数理解析研究所講究録
第156巻 1972年 213-229

ANALYZERの簡単化

阪大 基礎工 菊野 亨

1. まえがき

コンパイラにおけるいわゆる"解析部"(analysis part)の主な目的は、与えられたソースプログラムを分析し、パラメータをもってセマンチックルーチンをつぎつぎと呼ぶことである。たとえば、記号表へのいくつかのポインタをパラメータとして内部コード発生ルーチンをつぎつぎと呼ぶ。

本稿では、このようなアナライザのモデルとして、一定量までの先よみを許した決定性のプッシュダウンオートマトンを考える。(図1) プッシュダウンスタックは、動作を決定するために用いるコントロール記号を書きこむC-トラックと、呼ぶべきセマンチックルーチンに対するパラメータを書きこむS-トラックに

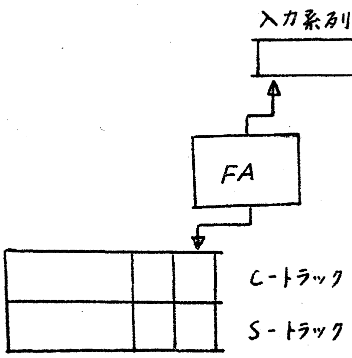
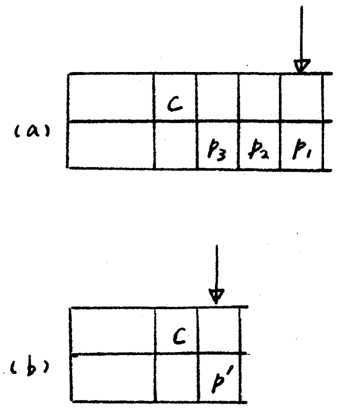


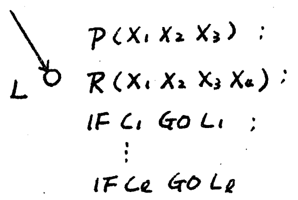
図1

わかれている。このアナライザのコントロール部はフローチャートの形で表現される。



図－2

今かりにプッシュダウンスタックが図2(a)のようであるとし、コントロールがフローチャート上で図3の節点Lにきた場合を考える。Pは新しいパラメータを求めるルーチン名である。まず現在のS－トラックのtopの3個のp_1, p_2, p_3をそれぞれX_1, X_2, X_3に代入するパラメータとしてルーチンPを呼び、新しいパラメータp'を求める。次にX_1, X_2, X_3, X_4にそれぞれパラメータp_1, p_2, p_3, p'を代入しセマンチックルーチンRを呼ぶ。スタックは長さ2だけ短かくなる。p'をS－トラックのtopに書きこみ(図2(b))、左となりのコマのコントロール記号CがC_iに等しければ、次は節点L_iから実行する。フローチャートでは、このようにルーチンを呼ぶことを指定した節点のほか、C－トラックのtopにコントロール記号を書きこむ動作を指定した節点や、先よみのヘッドを一つ右へ動かし、入力記号の先よみを指定する節点などもある。



図－3

二つのアナライザA, Bは、任意の入力系列(ソースプロ

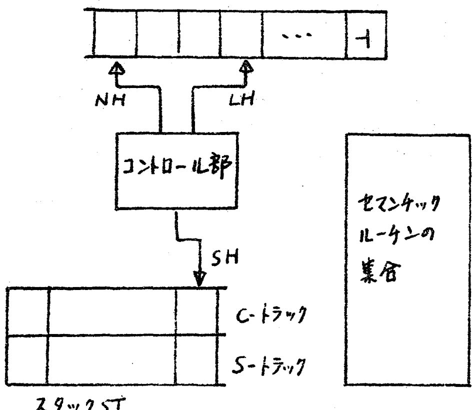
グラムまたは語い解析を終えたもの）χに対し、Aがχを拒否するならBも拒否し、Aが受理するならBも受理し、かつAとBが同じパラメータで同じセマンチックルーチンをつぎつぎと呼んだとき、等価であるという。C-トラックにコントロール記号を書きこむ動作や、先よみをする動作などは、等価性には直接関係はない。等価性に意味があるのは、パラメータをもってセマンチックルーチンRを呼ぶという動作のみである。すなわち、セマンチックルーチンRをパラメータ$p_1, \cdots, p_\ell$をもって呼ぶ動作を、CALL R($p_1, \cdots, p_\ell$)と書くことにすると、アナライザの全動作系列のうちでセマンチックルーチンを呼ぶ動作だけを抜き出した系列…, CALL R($p_1, \cdots p_\ell$)…が（記号の系列として）同じであるとき等価であるという。パラメータ$p_1, \cdots, p_\ell$をもってルーチンPを呼んで新しいパラメータp'を求めるとき、p'はルーチン名Pとそのときのパラメータの値$p_1, \cdots, p_\ell$とその時点までに読みこんだ入力系列によって一意にきまるものとし、それらが全部同じときのみ同じp'が求まるとする（いわゆる自由解釈）。

