亜順序文法とその構文解析法†

落水浩一郎†† 水本 雅晴†† 豊田 順一†† 田中 幸吉††

Abstract

Regular-type sequential grammars and quasi-sequential grammars are proposed as belonging to the class of generative grammars which can describe FORTRAN IV. Regular-type sequential grammars are defined by giving some restrictions to the sequential grammars, and those restrictions are that regular-type sequential grammars have no self-embedding properties and that their parsing processes are deterministic.

Quasi-sequential grammars are defined as an extension of regular-type sequential grammars by attatching the production rules with self-embedding properties by special methods described in this paper.

The parsing processes for these grammars are executed as follows:

The set of production rules is classified into ordered subsets, and scanning-reduction process is devided into the number of quotient sets. At each process, only one subset in the set of production rules is used.

Henceforce, the size of the syntax table is much decreased.

The authors are aiming to apply this theory to the compiler-compiler for the FORTRAN-type programming languages.

1. まえがき

FORTRAN IVの文法[1]のうち，Backus Naur Form (BNF) で記述可能な部分に対応する生成文法のクラスとして，順序文法に若干の制限をつけ加えた亜順序文法を定義し，その構文解析法を論じる．(以後，FORTRAN IV の文法という表現を使用した場合は，BNFで記述可能な部分をさすことにする．) FORTRAN IV の文法構造は，順位関数をもつ順位文法のクラスで近似できることは，すでに示されている[2]．しかし，後述するように FORTRAN IV の文法は順序文法に若干の制限をつけ加えたもので表現可能であり，しかも自己埋め込み性(self-embedding property)を有する生成規則は数種類であるので，そのような生成規則の左辺の変数を疑似的な終端記号 (terminal symbol) としてとりあつかうような文法を考えておけば，この文法の生成する言語はあきらかに正則となる．(このような文法を本論文では正則的順序文法とよぶ.) しかるのちに自己埋め込み性を有する生成規則をつけ加えて，FORTRAN IV に対する文法を定義すると，文法の記述および，その構文解析は著しく簡単になる．このような文法のクラスがすなわち亜順序文法である．以下，まず2節において正則的順序文法および亜順序文法の定義を与え，3節ではその構文解析法を論じる．筆者らの提案する構文解析法においては，変数の順序関係をもとにした生成規則の集合の類別が，手続上，重要な役割を果たす．4節では亜順序文法，正則的順序文法により FORTRAN IV の文法を記述する．

2. 正則的順序文法と亜順序文法

(定義 1) 記号別 α が記号列 β を含むとき (すなわち，β が α の部分系列であるとき)，$\alpha\supset\beta$ とかく．α が β を含まないとき $\alpha\not\supset\beta$ とかく．

(定義 2) 文脈自由文法 $G=(V_N, V_T, P, S)$ にお

---

† Quasi-sequential Grammars and their Parsing Algorithms, by Koichiro OCHIMIZU, Masaharu MIZUMOTO, Junichi TOYODA, Kohkichi TANAKA (Department of Information and Computer Scinces, Faculty of Engineering Science, Osaka University)

†† 大阪大学基礎工学部情報工学科

いて，V_N の要素を $A_1, A_2, \cdots, A_N$ と順序づけたとき ($S=A_1$), 生成規則の形が $A_i\to\alpha, \alpha\in V^*(V=V_N\cup V_T)$, $\alpha\not\supset A_j, j<i$ となるとき，Gを順序文法 (sequential grammar) という[3].

(定義 3) 文脈自由文法 $G=(V_N, V_T, P, S)$ において，V_N の要素が順序づけられており ($S=A_1$), すべての生成規則が

$$A_i\to A_i\alpha, \alpha\in V^*, \alpha\not\supset A_j,\ j\leqq i \qquad (1)$$

かまたは

$$A_i\to\beta, \beta\in V^{+\dagger}, \beta\not\supset A_j,\ j\leqq i \qquad (2)$$

であるとき，生成規則の集合Pをk個の部分集合に，つぎのような手順で類別する．

(1) 左辺が $A_i(1\leqq i\leqq n)$ である生成規則の集合を $P(A_i)$ とする．

(2) 生成規則を $P(A_1), P(A_2), \cdots, P(A_n)$ の順に並べる(図1)．ただし，$A_i\to C_{ih}\alpha_{ih}$ は $P(A_i)$ の h 番目の生成規則であり，C_{ih} は生成規則の右辺における左端の記号を表わし，α_{ih} は生成規則の右辺から左端の記号をとりのぞいた記号列を表わす． また $m_1,\cdots,m_n$ はそれぞれ $P(A_1),\cdots,P(A_n)$ の要素数を表わす．

(3) P の一つの部分集合 $P(A_i)$ に対してつぎのような集合 $\boldsymbol{S}_i$ を対応させる．

$$\boldsymbol{S}_i=\{A_P \mid A_P\subset\alpha_{ih}, A_P\in V_N, 1\leqq h\leqq m_i\}$$

