アルゴリズム 30－13
(1992. 11. 20)

NC^1 に属す線形言語の部分クラスについて

山中 太市郎　藤田 聡　阿江 忠

広島大学工学部第二類 (電気系)

〒724　東広島市鏡山 1-4-1

あらまし　NC^1に属す線形言語 (LIN) の部分クラスとして準決定性線形言語 (QDLIN) を提示する。今までに NC^1に属すことが判明している CFL の部分クラスが決定性機械による特徴づけや言語の無あいまい性のような性質を有するのに対し、QDLIN のクラスには本質的にあいまいな言語が存在するなど、より非決定性に近いクラスであることを示す性質を持つ。QDLIN の概念は超線形言語 (MLIN) にまで拡張され、準決定性超線形言語 (QDMLIN) と呼ぶ MLIN の部分クラスの言語に対しても NC^1に属すことが明らかになる。また、QDLIN は CREW PRAM により $O(\log n)$ 時間、n プロセッサで認識可能であることも示される。

和文キーワード　並列計算, 線形言語, NC^1, 非決定性, ATM, PRAM

On Subclasses of Linear Languages in NC^1

Taichiro YAMANAKA, Satoshi FUJITA and Tadashi AE

Electrical Engineering, Faculty of Engineering, Hiroshima University

Kagamiyama 1-4-1, Higashi-Hiroshima, 724 Japan

Abstract　We present the quasi-deterministic linear languages (QDLIN, for short) that is a new subclass of linear languages (LIN) in NC^1. Several subclasses of CFLs in NC^1 have been characterized by the deterministic machines, and then, have some "deterministic" properties (cf. unambiguity). However, there exist inherently ambiguous languages in the class of QDLINs, and then, QDLINs have somewhat non-deterministic properties. The definition of QDLIN is extended to the meta-linear languages (MLIN), and the quasi-deterministic meta-linear languages (QDMLIN) of our definition are in NC^1. We also show that QDLINs can be recognized in $O(\log n)$ time using n processors on a CREW PRAM.

英文 key words　parallel computation, linear language, NC^1, non-determinism, ATM, PRAM

1 まえがき

Pippenger が導入したクラス NC は“効率のよい”並列計算を行なうことができる問題のクラスとして一般に理解されている [9][12]。ここでいう“効率のよい”並列計算とは、並列計算機上で多項式個のプロセッサを用いて $\log n$ の多項式で押えられる時間で実行可能な計算のことを指す。

クラス NC は多項式サイズで $O(\log^k n)$ の深さの一様論理回路族によって受理される言語のクラスを NC^kと表すとき、$\mathrm{NC} = \cup_{k\geq 1}\mathrm{NC}^k$によって定義されるが、Ruzzo [14] により交代チューリング機械 (以下、ATM) [2] による NC の特徴づけがなされたことから、NC への所属に関する議論に ATM が用いられるようになった。

ところで、NC と形式言語のクラスとの対応を見たとき (表 1 参照)、文脈自由言語 (CFL) が NC^2に属すことが知られているが [14]、CFL が NC^1に属すかどうかは未解決問題である。

一方、これをより実際的な並列計算の立場から見れば、並列計算のモデルとして通常用いられている並列ランダムアクセス機械 (PRAM) [4] においては、CREW PRAM で $O(\log^2 n)$ 時間、n^6プロセッサで CFL が認識可能であることは判明している [5]。

CFL が NC^1に属すかどうかはそれ自体興味深い問題ではあるが、否定的とする予想も多い。そこで、“CFL の部分クラスのうちで NC^1に属す最大のクラスは何か”というのもごく自然な疑問であり、実際これまでに NC^1に属す CFL の部分クラスがいくつか知られている。Ibarra ら [8] は k文字上の one-sided Dyck 言語 D_k [6]、structured CFL (SCFL) [13]、bracketed CFL (BCFL) [6]および決定性線形文脈自由言語 (DLCFL) が NC^1に属すことを示した [8]。ここで DLCFL とは、文献 [8] において定義された決定性線形文脈自由文法 (DLCFG) によって生成される言語で、DLCFG は $A \to x_1B_1y_1$, $A \to x_2B_2y_2$, $A \to x_3$なる書き換え規則があったとき、“$i \neq j$に対し x_iは x_jの接頭語ではない”という性質を持つ。ただし A, B_1, B_2は非終端記号、x_1, x_2, x_3, y_1, y_2は (空列も許す) 終端記号列である。また Dymond は BCFL を真に含む input-driven 言語 (ID) が NC^1に属すことを示している [3]。

これに対し前述の PRAM では、BCFL および ID に対する CREW PRAM 上での $O(\log n)$ 時間、$n/\log n$ プロセッサで動作する最適並列認識アルゴリズムが知られている [5][16]。その他のクラスでは決定性 CFL (DCFL) が CREW PRAM で $O(\log n)$ 時間、n^3プロセッサで [10]、無あいまいな CFL が同じく CREW PRAM で $O(\log n)$ 時間、n^7プロセッサで [15] 認識可能であることが知られているがこれらの言語に対しては (ATM などによる) NC^1に属すという厳密な証明は (著者らの知る限り) なされていない。

一方、線形言語 (LIN) は NC^1に属すかどうかわかっていない有名な CFL の部分クラスの一つである。また、同時書き込みを許さないタイプの PRAM においても $O(\log n)$ 時間で認識するアルゴリズムは発見されていない。ただし、$O(\log^2 n)$ の時間では、EREW PRAM 上で n^3プロセッサを使って認識可能である [17]。

前述の DLCFL は LIN の部分クラスであるが、DLCFG の性質を考慮すると、DLCFL のクラスには LIN のなかでもかなり特殊な言語しか属さない (つまり狭いクラス) ということが予想される。実際、DLCFL のクラスが 1 ターン DPDA のクラスと一致するということが明らかになることも示される。

本研究は、NC^1に属す LIN のより広い部分クラスを探り出すことを目的としている。そのために、まず **2** で準決定性線形言語 (quasi-deterministic linear language, 以下 QDLIN) と呼ぶ LIN の部分クラスを文法的に定義し、他の CFL の部分クラスとの関係について議論する。**3** で有限オートマトン的な立場も含めて、QDLIN の持つ一般的な性質について議論した後、**4** で QDLIN のクラスが NC^1に属すことを示す。さらに **4** ではこの部分クラスを認識するための PRAM 上で動作する並列アルゴリズムについても説明する。最後に **5** で LIN が NC^1に属す可能性についてふれる。

2 準決定性線形言語

この章では、LIN の部分クラスとなる準決定性線形言語 (QDLIN) を新たに定義し、他の CFL の部分クラスとの関係について議論する。

2.1 定義

まず、線形文法 (LG) の定義から始める。

定義 1 線形文法 G は 4 項組

$$G = (V_N, V_T, S, P)$$

である。ここで、

1. V_Nは非終端記号の有限集合,
2. V_Tは終端記号の有限集合で $V_N \cap V_T = \emptyset$,
3. Sは開始記号で $S \in V_N$,
4. Pは生成規則の有限集合で、その要素は次の 3つのいずれかの形で書かれている

$$A \to aB, A \to Ba \text{ または } A \to a$$

ただし、$A, B \in V_N, a \in V_T \cup \{\varepsilon\}$ である†。 □

LG によって生成される言語を線形言語 (LIN) という。以後、言語 X のクラスを$\mathcal{L}$(X) と書くことにする。例えば、$\mathcal{L}$(LIN) は LIN のクラスを表す。次に準決定性線形文法 (QDLG) を以下のように定義する。