本稿では、まず、与えられた二つのアナライザが等価であるかどうか決定可能であることを述べ、次に、与えられたアナライザと等価なアナライザのうちで、もっとも簡単なもの（直観的に言って、プログラムで書くとしてもっとも速く実

行するプログラムの中で、もっとも短かいプログラム（詳しくは後述）を求める方法を述べる。

なお、特定の文法Gを対象にし、与えられた入力系列がGで導出されるセンテンスかどうかを判別し、もしセンテンスならその導出木（節点のラベルはGの非終端記号またはGのルール名）のすべてを与えるものを"Gに対するパーザ"という。パーザの簡単化については文献(2)－(5)などで扱われているが、本稿でいうアナライザの簡単化はパーザのそれとは異なる。

2. アナライザのモデル



図－4

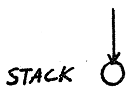
便宜的に図4のようなモデルを考える。動作開始時には、ヘッドNH，LHとも入力テープの左端に、ヘッドSHはスタックの底にある。動作中のNHとLHの距離は、高々Kコマとする。アナライザのコントロールは計算機のプログラムに対応した次のようなフローチャートで書かれる。フローチャートの各節点には、そこにコントロールがきたときにとるべき基本動作が書いてある。

入力アルファベットの集合Σの部分集合Mに対し、$\Sigma - M$を一つの記号とみて、この記号の集合を$\mathfrak{M}$とする。

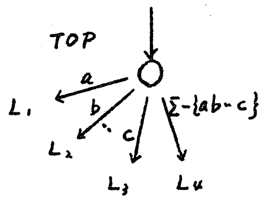
$$\mathcal{L} = \{\epsilon\} \cup X \cup X^2 \cup \cdots \cup X^k \qquad X \in \Sigma \cup \mathfrak{M}$$

とする。$X \in \Sigma \cup \mathfrak{M}$ に対し、関数αを$X \in \Sigma$ なら$\alpha(X) = X$, $X = \Sigma - M \in \mathfrak{M}$ なら$\alpha(X) = \{x \mid x \in \Sigma,\ x \notin M\}$ とする。

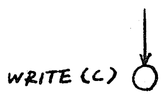
基本動作



(1) STACK　SHを1コマ右へ動かして、そのC-トラックとS-トラックにNHのコマの入力記号をスタックする。† NHは1コマ右へ動く。



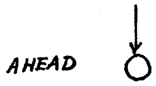
(2) TOP　SHのコマのC-トラックの記号をみて、それによって次の動作をきめる。たとえば、図の場合、その記号がaならL_1へ、bならL_2へ、…、cならL_3へ、それ以外ならL_4へとコントロールが移る。(TOPはやるとしてもSTACKの直後だけとする)。



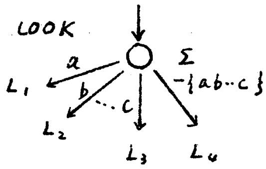
(3) WRITE(c)　SHのコマのC-トラックにコントロール記号cを書きこむ。

---

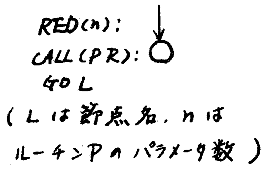
† 入力系列が語い解析を終えたものと考える場合は、たとえば、identifierであることを示す記号IがC-トラックに、記号表へのポインタがS-トラックにスタックされる。