(4) (i) $i=1, l=1, m=1, \boldsymbol{S}=\boldsymbol{S}_1, P(1)=\phi$ とする．

(ii) もし，$A_{i+1}\in\boldsymbol{S}$ ならば，$P(m)=P(m)\cup P(A_i)$ $m=m+1, P(m)=\phi, l=l+1, \boldsymbol{S}=\boldsymbol{S}_l$ をこの順に実行し(iii)へ行く．$A_{i+1}\notin\boldsymbol{S}$ ならば，$P(m)=P(m)\cup P(A_i)$, $l=l+1, \boldsymbol{S}=\boldsymbol{S}\cup\boldsymbol{S}_l$ をこの順に実行して(iii)へ行く．

(iii) $i=i+1$, もし $i<n$ ならば (ii) へ行く．$i\geqq n$ ならば．$P(m)=P(m)\cup P(A_i)$ として止まる．このときの m の値を k とすれば，$P=P(1)\cup P(2)\cup\cdots\cup P(k)$, $P(m_i)\cap P(m_j)=\phi, m_i\neq m_j$ となる． このとき文法 G を rank(k) であるという．

$$P(A_1)\begin{cases}A_1\to C_{11}\alpha_{11}\\ \quad\vdots\\ A_1\to C_{1h}\alpha_{1h}\\ \quad\vdots\\ A_1\to C_{1m_1}\alpha_{1m_1}\end{cases}$$

$$\cdots\cdots\cdots\cdots\cdots\cdots$$

$$P(A_i)\begin{cases}A_i\to C_{i1}\alpha_{i1}\\ \quad\vdots\\ A_i\to C_{ih}\alpha_{ih}\\ \quad\vdots\\ A_i\to C_{im_i}\alpha_{im_i}\end{cases}$$

$$\cdots\cdots\cdots\cdots\cdots\cdots$$

$$P(A_n)\begin{cases}A_n\to C_{n1}\alpha_{n1}\\ \quad\vdots\\ A_n\to C_{nh}\alpha_{nh}\\ \quad\vdots\\ A_n\to C_{nm_n}\alpha_{nm_n}\end{cases}$$

図 1 生成の規則の順番づけ

Fig. 1 Ordering of Production Rules

(例 1)

$G=(V_N, V_T, P, A_1)$

$V_N=\{A_1, A_2, A_3, A_4\}$, $V_T=\{a_1, a_2, a_3, a_3\}$

$P=\{A_1\to A_1a_1A_3, A_1\to A_2, A_1\to a_2, A_2\to A_2A_4,$
$A_3\to A_3a_3A_4, A_3\to a_3, A_4\to a_4\}$

において

$P(A_1)=\{A_1\to A_1a_1A_3,\ A_1\to A_2,\ A_1\to a_2\}$, $\boldsymbol{S}_1=\{A_3\}$

$P(A_2)=\{A_2\to A_3A_4\}$, $\boldsymbol{S}_2=\{A_4\}$

$P(A_3)=\{A_3\to A_3a_3A_4, A_3\to a_3\}$, $\boldsymbol{S}_3=\{A_4\}$

$P(A_4)=\{A_4\to a_4\}$, $\boldsymbol{S}_4=\phi$

であり

$P(1)=P(A_1)\cup P(A_2), P(2)=P(A_3), P(3)=P(A_4)$

となり，G は rank (3)である．

(定義 4) rank(k) の文脈自由文法 $G=(V_N, V_T, P, A_1)$ において，つぎの (1)～(4) の条件をすべて満たすものを，**rank(*k*) の正則的順序文法**という．

(1) $\forall A_i\in V_N$ に対して，必ず生成規則 $A_i\to\alpha$, $\alpha\in V^+$ が存在し，そのうち少なくとも1つは $A_i\overset{*}{\underset{G}{\Rightarrow}}w$, $w\in V_T{}^+$ とならなければいけない．

(2) $A_i\to\alpha$, $A_j\to\alpha$, $A_i\neq A_j$, $\alpha\in V^*$ であるような生成規則の組は存在しない．

(3) 任意の m に対して，$P(m)$ の一つの要素を $A_j\to\alpha$ とするとき，$\alpha'\supset\alpha$ を満たす α' を右辺としてもつような，$P(m)$ に属する生成規則の集合を Q とするとき ($Q=\phi$ の場合もありうる)，$(P(1)\cup P(2)\cup\cdots\cup P(m))-(\{A_j\to\alpha\}\cup Q)$ により A_1 から生成される任意の文形 (sentential form) β は，$\beta\not\supset\alpha$ なる関係を満たす．