定義 2 線形文法 $G = (V_N, V_T, S, P)$ は以下に示す 3 つの条件

1. 開始記号 Sはどの規則の右辺にも現れない。

†εは空語を表す。

表 1: CFL とその部分クラスの並列計算量

言語	文献	NC^k	PRAM タイプ	PRAM 時間	PRAM プロセッサ
bracketed CFL	Rytter and Giancarlo, 1987 [16]		CREW	$O(\log n)$	$n/\log n$
	Ibarra et al., 1988 [8]	NC^1 (ATM)			
one-sided Dyck	Ibarra et al., 1988 [8]	NC^1 (ATM)			
structured CFL	Ibarra et al., 1988 [8]	NC^1 (ATM)			
input-driven	Gibbons and Rytter, 1988 [5]		CREW	$O(\log n)$	$n/\log n$
	Dymond, 1988 [3]	NC^1 (uniform circuit)			
derterministic linear CFL [8]	Ibarra et al., 1988 [8]	NC^1 (ATM)			
DCFL	Klein and Reif, 1988 [10]	*	CREW	$O(\log n)$	n^3
unambiguaus CFL	Rytter, 1987 [15]	*	CREW	$O(\log n)$	n^7
CFL	Ruzzo, 1981 [14]	NC^2 (ATM) *			
	Gibbons and Rytter, 1988 [5]		CREW	$O(\log^2 n)$	n^6
linear	Rytter, 1988 [17]	*	EREW	$O(\log^2 n)$	n^3
quasi-deterministic linear	本研究	NC^1	CREW	$O(\log n)$	n

*は NC^1 に属すことが未解決であることを表す。

2. $A \in V_N - \{S\}$ であるような任意の $A \to aB \in P$ に対して、
$A \Rightarrow aB \Rightarrow \cdots \Rightarrow aY_1\alpha \Rightarrow abX_1\alpha$
$A \Rightarrow aB \Rightarrow \cdots \Rightarrow aY_2\beta \Rightarrow abX_2\beta$
ならば、$X_1 = X_2$。ただし、$A, B, X_1, X_2, Y_1, Y_2 \in V_N$, $a, b \in V_T$, $\alpha, \beta \in V_T^*$。

3. $A \in V_N - \{S\}$ であるような任意の $A \to Ba \in P$ に対して、
$A \Rightarrow Ba \Rightarrow \cdots \Rightarrow \alpha Y_1 a \Rightarrow \alpha X_1 ba$
$A \Rightarrow Ba \Rightarrow \cdots \Rightarrow \beta Y_2 a \Rightarrow \beta X_2 ba$
ならば、$X_1 = X_2$。ただし、$A, B, X_1, X_2, Y_1, Y_2 \in V_N$, $a, b \in V_T$, $\alpha, \beta \in V_T^*$。

を満たすとき準決定性線形文法 *(QDLG)* であるという。 □

QDLG は

1. $A \to aB_1, A \to aB_2 \in P$ ならば $B_1 = B_2$
2. $A \to B_1 a, A \to B_2 a \in P$ ならば $B_1 = B_2$
ただし、$A, B_1, B_2 \in V_N - \{S\}$

の性質を満たすことに注意したい。QDLG によって生成される言語を準決定性線形言語 (QDLIN) という。

2.2 他の CFL の部分クラスとの関係

ここでは、他の CFL の部分クラス、主として NC^1 に属すことが知られているクラス (表 1 参照) との関係について調べる。まず bracketed CFL [6] (以下 BCFL)、structured CFL [13] (SCFL) および input-driven 言語 [3] (ID) の定義を以下に示す。

定義 3 文脈自由文法 *(CFG)* $G = (V_N, V_T, S, P)$ は、

1. 記号 "(" および ")" が V_T にあり、

2. 各規則は

$$A \to (\alpha) \qquad \text{ただし}\ \alpha \in (V_N \cup V_T - \{(,)\})^*$$

の形をしている

とき、*bracketed CFG (BCFG)* であるという。*BCFG* によって生成される言語を *bracketed CFL (BCFL)* と呼ぶ。 □

定義 4 $G = (V_N, V_T, S, P)$ を任意の *CFG* とする。G によって誘導される *structured CFG (SCFG)* G' は
$G' = (V_N, V'_T, S, P')$ である。ここで

1. $V'_T = V_T \cup \{[_A,]_A | A \in V_N\}$

2. $P' = \{A \to [_A\alpha]_A | A \to \alpha$ は P の規則 $\}$

である。*SCFL* によって生成される言語を *structured CFL (SCFL)* と呼ぶ。 □

定義 5 *CFL* L は L を受理する、あるプッシュダウンオートマトン *(PDA)* M が存在して M のスタックの動作 *(プッシュあるいはポップ)* が計算状況の状態およびスタック記号によらず、入力記号のみから決定されるとき、*input-driven (ID)* であるという。 □

定義より明らかなように、BCFL と SCFL は本質的には同等の性質を持つとして、以後の議論では BCFL のみを扱うことにする。BCFL, ID および決定性 CFL (DCFL) のクラス間には次のような包含関係が成り立つことが知られている [3]。

定理 1 (文献 [3]) *$\mathcal{L}$(BCFL)$\subsetneq\mathcal{L}$(ID)$\subsetneq\mathcal{L}$(DCFL)* □

次に、文献 [11] において定義された LIN の興味深い部分クラスである単純線形言語 (以下、SLIN) を QDLIN との比較のために導入する。

定義 6 (松田 [11]) *DLG $G=(V_N,V_T,S,P)$ は、*

1. $V_N=V_1\cup V_2,\ V_1\cap V_2=\emptyset$ で、

2. $P\subseteq[V_1\times(V_TV_N\cup V_T\cup\{\varepsilon\})]\cup[V_2\times(V_NV_T\cup V_T\cup\{\varepsilon\})]$

を満たすような V_1,V_2が存在し、さらに、

3. $A\rightarrow u, A\rightarrow v\in P$ならば $h(u)\neq h(v)$ あるいは $u=v$

のとき、単純線形文法 *(SLG)* であるという。ただし h は、

$$h(\alpha)=\begin{cases}\varepsilon & \text{for } \alpha\in V_N\\ \alpha & \text{for } \alpha\in V_T\end{cases}$$

なる準同形写像とする。*SLG* によって生成される言語を単純線形言語 *(SLIN)* という。 □

クラス$\mathcal{L}$(SLIN) は、そのどの言語も無あいまいであるという重要な性質を持つ [11](それゆえ CREW PRAM で $O(\log n)$ 時間で認識可能である [15])。LIN は受理機械との対応でいうと、1 ターン PDA [6][7]のクラスと一致するが、もう一方で 2 テープ (非決定性) オートマトン (略して 2-NFA) [6][11]による特徴づけもなされている。SLIN に対しては、決定性 2 テープオートマトン (2-DFA) による以下のような特徴づけが知られている。