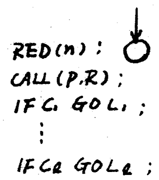
(4) AHEAD　LH を 1 コマ右へ動かす。



(5) LOOK　LH のコマの記号をみて，(2) と同じように次の動作をきめる。



(6) RED(n); CALL(P,R); GO L　現在の S-トラックの top の n 個の記号 $(p_1 \cdots p_n)$ をパラメータとしてルーチン P を呼び，新しいパラメータ p' を求める。パラメータ $(p_1 \cdots p_n, p')$ をもってセマンチックルーチン R を呼ぶ。スタックは長さ $(n-1)$ だけ短かくなる。p' を S-トラックの top に書きこむ。次は節点 L から実行する。



(7) RED(n): CALL(P,R); IF C_1 GOL_1; … ; IF C_ℓ GOL_ℓ

(6) と次のことを除いて同じ動作をする。すなわち，スタックが長さ $(n-1)$ だけ短かくなった後の左となりのコマの C-トラック内のコントロール記号 C が C_i なら節点 L_i から実行する。

(8) ACCEPT　入力系列を受理して停止する。このときには，NH，LH はいずれも入力テープの右端の記号 ⊣ のコマにあり，SH は ST の底から 2 番目を指していることが保障されているものとする。

(9) REJECT　入力系列を拒否して停止する.

3. アナライザの等価性

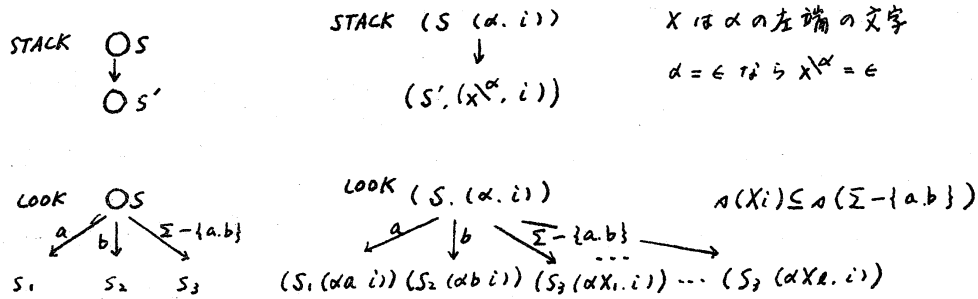
アナライザは、そのコントロール部のフローチャートで与えられる。そのフローチャートから一定の方法で文法 $G=(V_N, V_T, R, S)$ と写像 $h: V_N \to \tilde{V}_N$ を作る。G の任意の導出木 t に対し(端点でない節点の)各ラベル $X \in V_N$ を $h(X) \in \tilde{V}_N$ で置きかえて得られる木を簡単に $h(t)$ と書く。$\mathcal{J}(G, h) = \{h(t) \mid t \in D(G)\}$ (ただし $D(G)$ は G の導出木の全体)とする。任意のアナライザ A, B に対しその方法で作った文法 G_A, G_B と写像 h について、$\mathcal{J}(G_A, h) = \mathcal{J}(G_B, h)$ のとき、かつそのとき A と B は等価であるがなりたつ。

3.1　フローチャートの変換

アナライザ A はフローチャートの形で与えられる。これを $FL(A)$ とする。この $FL(A)$ から、各節点を分解して、先よみ情報(先よみ系列と LH の位置)が一意に対応するようにする。これを $FL(A)'$ とする。つまり、$FL(A)'$ の各節点は $(S, (\alpha\ i))$ の形をしており、$|\alpha| \leq K$, $\alpha \neq \epsilon$ のときは $i=|\alpha|$ か $|\alpha|+1$, $\alpha = \epsilon$ のときは $i=1$ とする。次にその手続きを述べる。

$FL(A)$ の各節点 S について、$\alpha \in \mathcal{Z}$, i のすべての組み合わせで節点 $(S, (\alpha, i))$ を作る。ラベルは S に書かれているもの

をそのまま書く。ただし ACCEPT に対応する節点は $(S_F,(\dashv,1))$ だけとする。(S_F は FL(A) で ACCEPT をラベルとしてもつ節点)。次に、枝は節点のラベルに対応して、次のように結ぶ。



TOP, WRITE(c) は FL(A) で S から S′ へ枝があれば、(S, (α, i)) から (S′, (α, i)) へ、AHEAD の場合は (S, (α, i)) から (S′, (α, i+1)) へ枝を結ぶ。開始点は $(S_0,(\epsilon,1))$ (S_0 は FL(A) の開始点) とし、最後に開始点から到達できぬ節点はすべて除く。

