

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ティックルー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、一定量お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

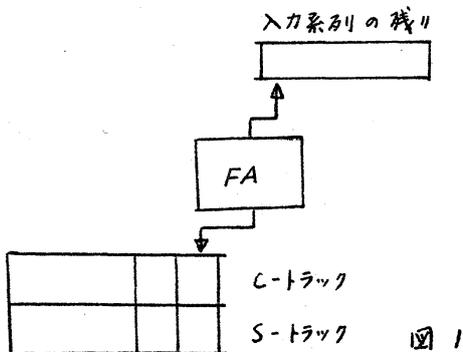


図1

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、呼ぶべきセマンティックルーンに対するパラメータを書きこむS-トラックに

わかれてくる。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。

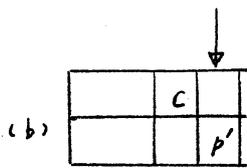
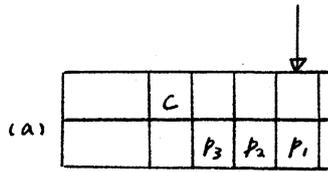


図-2

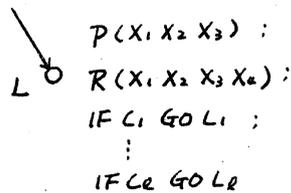


図-3

今かりにフォッシュダウンスタックが図2(a)のようであると、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P₁, P₂, P₃をそれぞれX₁, X₂, X₃に代入するパラメータとしてルーチンPを呼ぶ。新しいパラメータP'を求める。次にX₁, X₂, X₃, X₄にそれぞれパラメータP₁, P₂, P₃, P'を代入しセマンティックルーチンRを呼ぶ。スタックは長さ2だけ短くなる。P'をS-トラックのtopに書きこみ(図2(b))、左と右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す、入力記号の先よみを指定する節点などもある。

二つのアナライザA, Bは、任意の入力系列(ソースフロ

グラム \mathcal{A} は 語 α の 解析 と 終 ϵ の もの λ 又 対 し、 A が α を 拒 否 す る 存 ら B も 拒 否 し、 A が 受 理 す る 存 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 テ ッ ク ル ー チ ン を つ ぎ つ ぎ と 呼 ん だ と き、等 価 だ け ら と い う。C-トラ ッ ク に コ ン ト ロ ー ル 記 号 を 書 き こ ん だ 動 作 や、先 づ き を す る 動 作 は ど け、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 テ ッ ク ル ー チ ン R を 呼 ぶ と い う 動 作 の み だ け だ。す な わ ち、セ マ ン テ ッ ク ル ー チ ン R を パ ラ メ ー タ p_1, \dots, p_n を も っ て 呼 ぶ 動 作 を、 $CALL R(p_1, \dots, p_n)$ と 書 く こ と に す る と、ア ナ ラ イ ー ガ の 全 動 作 系 列 の う ち で セ マ ン テ ッ ク ル ー チ ン を 呼 ぶ 動 作 だ け を 抜 き 出 し た 系 列 $\dots, CALL R(p_1, \dots, p_n) \dots$ が (記 号 の 系 列 と し て) 同 じ だ け ら と き 等 価 だ け ら と い う。パ ラ メ ー タ p_1, \dots, p_n を も っ て ル ー チ ン P を 呼 び 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, \dots, p_n と そ の 時 点 だ け に 読 み こん だ 入 力 系 列 に よ っ て 一 意 に 決 ま る 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、予 ず、与 え ら れ た 二 つ の ア ナ ラ イ ー ガ が 等 価 だ け ら だ け ら が 決 定 可 能 だ け ら だ け ら を 述 べ、次 に、与 え ら れ た ア ナ ラ イ ー ガ と 等 価 な ア ナ ラ イ ー ガ の う ち で、よ っ と よ 簡 単 な も の (直 観 的 に 言 っ て、フ ロ グ ラ ム で 書 ぶ と し て よ っ と よ 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)を求めろ方法を述べろ。