定理 2 (松田) *言語 L が SLIN であるための必要十分条件は $L=\widetilde{T}(A)$ なる 2-DFA A が存在することである。ただし、$\widetilde{T}(A)=\{xy^R|(x,y)\in T(A)\}$。*†† □

X∈ {BCFL, ID, DCFL} と QDLIN の関係を明らかにする前に、X∈ {BCFL, ID, DCFL} と SLIN の関係を調べる。

定理 3 *$\mathcal{L}$(SLIN) は$\mathcal{L}$(X), X∈ {BCFL, ID, DCFL} と比較不能である。*

(証明) $L_1=\{a^nb^n|n\geq 0\}$, $L_2=\{ww^R|w\in\{a,b\}^*\}$, $L_3=\{a^{m+1}b^ma^nb^{n+1}|m,n\geq 1\}$ とすると、
(1) $L_1\in\mathcal{L}$(BCFL) かつ文献 [11] より $L_1\in\mathcal{L}$(SLIN)
(2) $L_2\in\mathcal{L}$(SLIN) であるが $L_2\notin\mathcal{L}$(DCFL)[11]
(3) $L_3\in\mathcal{L}$(BCFL) であるが LIN に対する反復補題より $L_3\notin\mathcal{L}$(LIN)
となることからいえる。 □

定理 4 *$\mathcal{L}$(SLIN)$\subsetneq\mathcal{L}$(QDLIN)*

††y^Rは yの反転を表す。

(証明) $\mathcal{L}$(SLIN)$\subseteq\mathcal{L}$(QDLIN) なることは SLIN の無あいまい性が QDLG であるための条件を満たす性質であることより明らか。

$\mathcal{L}$(SLIN)$\neq\mathcal{L}$(QDLIN) なることは、
$L_4=\{a^ib^jc^k|i=j \text{ or } j=k\}$ としたとき、L_4が QDLG
$G=(\{S,A,A',X,B,B',Y\},\{a,b,c\},S,P)$

$$\begin{array}{llll} P=\{ & S\rightarrow aA, & S\rightarrow Bc, & S\rightarrow\varepsilon,\\ & A\rightarrow aA, & B\rightarrow Bc, & A\rightarrow\varepsilon,\\ & A\rightarrow bX, & B\rightarrow aY, & B\rightarrow\varepsilon,\\ & A'\rightarrow bX, & B'\rightarrow aY, & \\ & X\rightarrow A'c, & Y\rightarrow B'b, & \\ & X\rightarrow c, & Y\rightarrow b, & \} \end{array}$$

に対して $L_4=L(G)$ なること、および L_4が本質的にあいまいであることからいえる。 □

定理 3 および定理 4 より直ちに次の結果を得る。

系 4.1 $\mathcal{L}$(QDLIN) は$\mathcal{L}$(X), X∈ {BCFL, ID, DCFL} と比較不能である。 □

本節の最後として、Ibarra らが導入した決定性線形 CFL (DLCFL) [8] と QDLIN との関係について調べる。表 1 に示したように、DLCFL は NC^1に属す。DLCFL の定義を以下に示す。

定義 7 *CFG $G=(V_N,V_T,S,P)$ は*

1. $P=\{A\rightarrow xBy|A,B\in V_N,\ x,y\in V_T^*\}$
$\cup\{A\rightarrow x|A\in V_N,\ x\in V_T^*\}$

2. 規則 $A\rightarrow x_1By_1$, $A\rightarrow x_2By_2$および $A\rightarrow x_3$ があるならば、x_iは x_jの接頭語とならない *(ただし $i\neq j$)*

のとき、決定性線形 *CFG (DLCFG)* であるという。*DLCFG* によって生成される言語を決定性線形 *CFL (DLCFL)* という。 □

DLCFL は PDA により以下のように特徴づけられる。

定理 5 *$\mathcal{L}$(DLCFL)=$\mathcal{L}$(DCFL)$\cap\mathcal{L}$(LIN)*

(証明) (1) $\mathcal{L}$(DLCFL)$\subseteq\mathcal{L}$(DCFL)$\cap\mathcal{L}$(LIN) なること。
DLCFG $G=(V_N,V_T,S,P)$ に対して、一般性を失うことなく $A\rightarrow xBy, A\rightarrow z\in P$ ならば $|x|,|y|,|z|\leq 1$ とする。このとき PDA $M=(Q,\Sigma,\Gamma,\delta,q_0,Z_0,F)$ を次のように構成する。
1. $Q=V_N,\ \Sigma=V_T,\ q_0=S,\ F=\{S\}$,
$\Gamma=\{[Aa]|A\rightarrow aBb\in P,\ A,B\in V_N,$
$a,b\in V_T\cup\{\varepsilon\}\}\cup\{Z_0\}$,
2. 関数δは
$A\rightarrow aBb\in P$ならば、任意の $Z\in\Gamma$に対して
$\delta(A,a,Z)=(B,[Aa]Z)$ ($[Aa]$ をプッシュする)
$\delta(B,b,[Aa])=(A,\varepsilon)$ ($[Aa]$ をポップする)
$A\rightarrow a\in P$ならば、任意の $Z\in\Gamma$に対して
$\delta(A,a,Z)=(A,Z)$ (スタック、状態は変化しない)
ただし $A,B\in V_N,\ a,b\in V_T\cup\{\varepsilon\}$。このように Mを構成すると $T(M)=L(G)$ となるのは明らかであり、しかも Gの性質から Mは DPDA である。
(2) $\mathcal{L}$(DLCFL)$\supseteq\mathcal{L}$(DCFL)$\cap\mathcal{L}$(LIN) なること。
逆に空スタックで受理する 1 ターン DPDA

$M=(Q,\Sigma,\Gamma,\delta,q_0,Z_0,\emptyset)$ から以下に示す標準的な方法 (例えば、文献 [6] 参照) により LG $G=(\{S\}\cup[Q\times\Gamma\times Q\times(\Gamma\cup\{\varepsilon\})],\Sigma,S,P)$ を構成する。Pの規則は

1. 各 $q'\in Q$ に対して　$S\to[q_0,Z_0,q',\varepsilon]$,
2. $\delta(q,a,Z)=(q',\varepsilon)$ ならば $[q,Z,q',\varepsilon]\to a$
3. $\delta(q,a,Z)=(q',Y)$ ならば、任意の $X\in\Gamma\cup\{\varepsilon\}$ および $q''\in Q$ に対して
   $[q,Z,q'',X]\to a[q',Y,q'',X]$
   $[q'',X,q',Y]\to[q'',X,q,Z]a$
4. $\delta(q,a,Z)=(q',YZ'),\ Z'\neq\varepsilon$ならば、任意の $q''\in Q$ に対して
   $[q,Z,q'',Z']\to a[q',Y,q'',\varepsilon]$
5. $\delta(q,a,Z)=(q',\varepsilon)$ ならば、任意の $Z'\in\Gamma$および $q''\in Q$ に対して
   $[q'',Z',q',\varepsilon]\to[q'',Z',q,Z]a$