3.2 文法 G_A と写像 h

A の行なう基本動作の中で、STACK と RED(n); CALL(P, R) の系列と ACCEPT だけが等価性に意味がある。そこで STACK や RED から次の STACK や RED の間にある動作系列はまとめて、ξ-系列とする。すなわち、ξ-系列とは次のようなフローチャート FL(A)′ 上の節点の系列である。(1) 始まりは、開始点 $(S_0,(\epsilon,1))$ か、RED での飛び先か、STACK (TOP) の直後の節点で、(2) 終わりは、ラベルが STACK か RED か ACCEPT であるような節点で、(3) その途中には、これら3つの基本動作をラベルとしてもつ節

点をふくまない。 ξ-系列の始まりの節点のラベルを $s(\xi)$, 終わりの節点のラベルを $e(\xi)$ とする。ξ-系列の全体を Ξ とする。

文法 $G_A = (V_N, V_T, R, S)$ を次のように $FL(A)'$ から ξ-系列を使って作る。$V_N = \{(\xi, X\, \xi') \mid \xi, \xi' \in \Xi,\ X \in \Sigma \cup N\}$, $N = \{(P, R) \mid P$ はルーチン, R はセマンチックルーチン$\}$, $V_T = \Sigma$ とする。R は次のように定義する。

各 $X \in V_T$ に対し, 以下の条件 (a)〜(c) がなりたつような

$$(\xi, X, \xi') \longrightarrow X$$

を R のルールとする。 (a) $s(\xi) = STACK(TOP)$, (b) $FL(A)'$ で ξ から ξ' へ枝がある. (c) ξ の実行でスタック(判定された)文字を Y とすると, $Y \in \Sigma$ なら $X = Y$, $Y \in \mathcal{M}$ なら $X \in \mathcal{A}(Y) \cap \Sigma$ 。

また, $(P, R) \in N$ に対し, 以下の条件 (d)〜(g) がなりたつような

$$(\xi_{01}\,(PR)\,\xi_{02}) \longrightarrow (\xi_{11}\, X_1\, \xi_{12})(\xi_{21}\, X_2\, \xi_{22}) \cdots (\xi_{n1}\, X_n\, \xi_{n2})$$

を R のルールとする. (d) $s(\xi_{n2}) = RED(n); CALL(PR);$ (e) $\xi_{01} = \xi_{11}$, $\xi_{12} = \xi_{21}, \cdots, \xi_{n-1,2} = \xi_{n,1}$, $X_i \in \Sigma \cup N$ (f) $e(\xi_{21}) = e(\xi_{12})$, $e(\xi_{31}) = e(\xi_{22})$ $\cdots$ $e(\xi_{n,1}) = e(\xi_{n-1,2})$ は $STACK(TOP)$ (g) $s(\xi_{n2}) = RED(n); CALL(PR); GO\,L$ なら ξ_{02} は節点 L から始まる ξ-系列, $s(\xi_{n,2}) = RED(n); CALL(PR); IF\ c_1\ GO\,L_1; \cdots; IF\ c_\ell\ GO\,L_\ell$ なら ξ_{11} はラベルが $WRITE(c)$ の節点をふくんでおり, もし $c = c_j$ なら ξ_{02} は節点 L_j から始まる ξ-系列。

$S = \{ (\xi.(PR).\xi') \mid$ ξは $(S_0.(\epsilon.1))$から始まるξ-系列，$e(\xi') =$ ACCEPT，(PR)はξ'に入る直前に呼んだルーチン $\}$

ここで，無駄な非終端記号，ルールは除く。次に準同型写像 $h : V_N \to V_T \cup N$ を任意の $(\xi.X.\xi') \in V_N$ に対し $X = a \in V_T$ なら $h(\xi.a.\xi') = a$，$X = (PR) \in N$ なら $h(\xi(PR)\xi') = (PR)$ とする。

<u>補題 1</u>　アナライザAとBが等価であるための必要十分条件は，上で述べた方法で作った文法 G_A，G_B，写像hについて $J(G_A.h) = J(G_B.h)$ がなりたつことである。

