元ではない。

(2)  $\mathcal{L}(\text{CFG}) \subseteq \mathcal{L}(\text{左線形右線形CFGP})$ 、 $\mathcal{L}(\text{CFG}) \subseteq \mathcal{L}(\text{右線形左線形CFGP})$  は [3] で示されている。 $\mathcal{L}(\text{左線形右線形CFGP}) \subset \mathcal{L}(\text{CFGP})$ 、 $\mathcal{L}(\text{右線形左線形CFGP}) \subset \mathcal{L}(\text{CFGP})$  は自明。これらの包含関係が  $\subseteq$  であることは (1 b) と同様に示される。命題 2 より  $\mathcal{L}(\text{右線形右線形CFGP}) = \mathcal{L}(\text{CFG})$  であることに注意したい。

$\mathcal{L}(\text{左線形右線形CFGs}) \cup \mathcal{L}(\text{右線形左線形CFGs}) \subseteq \mathcal{L}(\text{CFGs})$  か否かは未解決である。次の命題は命題 3 の analogy である (証明方法も)。

命題 4  $\mathcal{L}(\text{CFG}) \subseteq \mathcal{L}(\text{線形CFGP}) \subseteq \mathcal{L}(\text{CFGP})$ 。

$\text{CFG } G = (N, \Sigma, P, S)$  において、どのプロダクションも線形であるか、または、 $S \rightarrow \alpha$  ( $\alpha \in (N \cup \Sigma)^*$ ) の形をしており、かつ、どのプロダクションの右辺にも  $S$  が現われないとき、 $G$  は超線形 [8] であるという。補題 2 の  $L_0$  が超線形 CFGP で生成されていることより、次を得る。

命題 5  $\mathcal{L}(\text{線形CFGP}) \subseteq \mathcal{L}(\text{超線形CFGP})$

$\mathcal{L}(\text{超線形CFGP}) \subseteq \mathcal{L}(\text{CFGP})$  か、 $\mathcal{L}(\text{超線形CFGP}) \subseteq \mathcal{L}(\text{超線形CFGs})$  か否かなどは未解決である。

メモリ構造としては、first-in last-out方式の (プッシュダウン) スタックに対応するものとして first-in first-out方式のキューが考えられる。また、キューの内部を読むことの出来る (書き換えは不可) ようなキューの拡張も考えられる。スタックの場合と異なり、キューの場合にはキューが空であるか否かを判定する能力を持たせるか否かで能力に差が生じるものと思われる。ここでは、空であるか否かを判定する能力を持たないモデルを考える (文法による定義ではこの方が自然である)。キューをメモリ構造として持つ CFGM を  $\text{CFGQ}$  (context-free grammar with a queue) という。また、double-ended queue をメモリ構造として持つ CFGM を  $\text{CFGDEQ}$  という。

命題 6  $X \in \{\text{左線形左線形}, \text{左線形右線形}, \text{右線形左線形}, \text{右線形右線形}\}$  について、 $\mathcal{L}(\text{XCFGQ}) \subset \mathcal{L}(\text{XCFGQ})$ 。特に、 $\mathcal{L}(\text{CFG}) \subseteq \mathcal{L}(\text{CFGQ})$ 。

証明  $\text{XCFGQ}$  においてキューが空か否かを判定出来れば  $C$  が成り立つ。 $\text{XCFGQ}$  が初めにカウンタの底に bottom marker を書き込むのに対応して、 $\text{XCFGQ}$  は非決定性にある時点で end-of-queue marker を書き込んでやることにより、このシミュレーションが可能である。

double-ended queueは

---

fff..ff#ggg...gg

---

の形で2本のカウンタをシミュレート出来る（#は両方のカウンタのbottom markerの役割を果たす）ので、[2]の $\mathcal{L}(\text{CFG2C})$ =帰納的可算言語のクラスの証明より次を得る。

命題7  $\mathcal{L}(\text{CFGDEQ})$ =帰納的可算言語のクラス。

Aho [3]は、右（左）線形右（左）線形以外にもいくつかの制限付CFGによるインデックス文法（=CFG<sub>P</sub>）を考えて興味深い結果を示している。CFG<sub>M</sub>において同様なことを考えてみるのも価値があろう。

謝辞 辛抱強く、この小考察を書く機会を下さった、研究代表者の一橋大学の町田元氏に感謝します。

## 文献

1. 守屋悦朗、ある拡張文法／オートマトンに関するコメント、数理解析研究所講究録695 (1989)、21-26.
2. E.Moriya, Context-free grammars with memory, submitted for publication, 1989.
3. A.V.Aho, Indexed grammars - an extension of context-free grammars, J.ACM 15 (1968), 647-671.
4. R.V.Book and S.Ginsburg, Multi-stack-counter languages, Math. Systems Theory 6 (1972), 37-48.
5. S.Ginsburg, S.A.Greibach and M.A.Harrison, One-way stack automata, J.ACM 14 (1967), 389-418.
6. P.C.Fischer, A.R.Meyer and A.L.Rosenberg, Counter machines and counter languages, Math. Systems Theory 2 (1968), 265-283.
7. J.E.Hopcroft and J.D.Ullman, Nonerasing stack automata, JCSS 1 (1967), 166-186.
8. S.Ginsburg, "The Mathematical Theory of Context-Free Languages," McGraw-Hill, N.Y., 1966.