この構成法は文献 [6] より $L(G)=T(M)$ であることが保証されている。2 と 3 に注目すると、$q''=q'$かつ $X=\varepsilon$のときそれぞれ $[q,Z,q',\varepsilon]\to a$, $[q,Z,q',\varepsilon]\to a[q',Y,q',\varepsilon]$ となるが、DPDA の定義より$\delta(q,a,Z)$ は一意に決まるのでこの 2 つの規則は同時に存在しない。同様に 3 と 4 に注目すると、$X=Z'$のとき任意の q''に対して $[q,Z,q'',Z']\to a[q',Y,q'',Z']$, $[q,Z,q'',Z']\to a[q',Y,q'',\varepsilon]$ となるが、$\delta(q,a,Z)$ は一意に決まるので各 3 項組 (q,a,Z) に対して 3 と 4 は同時に成立しない。したがって G は DLCFG であることが確かめられる。　□

系 5.1 $\mathcal{L}$(QDLIN)$\neq\mathcal{L}$(DCDFL)　□

以上の結果をまとめて図 1 に示す。

3　準決定性線形言語の性質

この章では若干ではあるが、QDLIN の持つ重要な性質について述べる。3.1 および 3.2 は SLIN との対比において、さらに 3.3 は並列認識において重要となる。

3.1　あいまい性

定理 4 より次の系を得る。

系 4.2 $\mathcal{L}$(QDLIN) には本質的にあいまいな言語が存在する。　□

前章で述べたように SLIN のクラスはあいまいではない [11]。また NC^{1}に属することが知られている ID, DLCFL などのクラスはいずれも DCFL の部分クラスでこれらもやはりあいまいではない。つまり、現在までに知られている NC^{1}に属す言語は、いずれも決定性に近い性質を備えている。

3.2　閉包性

定理 6 *$\mathcal{L}$(QDLIN) は有限個の集合和のもとで閉じている。*

(証明)　正整数 kに対して、$G_1,G_2,\dots,G_k$を QDLG とする。ただし、$G_i=(V_{Ni},V_{Ti},S_i,T_i)$。$k=1$ のときは $L(G_1)$ が QDLIN であることは自明。したがって $k\geq 2$ を仮定する。一般性を失うことなく $\bigcap_{i=1}^{k}V_{Ni}=\emptyset$ とする。このとき、各 P_iの $S_i\to\alpha$なる規則の左辺の S_iを新たな非終端記号 S と置き換えて、

$$G=(\{S\}\cup\bigcup_{i=1}^{k}(V_{Ni}-\{S_i\}),\bigcup_{i=1}^{k}V_{Ti},S,P),$$

$$P=\{A\to\alpha|A\to\alpha\in\bigcup_{i=1}^{k}P_i,\ A\in\bigcup_{i=1}^{k}(V_{Ni}-\{S_i\})\}\cup\{S\to\alpha|S_i\to\alpha\in P_i,\ 1\leq i\leq k\}$$

なる QDLG を構成すると、明らかに

$L(G)=\bigcup_{i=1}^{k}L(G_i)$。　□

$\mathcal{L}$(SLIN) は (有限個の) 集合和のもとで閉じていないことが知られている [11]。図 1の L_6がそのような例で [1]、従って、QDLIN が SLIN よりクラスとして大きいことを示す例にもなっている。

3.3　正規集合に基づく性質

ここで述べる QDLIN の性質は、4 章で述べる ATM での認識において非常に重要な役割を果たすことになる。

補題 1 *A を QDLG $G=(V_N,V_T,S,P)$ の開始記号 S以外の非終端記号とする。このとき A に対して以下の 2つが成立する。*

1. $A\overset{*}{\Rightarrow}xXy_1\Rightarrow xaB_1y_1$
   $A\overset{*}{\Rightarrow}xYy_2\Rightarrow xaB_2y_2$
   ただし、$B_1,B_2,X,Y\in V_N,\ a\in V_T,\ x,y_1,y_2\in V_T^*$
   ならば、　$B_1=B_2$。
2. $A\overset{*}{\Rightarrow}x_1Xy\Rightarrow x_1B_1by$
   $A\overset{*}{\Rightarrow}x_2Yy\Rightarrow x_2B_2by$
   ただし、$B_1,B_2,X,Y\in V_N,\ b\in V_T,\ y,x_1,x_2\in V_T^*$
   ならば、　$B_1=B_2$。

(証明)　1. のみ証明する。2. も全く同様にして証明できる。証明は記号列 xa の長さ $|xa|=k$に関する帰納法による。

始めに $k=1$ のときは定義より明らか。次に $k\ (k\geq 1)$ のとき成り立っていると仮定する。このとき $k+1$ の場合を考える。帰納法の仮定により、$|x|=k$として

$A\overset{*}{\Rightarrow}x'X'z_1\Rightarrow x'a'B_1'z_1=xB_1'z_1$
$A\overset{*}{\Rightarrow}x'Y'z_2\Rightarrow x'a'B_2'z_2=xB_2'z_2$

ならば　$B_1'=B_2'$。$y_1=z_1'z_1,\ y_2=z_2'z_2, B'=B_1'=B_2'$ とすると、G の定義より

$B'\overset{*}{\Rightarrow}Xz_1'\Rightarrow aB_1z_1'$
$B'\overset{*}{\Rightarrow}Yz_2'\Rightarrow aB_2z_2'$

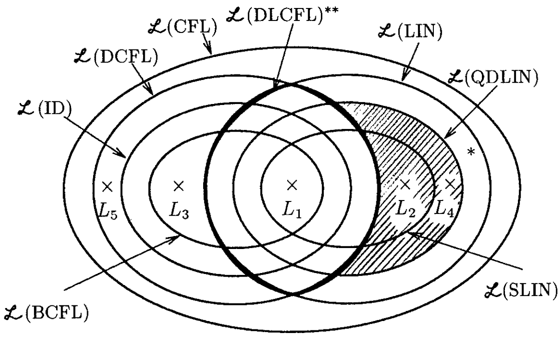
ならば　$B_1=B_2$であるから、

$A\overset{*}{\Rightarrow}xB_1'z_1\overset{*}{\Rightarrow}xXz_1'z_1\Rightarrow xaB_1z_1'z_1$
$A\overset{*}{\Rightarrow}xB_2'z_2\overset{*}{\Rightarrow}xYz_2'z_2\Rightarrow xaB_2z_2'z_2$

ならば $B_1=B_2$となり、$k+1$ の場合も成立する。　□

定義 8 *$G=(V_N,V_T,S,P)$ を QDLG とする。G より誘導された右線形文法 (RLG) $G_R=(V_N,V_T,S,P_R)$ とは、*

1. $P_R\supseteq P$で、



$L_1 = \{a^n b^n | n \geq 0\}$,
$L_2 = \{ww^R | w \in \{a,b\}^*\}$,
$L_3 = \{a^{m+1} b^m a^n b^{n+1} | m, n \geq 1\}$,
$L_4 = \{a^i b^j c^k | i = j \text{ or } j = k\}$,
L_5: two-sided Dyck language over four letters [3][6]

$*\mathcal{L}(\mathrm{QDLIN}) \subsetneq \mathcal{L}(\mathrm{LIN})$ かは不明。
$**\mathcal{L}(\mathrm{DLCFL})$ 内の階層が存在するかは不明。しかし、
$L_6 = \{a^n b^n | n \geq 0\} \cup \{a^n c b^{2n} | n \geq 0\}$
に対して、
$L_6 \in (\mathcal{L}(\mathrm{QDLIN}) - \mathcal{L}(\mathrm{SLIN})) \cap \mathcal{L}(\mathrm{DCFL})$ であることはわかっている。