$J(G_A.h) = J(G_B.h)$ かどうかは，文法の句構造的等価性の判定と類似の方法で決定できる。ゆえに，

<u>定理 2</u>　アナライザの等価性の問題は決定可能である。

4. アナライザの簡単化

ここでは，アナライザAが与えられたとき，次の条件C1～C5を満すようなアナライザBを求める方法を述べる。

C1　AとBは等価である。

C2　拒否はできるだけ早く行なう，すなわち，Aが受理する入力系列の全体をLとすると，入力系列 $x \notin L$ に対して

は、Bは次のようなxの初期部分系列yの最後の記号を見た（先よみで、あるいは、先よみがなければ STACK 後）時点で拒否する。(1) yのあとにいかなる系列をつけ加えてもLに属さない。(2) yの任意の（真の）初期部分系列y'に対しては、y'のあとに適当な系列をつけ加えればLに属する。

C3　先よみで知った情報は必らず利用する。すなわち、先よみでわかっている文字を再び見ることはないし、TOP 判定も行なわない。

C4　上のC1～C3 を満たすアナライザの中で、先よみの長さが常に最小で、WRITE も必要最小限しか行なわない。

C5　上のC1～C4 を満たすアナライザの中で、プログラムの長さが最短。

4.1　最簡のアナライザの求め方（あらまし）

S1　前節で述べた方法で、文法 G_A と写像 h とを求める。

S2　S1で求めた文法 G_A と写像 h を変換して、次の条件(a), (b) を満たすような文法 $\bar{G}_A$ と写像 $\bar{h}$ を作る。

(a)　$J(G_A, h) = J(\bar{G}_A, \bar{h})$

(b)　$\bar{G}_A$ のルールで、もし $U \longrightarrow z$, $V \longrightarrow z$ と右辺の同じルールがあれば、必ず $\bar{h}(U) \neq \bar{h}(V)$ となっている。

S3　文法 $\bar{G}_A$ に対するパーザを、前述の条件 C2, C3 を考慮

しながら Knuth と類似の方法で作る。

S4　S3で求めたパーザを修正し、与えられたもとのアナライザ A と等価なアナライザへ変換する。

S5　そのフローチャートを有限オートマトンとみて、それをおおう状態数最小のものをみつける。これは Paull-Unger と類似の問題である。

4.2　文法の変換（S2）

$\bar{G}_A$ と $\bar{h}$ は、G_A と句構造的に等価な逆方向決定性文法（backwards-deterministic grammar）の求め方において、h で同じ値になる非終端記号だけをまとめることによって得られる。

4.3　Knuth の K-状態

CFG を $G=(V_N, V_T, R, S)$ で表わす。一般性を失なわずに次の仮定をおく。

(i)　ルールの右辺は ε でない。

(ii)　各 $X \in V_N$ について $X \to \cdots \to X$ でない。

(iii)　無駄な非終記号はふくまない。

文法 G に対して $G' = (V_N \cup \{S_0\}, V_T \cup \{\dashv\}, R \cup \{S_0 \to S\dashv\}, S_0)$ を作る。$S_0, \dashv$ は $V = V_N \cup V_T$ に属さない新しい記号とする。

この G' を改めて $G=(V_N, V_T, R, S)$ とおく。R の元は $S_0 \to S\dashv$ を 0 として $\#(R)-1$（$\#(R)$ は R の元の個数）までの数で適当に順序づけられているとする。p 番目のルールを $X_p \to X_{p1} \cdots X_{pn_p}$

で表わす。 $\alpha \in V^+$, 整数 $K \geqq 0$ に対して $H(\alpha) = \{ t \in V_T{}^K \mid$ G で $\alpha \overset{*}{\Rightarrow} t\beta$ なる $\beta \in V^*$ が存在する。$\}$ とおく。Knuth (1) にしたがって、K-状態 $\mathcal{S}$ の集合 $\mathcal{D}$ を次のように求める。まず、$\mathcal{S}_0 = \{ [0.0.\dashv] \}$, $\mathcal{D} = \{ \mathcal{S}_0 \}$ とする。任意の $\mathcal{S} \in \mathcal{D}$, $Y \in V$ について $\mathcal{S}_Y = \{ [p, j+1, \alpha] \mid [p.j.\alpha] \in \mathcal{S}'$ かつ $X_{p,j+1} = Y \}$ を求め、空でなければ $\mathcal{D}$ へ入れる。ここで $\mathcal{S}'$ は $\mathcal{S}' = \mathcal{S} \cup \{ [q.0.\beta] \mid X_{p,j+1} = X_q,\ \beta \in H(X_{p,j+2}, \cdots X_{pn_p}\ \alpha)\ j < n_p$ なる $[p.j.\alpha] \in \mathcal{S}'$ が存在する $\}$ を満たす最小の集合である。