(4) $A_l\to u\in P(i)$, $A_h\to uv\in P(i)$, $u, v\in V^+$ を満たす生成規則の組は存在しない．

(定義 5) **rank (*k*+1) の亜順序文法** $G'=(V_N', V_T', P', A_1)$ をつぎのように定義する．まず rank(k) の正則的順序文法 $G=(V_N, V_T, P, A_1)$ において，$A_{i_1}, A_{i_2}, \cdots A_{i_r}, A_j\in V_N(i_1\leqq j, i_2\leqq j, \cdots, i_r\leqq j)$ であり，A_j が $\forall A_l\in V_N(l<j)$ に対して $A_l\overset{G}{\underset{*}{\not\Rightarrow}}A_j$† なる条件を満たしているとする．

† $V^+=V^*-\{\varepsilon\}$ を表わす．

† $\overset{*}{\underset{G}{\not\Rightarrow}}$ は G により生成されないという意味である．

このとき，$A_j \to \alpha_1 A_{i_1} \alpha_2 A_{i_2} \cdots \alpha_r A_{i_r} \alpha_{r+1}, \alpha_1, \alpha_2, \cdots, \alpha_{r+1} \in V^*$ なるPに属さない生成規則（ただし，$A_j \to \alpha_1 \alpha_2 \cdots \alpha_r \alpha_{r+1}$ は定義3の（1），（2）式で与えられるいずれかの形である.）を，$A_j \to \alpha_1 A_{q_1} \alpha_2 A_{q_2} \cdots \alpha_r A_{q_r} \alpha_{r+1}$ と $A_{q_1} \to \vdash A_{i_1} \dashv$, $A_{q_2} \to \vdash A_{i_2} \dashv$, $\cdots$, $A_{q_r} \to \vdash A_{i_r} \dashv$ の組に書きかえる．ここで$\vdash, \dashv \notin V_T, q_r = q_{r-1}+1, \cdots, q_2 = q_1 + 1$ であり，q_1 は V_N に属する変数の最大の添字より1だけ大きい数である．最後に $V_N' = V_N \cup \{A_{q_1}, \cdots, A_{q_r}\}, V_T' = V_T \cup \{\vdash, \dashv\} P' = P \cup \{A_j \to \alpha_1 A_{q_1} \alpha_2 \cdots \alpha_r A_{q_r} \alpha_{r+1}\} \cup \{A_{q_1} \to \vdash A_{i_1} \dashv, A_{q_r} \to \vdash A_{i_r} \dashv\}$ なる集合を構成し，$G' = (V_N', V_T', P', A_1)$ と定義する．このとき，$A_j \to \alpha_1 A_{q_1} \alpha_2 \cdots \alpha_r A_{q_r} \alpha_{r+1}$ は $P(A_j)$ に付加され，$\{A_{q_1} \to \vdash A_{i_1} \dashv, \cdots, A_{q_r} \to \vdash A_{i_r} \dashv\}$ は $P(0)$ として付加される.

［系 1］ 正則的順序文法 G により生成される言語 $L(G)$ は正則集合である.

（証 明） 定義より，正則的順序文法は自己埋め込み性をもたないので明らかである.

3. 亜順序文法の構文解析法

本節では亜順序文法の構文解析法を与える．文脈自由文法で生成される文の解析方法としては，全体に対しては，（1）直構文解析法，また部分集合に対しては，（2）文形中の隣接する記号間の順位関係を利用する方法，（3）文脈を利用する方法などがあるが[4]，本論文においては，（4）生成規則間の優先順位を利用する方法によって構文解析を行なう．すなわち，正則的順序文法に対しては，生成規則の集合Pを $P = P(1) \cup P(2) \cup \cdots \cup P(k)$ のようにk個の部分集合に類別したとき，対応する還元規則（すなわち，生成規則の左辺と右辺を入れ換えた規則）の集合 R を $R = R(1) \cup R(2) \cup \cdots \cup R(k)$ のように構成し，与えられた入力記号列をまず $R(k)$ のみを使用して，左から右に逆もどりなしに還元しながら走査する．その結果得られる文形を，こんどは $R(k-1)$ のみを使用して，再び左から右に一方向に還元しながら走査する．同様の手順をくりかえして，この過程をk回反復して解析する．最後に，$R(1)$ を使用する k 回目の走査で，文形が始記号に還元されたとき解析成功で，その他の場合は失敗である．一方，亜順序文法に対しては，自己埋め込み性を有する生成規則によって生成された部分文形は，必ず“$\vdash$”,“$\dashv$”ではさまれているので，その部分を還元する過程が付加される．すなわち，rank $(k+1)$ の亜順序文法に対する構文解析は基本的には2つの動作に分けられ，1つは文形中の“$\vdash$”,“$\dashv$”なる一対の記号の発見であり，他の1つは，正則的順序文法に対する構文解析法と同じものである.

（1） 付加した生成規則がたとえば $A_i \to \vdash A_j \dashv$, $i \geqq j$ ならば，文を左から右に最初の“$\dashv$”が見つかるまで走査する．“$\dashv$”が見つかれば，そこから左に“$\vdash$”を見つけるまで走査する．“$\vdash$”が見つかれば，その“$\vdash$”と“$\dashv$”の間にある記号列を，“$\vdash$”,“$\dashv$”を含めて正則的順序文法に対する構文解析法を適用し，A_i に還元する.