図 1: QDLIN と CFL の部分クラスの関係

2. $A \to aB \in P,\ A, B \in V_N - \{S\},\ a \in V_T$に対して
$A \underset{G}{\Rightarrow} aB \overset{*}{\underset{G}{\Rightarrow}} aY\alpha \underset{G}{\Rightarrow} abX\alpha$ または
$A \underset{G}{\Rightarrow} aB \overset{*}{\underset{G}{\Rightarrow}} aY\alpha \underset{G}{\Rightarrow} ab\alpha$ のとき、
それぞれ $B \to bX \in P_R,\ B \to b \in P_R$

を満たす文法である。同様に G より誘導された左線形文法 (LLG) $G_L = (V_N, V_T, S, P_L)$ とは

1. $P_L \supseteq P$で、

2. $A \to Ba \in P,\ A, B \in V_N - \{S\},\ a \in V_T$に対して
$A \underset{G}{\Rightarrow} Ba \overset{*}{\underset{G}{\Rightarrow}} \alpha Ya \underset{G}{\Rightarrow} \alpha Xba$ または
$A \underset{G}{\Rightarrow} Ba \overset{*}{\underset{G}{\Rightarrow}} \alpha Ya \underset{G}{\Rightarrow} \alpha ba$ のとき、
それぞれ $B \to Xb \in P_L,\ B \to b \in P_L$

を満たす文法である。 □

これらの文法 G_R, G_Lは元の G の性質からわかるように、有限オートマトン的な見方をすると、入力テープ上の記号を読んだときの動作は最初の記号を読んだとき、および最後の記号を読んだときの状態遷移は非決定的に選択されるが、その間の記号を読んだときの状態遷移は決定性 (つまり一意に決まる) という性質を持つ。ここで最後の記号を読んだときの状態遷移が非決定的であるというのはその時点で受理状態ではない遷移先があるということを意味する。

補題 2 *QDLG* $G = (V_N, V_T, S, P)$ より誘導された *RLG(LLG)* を $G_R(G_L)$ とする。このとき、$A \in V_N - \{S\}$ に対して

1. $A \underset{G}{\Rightarrow} aX \overset{*}{\underset{G}{\Rightarrow}} x'Y \underset{G}{\Rightarrow} x'Zb \overset{*}{\underset{G}{\Rightarrow}} xBy \underset{G}{\Rightarrow} xcy$
$y,\ x = x'x'' \in V_T^+,\ x' = ax_1,\ a, c \in V_T$
のときかつそのときに限り
$A \overset{*}{\underset{G_R}{\Rightarrow}} xc$ かつ $Y \overset{*}{\underset{G_L}{\Rightarrow}} cy$

2. $A \underset{G}{\Rightarrow} Xb \overset{*}{\underset{G}{\Rightarrow}} Yy' \underset{G}{\Rightarrow} aZy' \overset{*}{\underset{G}{\Rightarrow}} xBy \underset{G}{\Rightarrow} xcy$
$x,\ y = y''y' \in V_T^+,\ y' = y_1 b,\ b, c \in V_T$
のときかつそのときに限り
$A \overset{*}{\underset{G_L}{\Rightarrow}} cy$ かつ $Y \overset{*}{\underset{G_R}{\Rightarrow}} xc$

(証明) 1. は (a) $A \underset{G}{\Rightarrow} aX \overset{*}{\underset{G}{\Rightarrow}} x'Y \underset{G}{\Rightarrow} x'Zb \overset{*}{\underset{G}{\Rightarrow}} xBy \underset{G}{\Rightarrow} xcy$ なる導出があるときは、G_Rおよび G_Lで $A \overset{*}{\underset{G_L}{\Rightarrow}} cy$, $Y \overset{*}{\underset{G_R}{\Rightarrow}} xc$ によって表される導出がかならず存在し、(b) しかもこの形の導出は (G_R, G_L) において唯一つしかないことをいえばよいが、(a) は G_R, G_L の作り方から、(b) は G の定義および補題 1 から明らかである。2. も全く同様。 □

補題 2 は図 2 に示すように、G における開始記号 S 以外の非終端記号 A を根とする導出木 Tに対して、左の葉の部分と右の葉の部分とに分離された導出木 T', T'' がそれぞれ G_R, G_Lの導出木となり、逆に、一番最後に導出される葉が等しい G_R, G_Lの (根が Sでない) 導出木 T', T'' に対して根の部分の結合がうまくできるかどうか (図 2 の例では $A \overset{*}{\underset{G}{\Rightarrow}} x'Y$かどうか) を調べるだけで 2 つの導出木が結合できるかどうかが判断できるということを示している。ここで注意しなければならないのは、Tと (T', T'') の関係である。というのも Tに対して (T', T'') は一意に決まるが、(T', T'') に対して Tは一般的に一意に決まらないからである (図 3 参照)。

4 準決定性線形言語の並列認識

この章では、4.1 で indexing ATM を用いて QDLIN が NC^1に属すことを示し、4.2 で QDLIN の CREW PRAM における $O(\log n)$ 時間並列認識アルゴリズムを示す。

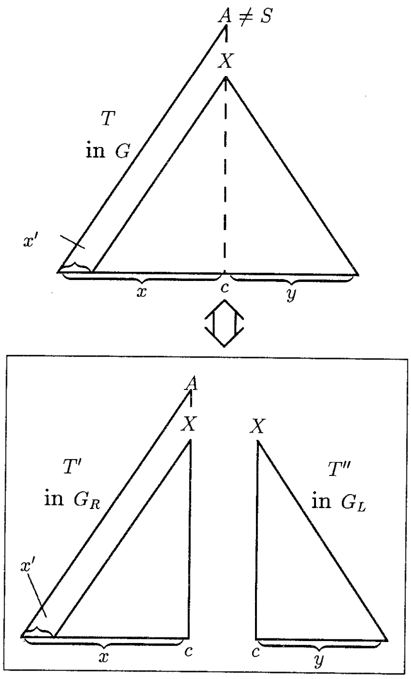


図 2: QDLG G と G より誘導された G_R, G_Lの関係

4.1 indexing ATM による認識

ここではまずクラス NC^kおよび NC の定義から始める。

定義 9 (**Ruzzo**[14]) サイズおよび深さ複雑度が $Z(n), T(n)$ $(n = 1, 2, \ldots)$ である論理回路族 $C = (c_1, c_2, \ldots)$ は、c_n の標準的符合化[14]が DSPACE($\log Z(n)$)で計算可能であるとき U_{BC}一様であるという。サイズおよび深さ複雑度が $O(Z(n))$, $O(T(n))$ である U_{BC}一様論理回路によって認識可能な言語のクラスを U_{BC}-SIZE,DEPTH($(Z(n), T(n))$ により表す。 □

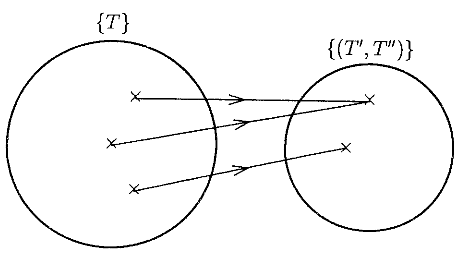


図 3: 空間 $\{T\}$ から空間 $\{(T', T'')\}$ への写像

定義 10

$$\begin{aligned} \mathrm{NC}^k &= U_{BC}\text{-SIZE,DEPTH}(n^{O(1)}, \log^k n) \\ \mathrm{NC} &= \bigcup_{k \geq 1} \mathrm{NC}^k \end{aligned}$$ □