K-状態 $\mathcal{S}$, ルール p に対し、$Z(\mathcal{S}, p) = \{ \alpha \mid [p.n_p.\alpha] \in \mathcal{S} \}$ とする。また $Z(\mathcal{S}, \mathrm{STACK}) = \{ \beta \mid j < n_p,\ \beta \in H(X_{pj+1} \cdots X_{pn_p}\ \alpha)$ なる $[p.j.\alpha] \in \mathcal{S}'$ が存在する。$\}$ とする。$Z(\mathcal{S}, \mathrm{REJECT}) = V_T{}^K - (Z(\mathcal{S}.\mathrm{STACK}) \cup Z(\mathcal{S}.p_1) \cup \cdots \cup Z(\mathcal{S}, p_\ell))$ とする。G が $LR(K)$ 文法なら、各 $\mathcal{S}$ に対して $Z(\mathcal{S}, P)$, $0 \leqq p \leqq \#(R) - 1$, $Z(\mathcal{S}, \mathrm{STACK})$, $Z(\mathcal{S}.\mathrm{REJECT})$ は互いに素である。ここで、各 $Z(\mathcal{S}.X)$ (X は $\mathrm{RED}(p), \cdots, \mathrm{RED}(q)$, STACK あるいは REJECT) の各要素 α を次のような α' でおきかえる。(i) α' は α の初期部分系列、(ロ) α' は他の $Z(\mathcal{S}, X)$ の要素の初期部分系列ではない。これは、不必要な先よみをさけるためである。

4.4 パーザの構成 (S3)

拒否をできるだけ早くおこなうため、次の動作を選ぶときは、拒否か STACK か RED(p) かが一意に決まるだけの先よみを

するが、それは常に必要最小限の長さだけ行なうようなパーザを考える．しかも一度先よみで知った情報は必らず利用する．G_A に対するパーザの構成法については文献(7)参照(拒否を遅らせないことと、AHEAD の動作を考慮する。)

4.5　アナライザへの変換（S 4）

G_A に対するパーザで RED(p) を RED(np) : CALL($\overline{T_1}$(Xp)) でおきかえる。このコントローラ C_1 をもつアナライザを P_1 とする。C_1 の WRITE(c) や RED(n) : CALL(PR) ; IFC1 GOL1 : … ; IFCℓ GOLℓ（または GOL）を単に WRITE や RED(n) : CALL(PR) でおきかえ、枝のラベルを入力記号、その枝から次の分岐点または端点に至るまでの基本動作の系列を、その入力記号に対する出力記号とみて、これを有限オートマトン（F_1 とする）と考える．

4.6　アナライザの簡単化（S 5）

あるアナライザ P_2 のコントローラ C_2 が与えられたとする．C_2 を前と同じように有限オートマトンとみたものを F_2 とする．有限オートマトン F_1 のある状態 t から定義されているすべての入出力系列が F_2 のある状態 s からも定義されていて等しいとき s は t をおおうという．次の条件がなりたつとき F_2 は F_1 をおおうという．