なお、特定の文法Gを対象にし、与えられろ入力系列が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不(節点のラベルはGの非終端記号またはGのルール名)のすべてを与えろものを"Gに対するパーサ"という。パーサの簡単化については文献(2)-(5)などで扱われろているが、本稿でいうアナライザの簡単化はパーサのそれとは異なる。

2. アナライザのモデル

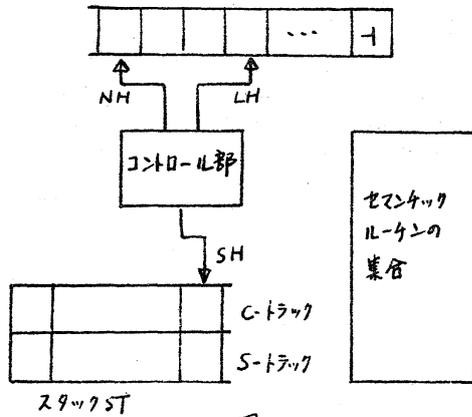


図-4

便宜的に図4のようなモデルを考えると、動作開始時には、ヘッド NH, LH と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ろ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、 $\Sigma - M$ を一つの記号とせよ。この記号の集合を \mathcal{M} とする。

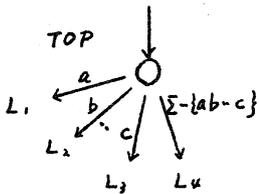
$$\mathcal{X} = \{\epsilon\} \cup X \cup X^2 \cup \dots \cup X^k \quad X \in \Sigma \cup \mathcal{M}$$

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

基本動作



(1) STACK SH のコマンド右へ動かして、 ϵ の C-トラックと S-トラックに NH のコマンドの入力記号をスタックする。† NH は 1 コマンド右へ動く。



(2) TOP SH のコマンドの C-トラックの記号をみよ。それによつて次の動作を定める。たとえば、図 4 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 ϵ として STACK の直後だけとする)。

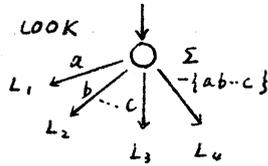


(3) WRITE(c) SH のコマンドの C-トラックにコントロール記号 c を書きこむ。

† 入力系列が語の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。

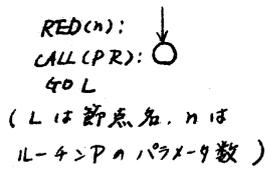


(4) AHEAD LH と 1 コマ 右へ動かす。



(5) LOOK LH の コマ の記号 と MZ. (2) と同

じょう 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 (P₁... P_n) をパラメ

ータとして IL-4 へ P を呼ぶ。新しいパラ

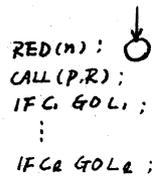
メータ P' を求める。パラメータ (P₁... P_n, P')

を、IL-4 へ R を呼ぶ。スタ

ックは長さ (n-1) だけ短くなる。P' を

S-トラックの top に書きこむ。次の節点 L から

実行する。



(7) RED(n); CALL(P,R); IFC GOL1; ...; IFC GOLn

(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

くなる、以後の左と右のコマの C-トラック

内のコントロール記号 C が Ci なら節点 Li

から実行する。



(8) ACCEPT 入力系列と受理して停止する。



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とする。

(9) REJECT 入力系列を拒否して停止する。

3. アナライガの等価性

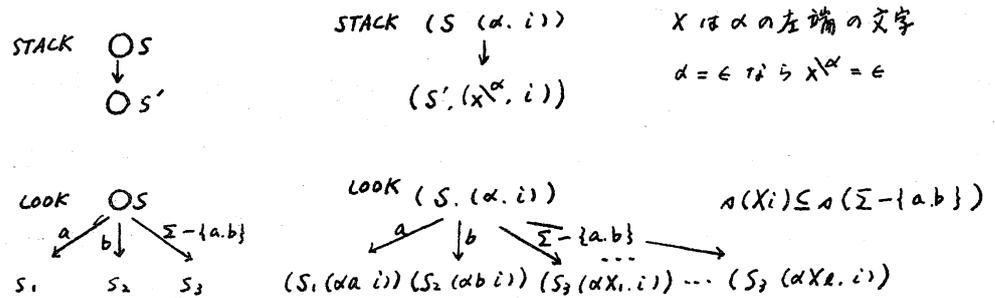
アナライガは、そのコントロール部のフローチャートで与えられる。そのフローチャートから一定の方法で文法 $G = (V_N, V_T, R, S)$ と写像 $h: V_N \rightarrow \tilde{V}_N$ を作る。 G の任意の導出木 t に対し (端点でない節点の) 各ラベル $X \in V_N$ を $h(X) \in \tilde{V}_N$ で置きかえて得られる木を簡単に $h(t)$ と書く。 $J(G, h) = \{ h(t) \mid t \in D(G) \}$ (ただし $D(G)$ は G の導出木の全体) とする。任意のアナライガ A, B に対しその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