回路族の "一様性" についてはいくつかの異なる定義が提案されており [14]、それらのどの定義を用いても NC^kは $k \geq 2$ の場合には変わらないことが知られている [14]。

交代チューリング機械 (ATM) [2] は非決定性チューリング機械 (NTM) を一般化した計算モデルで、各状態には全称状態か存在状態かを示す記号 "$\forall$","$\exists$" が割り当てられている。ATM の計算はノードを計算状況とし、根を初期計算状況とする計算木で表現される。ある計算状況の状態が全称状態ならば、それに続くすべての計算状況に遷移し、存在状態ならばそれに続く一つの計算状況を非決定的に選択し、それに遷移する。ATM Mが入力 x を受理するとは、Mにおいて葉がすべて受理計算状況であるような x の計算木が存在するということである。

さらに、ここでは ATM の入力に "ランダムアクセス" するために、ATM の代わりに indexing ATM [2][14] (以下、iATM と略す) を用いる。iATM は入力テープを読むために、他と区別された "読み取り状態" と呼ばれる停止状態とインデックステープと呼ばれる特別な作業用テープを持つ。読み取り状態に入るとある入力アルファベット a とインデックステープの内容を 2 進表現とする整数 i を guess し、a が入力テープの先頭から i 番目の記号であれば受理する (すなわち入力テープの i 番目の記号を読む) ことにする。

$O(T(n))$ 時間限定 (i)ATM によって受理される言語のクラスを ATIME($T(n)$) と表し、領域 $O(S(n))$ 時間 $O(T(n))$ 限定 (i)ATM によって受理される言語のクラスを ASPACE,TIME($S(n)$,$T(n)$) と表すことにすると、NC^k は ATM によって次のように特徴づけされる。

性質

ATIME($\log n$)$\subseteq$$\mathrm{NC}^1$ (文献 [8][14])

すべての $k \geq 2$ に対して

ASPACE,TIME($\log n$,$\log^k n$)=NC^k (文献 [14]) □

次に示すのが本研究で得られた主たる結果のうちの 1 つである。

定理 7 *QDLIN* は NC^1に属す。

(証明) 任意の QDLIN L は図 4 に示すような動作をする iATM Mによって受理される。Mは L を生成する QDLG G が持っている補題 2 で示した性質に基づいて設計されている。Mはまず長さ n の入力 wに対して、G における wの導出の最初の 2 ステップを guess する。次に整数 s を guess し、これが導出の最終ステップに現れる記号の位置であるとする。この後の動作は G での 2 ステップ目までの guess の仕方により場合分けされるが、ここでは 2 ステップ目に適応される規則 p_2が右線形である場合を考えよう。その他の場合も同様な考え方で動作する。

最初の 2 ステップの guess が完了しているため、これ以降は第 1 ステップで導出された記号は認識の対象としなくてよい。いま導出の第 1 ステップ終了時点で現れている非

終端記号を A、第2ステップ以降で生成しなければならない入力の部分記号列を $xcy \in V_T^*$ とする。ただし、c は入力記号列 w の s 番目の記号である。A は右線形規則 p_2 によって書き換えられることがわかっているので、図2に示したように $A \overset{*}{\underset{G}{\Rightarrow}} xcy$ かどうかはある X および x の接頭語 x' が存在して $A \overset{*}{\underset{G}{\Rightarrow}} x'X$, $A \overset{*}{\underset{G_R}{\Rightarrow}} xc$ かつ $X \overset{*}{\underset{G_L}{\Rightarrow}} cy$ かどうかを確かめれば十分である。これら3つの導出の存在の確認は正規集合を認識する方法を利用して $O(\log n)$ 時間で可能である。他の部分も $O(\log n)$ 時間で実行可能なことは明らかであるので、QDLIN L は iATM によって $O(\log n)$ 時間で認識可能である。 □

定理7より直ちに次の結果を得る。

系 7.1 QDLIN は DSPACE($\log n$) に属す。 □

ここで、QDLG の概念を超線形文法 [6][7] (MLG) まで拡張して準決定性超線形文法 (QDMLG) および準決定性超線形言語 (QDMLIN) を定義する。

```
begin /* 入力は w = a1a2...an であるとする。*/
  guess S =p1=>G w =p2=>G a1...al X ar...an
  (|a1...al| + |ar...an| = 2);
  l, r の2進表現および X ∈ VN を work tape に書く
  (ただし a1...al = ε の場合は l = 0,
  ar...an = ε の場合は r = n+1 とする);
  guess s(l+1 ≤ s ≤ r−1);
  s の2進表現を work tape に書く;
  /* 右線形のみまたは左線形のみで導出される場合 */
  if l = 2 and s = n then
    if X =*=>G a(l+1)a(l+2)...as then accept else reject;
  if r = n−1 and s = 1 then
    if X =*=>G as a(s+1)...a(r−1) then accept else reject;
  case p2 of
    p2 が右線形規則:
    begin
      guess i(l ≤ i ≤ s) and Y ∈ VN;
      do universally
        (a) X =*=>GR a(l+1)a(l+2)...a(s−1)as かを確かめる;
        (b) if i = l then X = Y かを確かめる
            else X =*=>G a(l+1)a(l+2)...ai Y かを確かめる;
        (c) Y =*=>GL as a(s+1)...a(r−1) かを確かめる;
      end;
    end;
    p2 が左線形規則:
    begin
      guess i(s ≤ i ≤ r) and Y ∈ VN;
      do universally
        (a) X =*=>GL as a(s+1)...a(r−1) かを確かめる;
        (b) if i = r then X = Y かを確かめる
            else X =*=>G Y ai a(i+1)...a(r−1) かを確かめる;
        (c) Y =*=>GR a(l+1)a(l+2)...as かを確かめる;
      end;
    end;
  end;
end.
```

図 4: QDLIN L を受理する iATM M の動作

定義 11 超線形文法 *(MLG)* $G = (V_N, V_T, S, P)$ は次の *4* つの形の規則からなる文法である。

$S \to A_1 \ldots A_m, \quad A_i \in V_N - \{S\}$

$A \to aB, A \to Ba, A \to a, \quad A, B \in V_N - \{S\},\ a \in V_T \cup \{\varepsilon\}$

MLG によって生成される言語を超線形言語 *(以下、MLIN)* という。 □

定義 12 *MLG* $G = (V_N, V_T, S, P)$ は

1. $A \in V_N - \{S\}$ であるような任意の $A \to aB \in P$ に対して、
   $A \Rightarrow aB \Rightarrow \cdots \Rightarrow aY_1\alpha \Rightarrow abX_1\alpha$
   $A \Rightarrow aB \Rightarrow \cdots \Rightarrow aY_2\beta \Rightarrow abX_2\beta$
   ならば、$X_1 = X_2$。ただし、$A, B, X_1, X_2, Y_1, Y_2 \in V_N$, $a, b \in V_T$, $\alpha, \beta \in V_T^*$。