（2） （1）と同様の手順で記号対“$\vdash$”，“$\dashv$”をさがし還元する．“$\vdash$”,“$\dashv$”の対が1つもない文形が得られたら，その文形に対して正則的順序文法に対する構文解析法を適用して解析は終了する.

［定理 1］ 正則的順序文法に対する構文解析において，適用すべき還元規則は一意に定まる.

（略証） $R(k+1-i)$ の要素による還元を第 i フェーズの還元過程とよぶ．まず，各フェーズにおいて，適用すべき還元規則が一意に定まることを証明しよう．すなわち，第 i フェーズの還元過程において，定義4の（2），（3），（4）より部分既約列の情報を使うことなく，適用すべき還元規則が一意に定まることは明らかである．また，第 i フェーズの還元過程と第 j フェーズの還元過程（$j<i$）において，j フェーズで還元されるべき把手 (handle) が i フェーズで還元されるようなことは，生成規則の rank 分けの過程および，定義4の（3），（4）よりあきらかなようにおこりえない.　（証明終）

［定理 2］ 亜順序文法に対する構文解析において適用すべき還元規則は一意に定まる.

（略証） “$\vdash$”,“$\dashv$”の発見過程が一意であり，さらに定義5より“$\vdash$”と“$\dashv$”によってはさまれた，“$\vdash$”，“$\dashv$”を含む記号列は必ず $P(0)$ 中の対応する還元規則の左辺に還元されること，および定理1より明らかである.　（証明終）

以上述べた概略的な構文解析の手順は記号表，構文表および rank 表の作成方法を示し，それらを使用した構文解析の手順をフローチャート化することにより完全に形式化される.

3.1 還元規則の書き換え

まず，還元規則の集合 $R(1)$（亜順序方法の場合は $R(0)$）の要素をつぎのように表現方法を変える.

（1） 左辺の左端の記号に共通なものがあれば，その記号で還元規則をまとめる．すなわち $\alpha_1\beta \to A_i, \alpha_1$

$\beta' \to A_j, \alpha_1\beta'' \to A_k$ を $\alpha_1[\beta \to A_i | \beta' \to A_j | \beta'' \to A_k]$ とする．[ ] で囲まれたものについて同様な操作を左側に共通の記号がなくなるまでくりかえす．

(2) (1)と同様の操作を $R(2)$, $R(3), \cdots, R(k)$ についても行なう．

(例 2) (1) のような還元規則の組が，たとえば $R(1)$ であるとき，(2)のようにまとめられる．

$$(1)\ \begin{cases} \alpha_1\beta_1 r_1 \to A_1 \\ \alpha_1\beta_1 r_2 \to A_2 \\ \alpha_1\beta_3 \to A_3 \\ \alpha_4 \to A_4 \\ B \to A_5 \end{cases}$$

$$(2)\ \begin{cases} \alpha_1[\beta_1[r_1 \to A_1 | r_2 \to A_2] | \beta_3 \to A_3] \\ \alpha_4 \to A_4 \\ B \to A_5 \end{cases}$$

3.2 記号表，構文表および rank 表

以下の議論においては，表1における記号表，構文表，rank 表を参照しながら，一般的に議論を進めて行く．対象とする還元規則は **3.1** で与えた手順で書き換えられたものとする．

(1) まず $R(1)$（亜順序文法のときは $R(0)$）を対象として記号表，構文表，rank 表をつぎのように作成する．

(記号表) 記号表の SI 列に還元規則の左辺の左端の記号を入れる．左辺がその記号だけからなるとき，その行の LI 列には0を，RI 列には右辺の記号を入れる．そうでないときは，後に続く記号は構文表の SII 列に左の記号から順に入れる．その最初の行番号を記号表 LI 列に入れ，RI 列には0を入れる．

(構文表) M 列には SII 列に書かれている記号が，“→” の左隣の記号ならば K を入れ，RII 列には，その記号に対応する，“→” の右隣の記号を入れる．

“[” の直後の記号の入っている行の ALT 列には，つぎの “|” の直後の記号の行番号を入れる．(ただし “|” より先に “[” がある場合は，対応する “]” の直後の “|” の右に隣接する記号の行番号を入れる．) また一組の [ ] の中に “|” が複数個出現する場合は，**i 番目の** “|” の直後の記号の入っている行番号を，**$i-1$ 番目の** “|” の直後の記号の入っている行の ALT 列に入れる．その他の欄にはすべて0を入れる．

(rank 表) L_1 には記号表の行の最大値を入れておく．以上の手順で $R(1)$（亜順序文法のときは $R(0)$）に対する表が完成する．

(2) 以下，(1)と同様の手順で $R(2), R(3), \cdots, R(k)$ と表をつくり，$R(i)$ に対する表は $R(i-1)$ 表の下につけ加え，L_i には $R(i)$ の記号表を作ったときの行の最大値を入れておく．