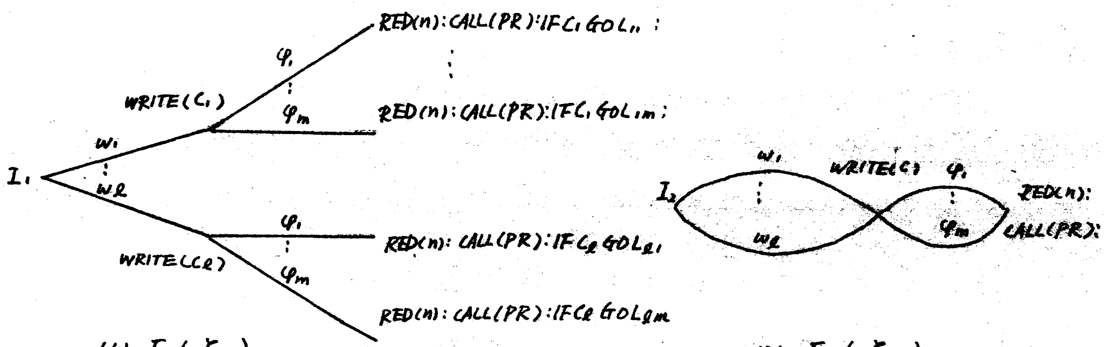
(1) F_2 の初期状態 I_2 は、F_1 の初期状態 I_1 をおおう．

(2) F_1, F_2 の入力系列 $w_1 \cdots w_\ell$, $\varphi_1 \cdots \varphi_m$ に対し図 5 のようになっ

ていれば $L_{11},\cdots,L_{1m},\cdots,L_{\ell 1}\cdots,L_{\ell m}$ の全体をともにおおうような状態 L' が F_2 に存在する。

(3) 上のような L' の中で少くとも1つの L' が存在して、図5-(2)の I_2 を L' とし、同図(1)の I_1 を上の $L_{11}\cdots L_{1m},\cdots,L_{\ell 1}\cdots L_{\ell m}$ の全部にわたって考えたときの図5における $L_{11},\cdots,L_{\ell m}$ の全体をともにおおうような状態が F_2 に存在する。

(4) 上の(3)をくり返し考えても、(3)はつねになりたつ。



(1) $F_1(\varepsilon_1)$　　(2) $F_2(\varepsilon_2)$

図-5

さて、次のことが証明される。

定理3　P_2 が P_1 と等価であるための必要十分条件は、F_2 が F_1 をおおうことである。

F_1 をおおう F_2 は、両立、許容の定義を適当に行なえば、F_1 の状態の両立許容分解から得られる。すなわち、P_1 の簡単化の問題は、順序機械における Paull - Unger の問題と類似の問題に帰着される。

とくに上の F_1, F_2 において、L から L' へラベル $(n,(PR),r)$ をもつ枝をつけて得られる有限オートマトンを $\tilde{F}_1, \tilde{F}_2$ とする。

<u>定理4</u>　$\tilde{F}_1, \tilde{F}_2$ が決定性の有限オートマトンならば、P_2 が P_1 と等価であるための必要十分条件は、$\tilde{F}_2$ の初期状態 I_2 が $\tilde{F}_1$ の初期状態 I_1 をおおうことである。

従って $\tilde{F}_2$ を求めることは、$\tilde{F}_1$ の状態の通常の意味の両立許容分解を求めることになる。状態数最小の $\tilde{F}_2$ を求めることは、よく知られた Paull-Unger の問題になる。

5. あとがき

実際のコンパイラにおいては、パラメータだけ書きかえて、内部コードを発生しない動作もある。本稿では、簡単のためつぎの内部コード発生までの何回かのパラメータの書きかえを、一つのルーチン P で表現したが、パラメータ書きかえのルーチンの入れ子構造や、セマンティックルーチン R へ代入したとき、ある種の等価性を定義しておけば、本稿で述べたのと同じ意味でアナライザの等価性が決定でき、また同様な方

法でアナライザの簡単化ができる。

また、本稿では先よみは同じ部分を2度繰り返しては見ない（LH は左にもどらない）としたが、この制限をなくせば、プログラムがより短かくなる可能性がある。いずれにせよ、計算時間と記憶容量のかねあいの問題である。

文　献

(1) D.E. Knuth : "On the translation of languages from left to right", Inform. Control 8, 5, pp. 607-639, (Oct. 1965)
(2) D. Pager : "A solution to an open problem by Knuth", Inform. Control 17, 5, pp. 462-473, (Dec. 1970)
(3) F.L. DeRemer : "Simple LR(K) grammars", Comm. ACM 14, 7, pp. 453-460, (July 1971)
(4) 林 : "CFG-PL 変換について", 情報処理, 12, 3, pp. 145-153 (1971-03)
(5) 雨宮 : "LR(k) Parser について" 昭46 情報処理学会大会, 115.
(6) M.C. Paull & S.H. Unger : "Minimizing the number of states in incompletely specified sequential switching functions", IRE Trans. Comput. E.C.-8, 3, pp. 356-366, (Sept 1959)
(7) 谷口, 菊野, 嵩 : "構文解析の簡単化について", 信学会 オートマトン研資 (1972-01)