アナライガ 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、各節点と分解して、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なる。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、 $\alpha \in \Sigma$, 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

を次のように書く。ただし ACCEPT に対応する節点は $(S_F(t, 1))$ だけとする。 (S_F は $FL(A)$ の ACCEPT をラベルとしておける節点)。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 $FL(A)$ の S から S' へ枝が引ければ、 (S, α, i) から (S', α, i) へ、AHEAD の場合は (S, α, i) から $(S', \alpha, i+1)$ へ枝を結ぶ。開始点は $(S_0, (t, 1))$ (S_0 は $FL(A)$ の開始点) とし、最後に開始点から到達できない節点はすべて除く。

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、 Σ -系列とする。すなわち、 Σ -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 $(S_0, (t, 1))$ か、RED の形の先か、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) 先の途中には、二れら3つの基本動作をラベルとしておける節

点を示す。 ξ -系列の始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, て作る。 $V_N = \{(\xi, X, \xi') \mid \xi, \xi' \in \Xi, X \in \Sigma \cup N\}$, $N = \{(P, R) \mid P \text{ は ルール}$
 η の, $R \text{ は セマニチック ルール}$ $\}$, $V_T = \Sigma$ 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) が成り立つように

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

と R のルールとする。 (a) $s(\xi) = \text{STACK}(\text{TOP})$, (b) $FL(A)'$ で ξ から ξ' へ枝がある。 (c) ξ の実行で n 文字 (判定された) 文字 Y とする。 $Y \in \Sigma$ ならば $X = Y$, $Y \in N$ ならば $X \in \eta(Y) \cap \Sigma$ 。

また, $(P, R) \in N$ に対し, 以下の条件 (d) ~ (g) が成り立つように

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

と R のルールとする。 (d) $s(\xi_{n2}) = \text{RED}(\eta); \text{CALL}(P, R)$; (e) $\xi_{01} = \xi_{11}$
 $\xi_{12} = \xi_{21}, \dots, \xi_{n-1,2} = \xi_{n,1}$, $X_i \in \Sigma \cup N$ (f) $e(\xi_{21}) = e(\xi_{12}), e(\xi_{31}) = e(\xi_{22})$
 $\dots e(\xi_{n,1}) = e(\xi_{n-1,2})$ if $\text{STACK}(\text{TOP})$ (g) $s(\xi_{n2}) = \text{RED}(\eta); \text{CALL}(P, R); \text{GOL}$
 ならば ξ_{02} は節点 L から始る ξ -系列, $s(\xi_{n2}) = \text{RED}(\eta); \text{CALL}(P, R); \text{IF}$
 $C_i \text{ GOL}; \dots; \text{IF } C_i \text{ GOL}$ ならば ξ_{11} はラベルが $\text{WRITE}(C)$ の節点を示
 してあり, もし $C = C_j$ ならば ξ_{02} は節点 L_j から始る ξ -系列。

$S = \{ (\xi, (PR), \xi') \mid \xi \text{ は } (S_0, (E, I)) \text{ から始まる } \xi\text{-系列, } e(\xi') = \text{ACCEPT, } (PR) \text{ は } \xi' \text{ に入る直前に呼んでルール} \}$

ここで、無駄な非終端記号、ルールは除く。次に準同型写像 $h: V_N \rightarrow V_T \cup N$ 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。

補題 1 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 $J(G_A, h) = J(G_B, h)$ が成り立つことである。