表の意味を説明しよう．まず，rank 表の “L_i” の添字 i の最大値 k は文法の rank を示している．L_i 列に入っている値は，第 i 回目の走査のときは，L_{-i+k+1}（k は rank 数）列の値から，L_{-i+k} 列の値に1を加えた値までに相当する，記号表の行を使用することを示している．記号表の LI 列には，つぎに比較すべき記号の構文表における行番号が示されている．その内容が0のときには RI 列の記号に還元することを示している．構文表のM列は，読みこんだ記号と，その行の記号が一致するときの，つぎの動作を示す．すなわち0ならばさらに一記号を読みこみ，構文表のつぎの行の SII 列の記号と比較し，K ならば RII 列の記号に還元する．ALT 列は，読みこんだ記号が構文表の対応する SII 列の記号と一致しないとき，つぎに比較すべき記号の行番号を示している．例2に対する記号表，構文表および rank 表の例を表1に示す．正則

表 1 例1に対する記号表，構文表および rank 表

Table 1 The symbol table, the syntax table and the rank table for example 1

行番号 g	SI	LI	RI
1	$H_1(\alpha_1)$	1	0
2	$H_1(\alpha_4)$	l_5+1	0
3	B	0	A_5
⋮	⋮	⋮	⋮

記号表

L_0	L_1	L_2	……	L_k
0	3	……	……	……

rank 表

$H_1(\alpha)$: 記号列 α の左端の記号
$H_2(\alpha)$: 記号列 α の左から2番目の記号
$T(\alpha)$. 記号列 α の右端の記号

行番号 r	SII	M	ALT	RII
1	$H_2(\alpha_1)$	0	0	0
⋮	⋮	⋮	⋮	⋮
l_1	$T(\alpha_1)$	⋮		⋮
l_1+1	$H_1(\beta_1)$	⋮	l_4+1	⋮
⋮	⋮	⋮	0	⋮
l_2	$T(\beta_1)$	⋮	⋮	⋮
l_2+1	$H_1(\gamma_1)$	⋮	l_3+1	⋮
⋮	⋮	⋮	0	⋮
l_3	$T(\gamma_1)$	K	⋮	A_1
l_3+1	$H_1(\gamma_2)$	0	⋮	0
⋮	⋮	⋮	⋮	⋮
l_4	$T(\gamma_2)$	K	⋮	A_2
l_4+1	$H_1(\beta_3)$	0	⋮	0
⋮	⋮	⋮	⋮	⋮
l_5	$T(\beta_3)$	K	⋮	A_3
l_5+1	$H_2(\alpha_4)$	0	⋮	0
⋮	⋮	⋮	⋮	⋮
l_6	$T(\alpha_4)$	K	⋮	A_4
⋮	⋮	0	⋮	0

構文表

的順序文法に対する構文解析の手順を，フローチャート化したものを図2に示す．図2において，MEMO I, MEMO II は，読みこむときは先頭の一記号を読みこみ，書きこむときは後部に一記号を書き加えることのできる長さ可変の記憶装置であり，i フェーズの還元過程においては，図3のように部分既約列は MEMO I の後部からつぎつぎに押しこまれる．フローチャー

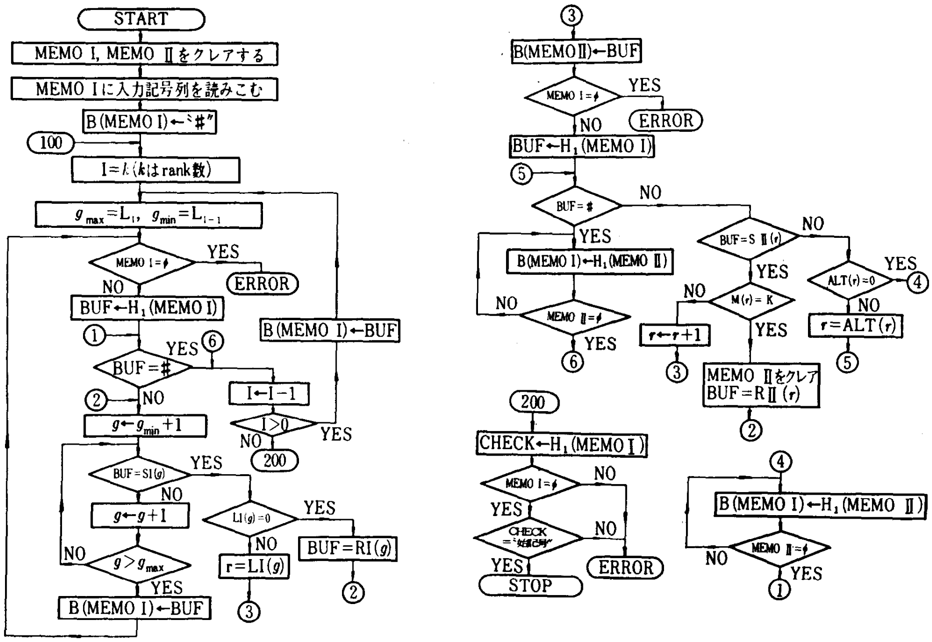
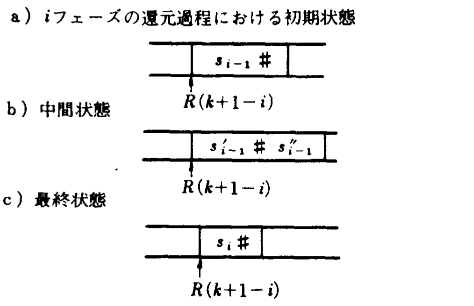