2. $A \in V_N - \{S\}$ であるような任意の $A \to Ba \in P$ に対して、
   $A \Rightarrow Ba \Rightarrow \cdots \Rightarrow \alpha Y_1 a \Rightarrow \alpha X_1 ba$
   $A \Rightarrow Ba \Rightarrow \cdots \Rightarrow \beta Y_2 a \Rightarrow \beta X_2 ba$
   ならば、$X_1 = X_2$。ただし、$A, B, X_1, X_2, Y_1, Y_2 \in V_N$, $a, b \in V_T$, $\alpha, \beta \in V_T^*$。

を満たすとき準決定性超線形文法 *(QDMLG)* であるという。*QDMLG* によって生成される言語を準決定性超線形言語 *(QDMLIN)* という。 □

系 7.2 QDMLIN は NC^1 に属す。 □

4.2 PRAM による認識

ここでは、一般的に並列アルゴリズムの設計および解析において使われている並列ランダムアクセス機械 (PRAM) [4] 上での QDLIN の並列認識について考察する。PRAM はランダムアクセス可能な一つの共有メモリを介して結合された複数個のプロセッサからなる SIMD 型の並列計算モデルである。PRAM ではプロセッサが共有メモリのメモリセルへアクセスする際、複数個のプロセッサが同じメモリセルへ同時にアクセスすることが考えられるため、これに制約を与えることにより

1. 同時読み出しも同時書き込みも許す (CRCW)
2. 同時読み出しは許すが同時書き込みは許さない (CREW)
3. 同時読み出しも同時書き込みも許さない (EREW)

の3つのタイプに大別される。能力的には 1., 2., 3. の順に小さくなる。本研究で用いるのは 2. の CREW PRAM である。

QDLIN の CREW PRAM での並列認識を説明する前に、2,3 の記法を示しておく。

(記法)

$$V_{R,i,j} = \{(A, B, 0) \mid A \overset{*}{\underset{G_R}{\Rightarrow}} a_i \ldots a_j B\} \cup \{(A, \#, s) \mid A \overset{*}{\underset{G_R}{\Rightarrow}} a_i \ldots a_s,\ i \le s \le j\}$$

$$V_{L,i,j} = \{(A, B, 0) \mid A \overset{*}{\underset{G_L}{\Rightarrow}} B a_j \ldots a_i\} \cup \{(A, \#, s) \mid A \overset{*}{\underset{G_L}{\Rightarrow}} a_s \ldots a_i,\ j \le s \le i\}$$

$$V_{i,j}^{(R)} = \{(A,B,1) | A \overset{*}{\underset{G}{\Rightarrow}} a_i \dots a_j B\}$$
$$\cup \{(A,B,0) | \text{for } i \le k < j \ A \overset{*}{\underset{G}{\Rightarrow}} a_i \dots a_k B \text{ and } {}^\forall C \in V_N, B \to a_{k+1} C \notin P\}$$
$$V_{i,j}^{(L)} = \{(A,B,1) | A \overset{*}{\underset{G}{\Rightarrow}} B a_j \dots a_i\}$$
$$\cup \{(A,B,0) | \text{for } j < k \le i \ A \overset{*}{\underset{G}{\Rightarrow}} B a_k \dots a_i \text{ and } {}^\forall C \in V_N, B \to C a_{k+1} \notin P\}$$

定理 8 *QDLIN* は *CREW PRAM* で $O(\log n)$ 時間、n プロセッサで認識可能である。

(証明) 一般性を失うことなく $G = (V_N, V_T, S, P)$ を簡約化された QDLG とする。CREW PRAM は G によって生成される QDLIN $L(G)$ を認識する ATM M を以下のように模倣する。

1. (a) 任意の $s(2 \le s \le n)$ および $A \in V_N$ に対して $A \overset{*}{\underset{G_R}{\Rightarrow}} a_2 \dots a_s$ かどうかを確認する。
   (b) 任意の $s(1 \le s \le n)$ および $A \in V_N$ に対して $A \overset{*}{\underset{G_R}{\Rightarrow}} a_1 \dots a_s$ かどうかを確認する。
2. (a) 任意の $t(1 \le t \le n-1)$ および $A \in V_N$ に対して $A \overset{*}{\underset{G_L}{\Rightarrow}} a_t \dots a_{n-1}$ かどうかを確認する。
   (b) 任意の $t(1 \le s \le n)$ および $A \in V_N$ に対して $A \overset{*}{\underset{G_L}{\Rightarrow}} a_t \dots a_n$ かどうかを確認する。
3. (a) 任意の $A, B \in V_N$ に対して $A \overset{*}{\underset{G}{\Rightarrow}} a_2 \dots a_i B$ となる最小の i を求める。
   (b) 任意の $A, B \in V_N$ に対して $A \overset{*}{\underset{G}{\Rightarrow}} a_1 \dots a_i B$ となる最小の i を求める。
   (c) 任意の $A, B \in V_N$ に対して $A \overset{*}{\underset{G}{\Rightarrow}} B a_i \dots a_{n-1}$ となる最大の i を求める。
   (d) 任意の $A, B \in V_N$ に対して $A \overset{*}{\underset{G}{\Rightarrow}} B a_i \dots a_n$ となる最大の i を求める。
4. (a) ($A \overset{*}{\underset{G_R}{\Rightarrow}} a_1 \dots a_s$ かつ $B \overset{*}{\underset{G_L}{\Rightarrow}} a_s \dots a_{n-1}$) または (b) ($A \overset{*}{\underset{G_R}{\Rightarrow}} a_2 \dots a_s$ かつ $B \overset{*}{\underset{G_L}{\Rightarrow}} a_s \dots a_n$)
   であれば、それぞれに対して
   (a) $A \overset{*}{\underset{G}{\Rightarrow}} a_1 \dots a_i B$ かつ $i < s$ または
   $B \overset{*}{\underset{G}{\Rightarrow}} A a_i \dots a_{n-1}$ かつ $s < i$
   (b) $A \overset{*}{\underset{G}{\Rightarrow}} a_2 \dots a_i B$ かつ $i < s$ または
   $B \overset{*}{\underset{G}{\Rightarrow}} A a_i \dots a_n$ かつ $s < i$
   を満たすかを確認する。
5. 4. を満たす長さ $n-1$ の各導出に対して 1 ステップ目としての S を書き換える規則を適用し、$S \overset{*}{\underset{G}{\Rightarrow}} a_1 \dots a_n$ となるものがあれば accept そうでないときは reject する。