$J(G_A, h) = J(G_B, h)$ がどうかは、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。かえに、

定理 2 アナライザの等価性の問題は決定可能である。

4. アナライザの単純化

ここでは、アナライザ A が与えられたとき、次の条件 $C1 \sim C5$ を満た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 $x \in L$ に対し L

は、 B は次のような Σ の初期部分系列 y の最後の記号を見
 込 (先読み) あるいは、先読みがなければ STACK 後) 時
 点で拒否する。(1) y のおとに Σ の系列をつけ加えて
 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' のおとに適當な系列をつけ加えれば L に属す
 る。

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

C4 上の C1 ~ C3 を満たすアライガの中で、先読みの長
 さが常に最小で、WRITE も必要最小限しか行わない。

C5 上の C1 ~ C4 を満たすアライガの中で、プログラムの
 の長さが最短。

4.1 最簡のアライガの求め方 (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、 $u \rightarrow v$, $v \rightarrow w$ と右辺の同じ

ルールがあれば、必ず $\bar{h}(u) = \bar{h}(v)$ となる。

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

したがら Knuth と類似の方法で作る。

S4 S3 で求めたパーカと修正し、与えられたものアナライカが A と等価なアナライカへ変換する。

S5 そのフロ-チャートと有限オートマトンとから、それをおおむね状態数最小のものを見つける。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 で同じ値に与る非終端記号だけを \bar{h} とめることにより得られる。

4.3 Knuth の K-状態

CFG を $G = (V_N, V_T, R, S)$ と表わす。一般性を失はぬおに次の仮定をおく。

- (i) $l - l$ の右辺は ϵ でない。
- (ii) 各 $X \in V_N$ について $X \rightarrow \dots \rightarrow X$ でない。
- (iii) 無駄な非終記号はなくもない。

文法 G に対して $G' = (V_N \cup \{S_0\}, V_T \cup \{+\}, R \cup \{S_0 \rightarrow S+\}, S_0)$ を作る。 $S_0, +$ 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 $S_0 \rightarrow S+$ を 0 とし $\#(R) - 1$ ($\#(R)$ 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 $l - l$ を $X_p \rightarrow X_{p_1} \dots X_{p_{n_p}}$

で表わす. $\alpha \in V^+$, 整数 $k \geq 0$ に対し $H(\alpha) = \{t \in V_T^k \mid$

$G \text{ で } \alpha \xrightarrow{*} t\beta \text{ なる } \beta \in V^* \text{ が存在する.}\}$ とおく. Knuth (1)

にしたがって, K -状態 S の集合 \mathcal{D} を次のように求める.

まず, $S_0 = \{[0, 0, \cdot]\}$, $\mathcal{D} = \{S_0\}$ とする. 任意の $S \in \mathcal{D}$,

$Y \in V$ について $S_Y = \{[p, j+1, \alpha] \mid [p, j, \alpha] \in S' \text{ かつ } X_{p, j+1} = Y\}$

を求め, 空ではない場合は \mathcal{D} へ入れる. ここで S' は $S' = S \cup$

$\{[0, 0, \beta] \mid X_{p, j+1} = X_0, \beta \in H(X_{p, j+2}, \dots, X_{p, n_p}, \alpha) \text{ なる } [p,$

$j, \alpha] \in S' \text{ が存在する}\}$ を満たす最小の集合である.

K -状態 S , $l - l_p$ に対し, $Z(S, p) = \{\alpha \mid [p, n_p, \alpha] \in S\}$ と

する. かつ $Z(S, \text{STACK}) = \{\beta \mid j < n_p, \beta \in H(X_{p, j+1}, \dots, X_{p, n_p}, \alpha) \text{ なる}$

$[p, j, \alpha] \in S' \text{ が存在する.}\}$ とする. $Z(S, \text{REJECT}) = V_T^k -$

$(Z(S, \text{STACK}) \cup Z(S, p_1) \cup \dots \cup Z(S, p_r))$ とする. G が $LR(K)$

文法ならば, 各 S に対し $Z(S, p)$, $0 \leq p \leq \#(R) - 1$, $Z(S, \text{STACK})$,

$Z(S, \text{REJECT})$ は互いに素である. ここで, 各 $Z(S, X)$ (X は

$\text{RED}(p), \dots, \text{RED}(r), \text{STACK}$ あるいは REJECT) の各要素 α を次の

ような α' でおきかえる. (1) α' は α の初期部分系列, (2) α' は

他の $Z(S, X)$ の要素の初期部分系列ではない. これは, 必要

な優先度と置くためである.