図 2 正則的順序文法に対する構文解析の手順

Fig. 2 The parsing algorithm of Regular-type sequential grammars



s_{i-1}：$i-1$ 回目の還元過程において還元された文形.
s_i　：i 回目の還元過程において還元された文形.
s'_{i-1}：i番目の還元過程における剰余列.
s''_{i-1}：i 番目の還元過程における部分既約列.
#　：エンド・マーカー. ═══ ; MEMO I.
$R(k+1-i)$：使用する還元規則の部分集合.

図 3 i フェーズの還元過程

Fig. 3 Reduction process of phase i

ト中，B(MEMO I)←BUF の形の命令は，BUF の内容を MEMO I の記号列の後部に書き加えること，また A←B の形の命令は B の内容が新しく A の内容としてセットされ，その際 B はクレアされることを意味している．また $A=B$ の形の命令は B の内容が新しく A の内容として定義されるが，B の内容はそのまま保持されること，および A←"B" の形の命令は，B 自身が A の内容としてセットされることを意味している．図 4 に亜順序文法に対する構文解析の手順をフローチャート化したものを示す．図 4 において MEMO III は MEMO I, MEMO II と同じ機能をもつ記憶装置であり，PDS I, PDS II はプッシュダウン記憶装置である．また F(PDS I)←DUM の形の命令は，DUM の内容を PDS I の先頭に押し込むことを意味している．

4. 亜順序文法によるFORTRAN IV の記述

FORTRAN IV の構成要素を付録の表 2 に示すように符号化した．表 2 において，$A_i(1\leqq i\leqq 107)$ は変数，$a_j(1\leqq j\leqq 84)$ は終端記号をそれぞれ表わす．A_{103} から A_{107} までは FORTRAN IV を亜順序文法で記述するために新しく導入した変数である．a_1 から a_{49} までは，FORTRAN における "END"，"BLOCK DATA" などの決り文句はそれ自体を終端記号としてとりあつかうこと，a_{50} から a_{72} までは，本来は変数として定義すべき<変数名>，<配列要素名>などの構成要素を，$A\to\alpha$, $B\to\alpha$ なる生成規則の組の出

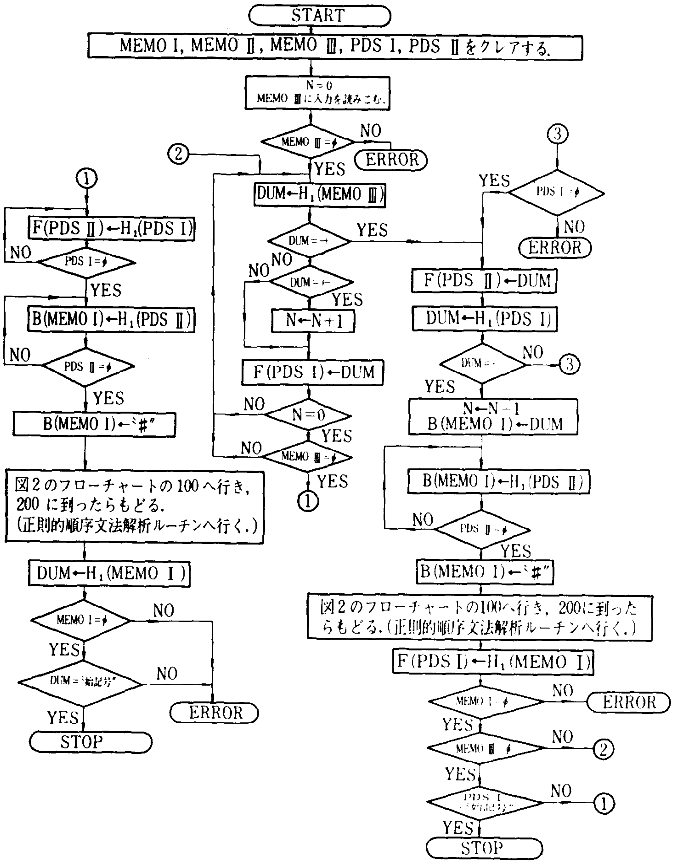


図 4 亜順序文法に対する構文解析の手順

Fig. 4 The parsing algorithm of Quasi-sequential grammars

現を生じさせないために終端記号としてとりあつかうことを意味している．a_{83}, a_{84} は自己埋め込み性を有する生成規則の処理のために新しく導入した終端記号である．以上の記法を用いて FORTRAN IV の文法を BNF で記述し，さらにそれらを生成規則の形に書きなおしたものを表 3 に示す．以下表 2，表 3 に記述された FORTRAN 文法のみを対象として，正則的順序文法および亜順序文法との関連性を述べる．