3. において a_i に対し条件を加えたのは A から始まる G での (左または右線形のみを使った) 入力 w に対する導出で、G_R の導出木と G_L の導出木が結合できるかどうかの情報として、ある非終端記号 B が現れるかどうかを調べるには導出木のどの位置に現れているかをすべて調べなくとも最初に現れるのがどの位置かを知れば十分だからである。まず 1.(a) は集合 $V_{R,2,n}$ を 1.(b) は集合 $V_{R,1,n}$ を求めることによって確認する。$V_{R,2,n}$ は以下のようにして求めることができる (図 5 参照)。ただし各 $V_{R,i,j}$ は値として 0 または 1 をとる配列 (つまり $V[r] = 0$ なら $r \notin V$、$V[r] = 1$ なら $r \in V$) により表現されているとする。

```
begin
  for i := 1 to n do in parallel
    for each r ∈ P_R do
      if rの右辺の終端記号が a_i then
        rを (定義に見合うように変形して)V_{R,i,i}に入れる;
    end;
  end;
  for m := 1 to ⌈log n⌉
    for i := 2 to n step 2^m do in parallel
      merge_R(i, i + 2^{m-1} − 1, i + 2^{m-1}, i + 2^m − 1);
    end;
  end;
end.
procedure merge_R(i1, j1, i2, j2)
begin
  for each (A, #, s_1) ∈ V_{R,i1,j1} do in parallel
  (A, #, s_1) を V_{R,i1,j2}に入れる;
  end;
  for each (A, B, 0) ∈ V_{R,i1,j1}, B ≠ # and
  (B, C, s_2) ∈ V_{R,i2,j2} do in parallel
  (A, C, s_2) を V_{R,i1,j2}に入れる;
  end;
end;
```

$V_{R,2,n}$ を $O(\log n)$ 時間で求めるには、各 $V_{R,i,i+2^m-1}$ を $O(1)$ 時間で求めなければならない。1つの $V_{R,i,i+2^m-1}$ を $O(1)$ 時間で求めるには上の手続きよりわかるように $V_{R,i,i+2^{m-1}-1}$ (または $V_{R,i+2^{m-1},i+2^m-1}$) の要素数が $O(2^m)$ であるので 2^m 個のプロセッサが必要となる。i の値は m が固定されると $n/2^m$ 個の値をとるので、結局 $V_{R,2,n}$ を $O(\log n)$ 時間で求めるために必要なプロセッサ数は $\frac{n}{2^m} \times 2^m = n$ となる。2. に対しても 1. と同様な方法で集合 $V_{L,n-1,1}$ および $V_{L,n,1}$ を求めることで確認する。

問題となるのは 3. を求める方法であるが、これは (a) の場合なら $A \overset{*}{\underset{G}{\Rightarrow}} a_i \dots a_j B$ の導出木の中に現れるすべての非終端記号を、最初にそれが現れるとき一緒に生成された入力記号の位置と共に記憶しておくようにすれば、非終端記号の数が有限個であるから $V_{ij}^{(R)}$ の要素にそれらを付加的情報として追加することで 1. および 2. の場合と同様な方法を使って (それらとの高々定数倍の計算コストの違いで) 計算することが可能である。(b) の場合も同様である。

4. および 5. も n プロセッサを用いると $O(\log n)$ 時間で実行可能であることは明らかであるので、上記の CREW PRAM による ATM の模倣は $O(\log n)$ 時間、n プロセッサで可能である。 □

5 むすび

本研究では、線形文法 (LG) に制限を加えた準決定性線形文法 (QDLG) によって生成される言語である決定性線形言語 (QDLIN) が NC^1 に属し、CREW PRAM で $O(\log n)$ 時間、n プロセッサで認識可能であることを示した。さら

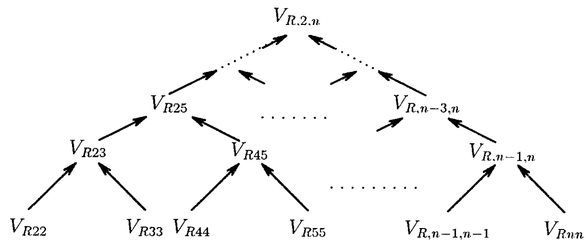


図 5: QDLIN を認識する PRAM の基本的な計算方法

に系として LG に与えた制限を超線形文法 (MLG) まで拡張した QDMLG によって生成される QDMLIN も NC^1に属すことが明らかになった。

ところで、Ibarra et al. [8]が導入した決定性線形 CFL (DLCFL) は本研究で DPDA によって特徴づけられたわけであるが、これに対し QDLIN は "ある意味で協調的かつ決定的な動作を行なう 2 台の有限オートマトン" あるいはもっと単純にいえば "ある種の制限を加えた 2 テープオートマトン" によって特徴づけられるのではないかと考えられる。これは松田 [11] によって定義された QDLIN の部分クラスである SLIN が決定性 2 テープ有限オートマトンによって特徴づけられているということにも対応している。

未解決問題として、"$\mathcal{L}$(QDLIN)=$\mathcal{L}$(LIN) か?" が残っており、これは LIN が NC^1に属すかどうかに関わる問題である。しかしながら現在までのところ、反例となることが予想される言語は見つかっていない。

参考文献

[1] 阿江, 大崎: "制限された消去オートマトン", 信学論 (D), Vol. J61-D, No. 7, pp. 504-511 (1978-07).

[2] Chandra A. K., Kozen D. C. and Stockmeyer L. J.: "Alternation", Journal of the Association for Computing Machinery, Vol. 28, No. 1, pp. 114-133 (Jan. 1981).

[3] Dymond P. W.: "Input-Driven Languages are in $\log n$ Depth", Information Processing Letters, Vol. 26, pp. 247-250 (1988).

[4] Fortune S. and Wyllie J.: "Parallelism in Random Access Machines", Proc. 10th ACM Symposium on Theory of Computation, pp. 114-118 (1978).

[5] Gibbons A. and Rytter W.: "Efficient Parallel Algorithms", Cambridge Univ. Press (1988).

[6] Harrison M. A.: "Introduction to Formal Language Theory", Addison-Wesley, Reading, MA (1978).

[7] Hopcroft J. and Ullman J.: "Introduction to Automata Theory, Languages and Computation", Addison-Wesley, Reading, MA (1979).

[8] Ibarra O., Jiang T. and Ravikumar B.: "Some Subclasses of Context-Free Languages in NC^1", Information Processing Letters, Vol. 29, pp. 111-117 (1988).

[9] 岩間一雄: "PRAM 上の並列アルゴリズム", 情報処理, Vol. 33, No. 9, pp. 1033-1041 (Sep. 1992).

[10] Klein P. and Reif J.: "Parallel Time $O(\log n)$ Acceptance of Deterministic CFLs on an Exclusive-Write P-RAM", SIAM J. Comput., Vol. 17, No. 3, pp. 463-485 (June 1988).

[11] 松田聖: "ある無あいまい文法族と決定性 2 テープオートマトン", 信学論 (D), Vol. 57-D, No. 1, pp. 46-53 (1974-01).

[12] 宮野悟: "並列化とその限界—理論的側面から", コンピュータソフトウェア, Vol. 7, No. 1, pp. 2-15 (Jan. 1990).

[13] Ritchie R. W. and Springsteel F. N.: "Language Recognition by Marking Automata", Information and Control, Vol. 20, pp. 313-330 (1972).

[14] Ruzzo W.: "On Uniform Circuit Complexity", J. Comput. System Sci., Vol. 22, pp. 365-383 (1981).

[15] Rytter W.: "Parallel Time $O(\log n)$ Recognition of Unambiguous Context-Free Languages", Information and Computation, Vol. 73, pp. 75-86 (1987).

[16] Rytter W. and Giancarlo R.: "Optimal Parallel Parsing of Bracket Languages", Theoretical Computer Science, Vol. 53, pp. 295-306 (1987).

[17] Rytter W.: "On Efficient Parallel Computations of Costs of Paths on a Grid Graph", Information Processing Letters, Vol. 29, pp. 71-74 (1988).