4.4 パーサの構成 (S3)

拒否とできるだけ早くおこなうため, 次の動作を選ぶとよ

は, 拒否か STACK か $\text{RED}(p)$ かが一意に決まるだけの優先度と

するが、これは常に必要最小限の長さだけ行なうようなパー
 ガと考える。しかも一度先のみで知った情報は必ず再利用す
 る。 \bar{G}_A に対するパーガの構成法については文献(7)参照(拒否と遷らせたいことと、AHEADの動作と考慮する。)

4.5 アナライザへの変換 (S4)

\bar{G}_A に対するパーガを $RED(p)$ と $RED(np):CALL(\bar{h}(Xp))$ 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 $RED(n):CALL(PR);IFC_1GOL_1;\dots;IFC_2GOL_2$ (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と、その入力記号に対する出力記号
 とし、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 (S5)

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 と前と同じように有限オートマトンとみなすものを F_2 とす
 る。有限オートマトン F_1 のある状態 s から定義されている
 オバエの入出力系列が F_2 のある状態 s' から定義されていると
 等しいとき s は s' とおおうという。次の条件が成り立つとき
 F_2 は F_1 とおおうという。

- (1) F_2 の初期状態 I_2 は、 F_1 の初期状態 I_1 とおおう。
- (2) F_1, F_2 の入力系列 $w_1, \dots, w_l, q, \dots, q_m$ に対し図5のように行な

これら $L_1, \dots, L_m, \dots, L_{l_1}, \dots, L_{l_m}$ の全体とともにおおうような状態 L' が F_2 に存在する。

(3) 上のよう形 L' の中で少くとも一つの 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 $L_1, \dots, L_m, \dots, L_{l_1}, \dots, L_{l_m}$ の全部にわたって考えたときの図 5 における L_1, \dots, L_m の全体とともにおおうような状態が F_2 に存在する。

(4) 上の (3) とくり返し考えても、(3) のつねに成り立つ。

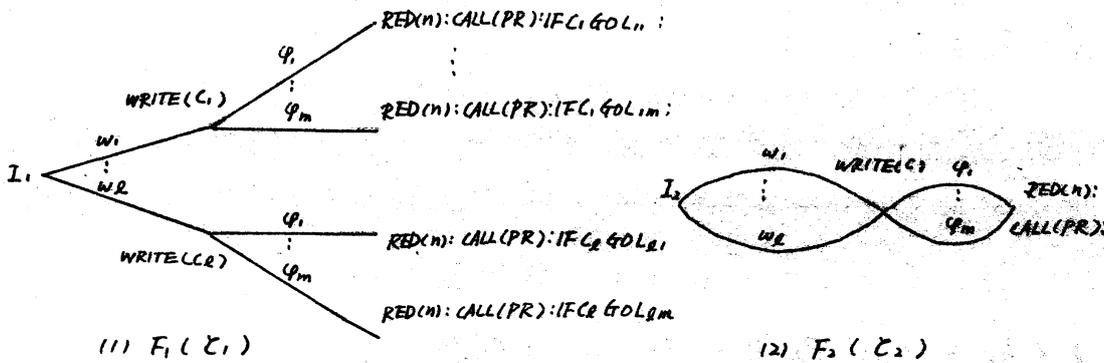


図 - 5

さて、次の二ことが証明される。

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 とおおう 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 γ 枝をつけて得られる有限オートマトンと \tilde{F}_1, \tilde{F}_2 とする。



定理 4 \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帰する。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

法のアナライズの簡単化ができる。

また、本稿では先ずは同じ部分を2度繰り返しては見ない (LR 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。いかにせよ、計算時間と記憶容量の兼ね合いの問題である。

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) 扇岩: "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. EC.* - 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究* (1972-01)