4.1 $G_{F_1}=(V_N, V_T, P, A_1)$, $V_N=\{A_1, \cdots, A_{102}\}$, $V_T=\{a_1, \cdots, a_{82}, A_{103}, \cdots, A_{107}\}$, $P=\{P_1, \cdots, P_{255}\}$ なる文法 G_{F_1} を考えれば，P_1 から P_{255} までの生成規則はすべて定義 3 の条件を満たしており，rank (26) である．

4.2 G_{F_1} は rank (26) であるが正則的順序文法ではない．なぜならば，

（i） 定義 4 の条件 2 を満たさない生成規則の組がつぎにあげるように 8 組存在する．

① $(A_{27}\to a_{50},\ A_{49}\to a_{50},\ A_{53}\to a_{50},\ A_{57}\to a_{50},\ A_{60}\to a_{50},\ A_{64}\to a_{50},\ A_{73}\to a_{50},\ A_{79}\to a_{50},\ A_{86}\to a_{50},\ A_{87}\to a_{50})$.

② $(A_{49}\to a_{51},\ A_{64}\to a_{51},\ A_{73}\to a_{51},\ A_{79}\to a_{51},\ A_{86}\to a_{51},\ A_{87}\to a_{51})$.

③ $(A_{49}\to a_{53},\ A_{53}\to a_{53},\ A_{57}\to a_{53},\ A_{60}\to a_{53},\ A_{73}\to a_{53},\ A_{87}\to a_{53})$.

④ $(A_{50}\to a_{57},\ A_{73}\to a_{57},\ A_{86}\to a_{57})$.

⑤ $(A_{50}\to a_{59},\ A_{73}\to a_{59})$.

⑥ $(A_{53}\to a_{61},\ A_{73}\to a_{61})$.

⑦ $(A_{57}\to a_{63},\ A_{60}\to a_{63})$.

⑧ $(A_{79}\to a_{71},\ A_{86}\to a_{71})$.

（ii） 定義 4 の条件 4 を満たさない生成規則の組が存在する．①$(A_{24}\to a_6a_{76}a_{52}a_{77},\ A_{24}\to a_6a_{76}a_{52}a_{77}A_{75})$

の組が $P(6)$ 存在する．②(イ)($A_{30}\to A_{45}$, $A_{45}\to A_{45}a_{74}A_{46}$), (ロ) ($A_{31}\to a_{10}a_{60}$, $A_{31}\to a_{10}a_{60}a_{76}A_{52}a_{77}$), (ハ) ($A_{41}\to a_{23}$, $A_{41}\to a_{23}a_{67}$), (ニ) ($A_{42}\to a_{30}$, $A_{42}\to a_{30}a_{67}$) (ホ) ($A_{44}\to a_{29}a_{60}$, $A_{44}\to a_{29}a_{60}a_{76}A_{72}a_{77}$) の５組が $P(8)$ に存在する．③ ($A_{65}\to A_{66}$, $A_{65}\to A_{66}a_{74}a_{65}$, $A_{65}\to A_{66}a_{74}a_{66}$) の組が $P(15)$ に存在する．

(iii) 定義４の条件３を満たさない生成規則の組が存在する．ただし(i)と重複するものは除く．①($A_{20}\to a_{50}a_{81}A_{82}$, $A_{21}\to a_{50}a_{81}A_{74}$, $A_{27}\to a_{50}$, $A_{49}\to a_{50}$, $A_{53}\to a_{50}$, $A_{57}\to a_{50}$, $A_{60}\to a_{50}$, $A_{64}\to a_{50}$, $A_{73}\to a_{50}$, $A_{79}\to a_{50}$ $A_{86}\to a_{50}$, $A_{87}\to a_{50}$), ② ($A_{20}\to a_{51}a_{81}A_{82}$, $A_{21}\to a_{51}a_{81}A_{74}$, $A_{49}\to a_{51}$, $A_{64}\to a_{51}$, $A_{73}\to a_{51}$, $A_{79}\to a_{51}$, $A_{86}\to a_{51}$, $A_{87}\to a_{51}$), ③ ($A_{24}\to a_6a_{76}a_{52}a_{74}a_{53}a_{77}A_{75}$, $A_{25}\to a_7a_{76}a_{52}a_{74}a_{53}a_{77}A_{75}$, $A_{49}\to a_{53}$, $A_{53}\to a_{53}$, $A_{57}\to a_{53}$, $A_{60}\to a_{53}$, $A_{73}\to a_{53}$, $A_{87}\to a_{53}$), ④($A_{26}\to a_{55}a_{76}A_{27}a_{77}a_{81}A_{82}$, $A_{26}\to a_{55}a_{76}A_{27}a_{77}a_{81}A_{74}$, $A_{57}\to a_{55}$), ⑤ ($A_{54}\to A_{54}a_{74}a_{63}$, $A_{54}\to a_{17}a_{63}$, $A_{60}\to a_{63}$), ⑥ ($A_{50}\to a_{58}a_{73}a_{57}$, $A_{73}\to a_{57}$, $A_{86}\to a_{57}$), ⑦ $A_{55}\to a_{18}a_{61}$, $A_{73}\to a_{61}$), ⑧ ($A_{50}\to a_{58}a_{73}a_{59}$, $A_{73}\to a_{59}$), ⑨ ($A_{46}\to A_{47}a_{75}$, $A_{47}\to A_{48}a_{75}A_{50}$, $A_{59}\to a_{75}a_{69}a_{75}A_{60}$, $A_{59}\to a_{75}a_{75}A_{60}$, $A_{84}\to A_{84}a_{75}A_{85}$, $A_{90}\to a_{75}$).

4.3 問題点(i)(ii)(iii)に対する解決法

(i)に対しては終端記号 a_{50}, a_{51}, a_{53}, a_{57}, a_{59}, a_{61}, a_{63}, a_{71} に構文解析の前処理として語彙解析ルーチンなどにおいて以下に述べるように適当なコントロール・シンボル ($\#^1, \cdots, \#^{10}$) を付加することによって，上記の各終端記号を区別することが可能である．すなわち算術式中で使用される a_{50}, a_{51}, a_{71} は $a_{50}\#^1$, $a_{51}\#^1$, $a_{71}\#^1$. DATA文中で使用される a_{50}, a_{51}, a_{53}, a_{57}, a_{59} は $a_{50}\#^2$, $a_{51}\#^2$, $a_{53}\#^2$, $a_{57}\#^2$, $a_{59}\#^2$. 以下に同様にして EQUIVALENCE 文中で使用される a_{50}, a_{51} には $\#^3$, COMMON 文中で使用される a_{50}, a_{53}, a_{63} には $\#^4$, CALL 文中で使用される a_{50}, a_{51}, a_{53}, a_{57}, a_{59}, a_{61} には $\#^5$, 型宣言行中で使用される a_{50}, a_{53}, a_{60} には $\#^6$, READ 文, WRITE 文における入出力並び中で使用される a_{50}, a_{51}, a_{53} には $\#^7$, FUNCTION 行または SUBROUTINE 行中で使用される a_{50}, a_{53}, a_{61} には $\#^8$, 文関数定義行中で使用され算術式または論理式中で使用されない a_{50} には $\#^9$, 論理式中で使用され算術式中で使用されない a_{50}, a_{51}, a_{71} には $\#^{10}$ を付加することによって各終端記号を区別する．このとき $A_i\to\alpha$, $A_j\to\alpha$ なる生成規則の組は $A_i\to\alpha\#^l$, $A_j\to\alpha\#^m$ $(1\leq l, m\leq 10, l\neq m)$ として構文表に入れる．

(ii)の問題点はつぎのようにして解決できる．

①，②―(ロ)(ハ)(ニ)(ホ)，③の組が生じた原因は，"入出力並び"の存在する READ 文と存在しないもの，"仮引数"の存在するサブルーチンと存在しないもの，"8進数"を付加した STOP 文, PAUSE 文とそうでないもの，"実引数"の存在する CALL 文と存在しないもの，増分パラメーターをもつ DO 文とそうでないものの存在である．この解決策としては後者の型の文に構文解析の前処理として適当なコントロールシンボル¢を付加することである．これらはいずれも $A_l\to u$, $A_h\to uv$ なる生成規則の組を $A_l\to u¢$, $A_h\to uv$ と書き換えて構文解析を行なうことに相当する．②―(イ)に関しては，$A_{30}\to A_{45}$, $A_{45}\to A_{45}a_{74}A_{46}$, $A_{45}\to a_9A_{46}$ の生成規則の組を，等価な $A_{30}\to A_{30}a_{74}A_{46}$, $A_{30}\to a_9A_{46}$ の組に変更することにより解決できる．

(iii)の問題点はつぎのように解決できる．

①〜⑧までの生成規則の組は(i)に対する解決策として示した前処理後はつぎのような組として表現される．①′ ($A_{20}\to a_{50}a_{81}A_{82}$, $A_{21}\to a_{50}a_{81}A_{74}$, $A_{27}\to a_{50}\#^9$, $A_{49}\to a_{50}\#^2$, $A_{53}\to a_{50}\#^8$, $A_{57}\to a_{50}\#^6$, $A_{60}\to a_{50}\#^4$, $A_{64}\to a_{50}\#^3$, $A_{73}\to a_{50}\#^5$, $A_{79}\to a_{50}\#10$, $A_{86}\to a_{50}\#^1$, $A_{87}\to a_{50}\#^7$), ②′($A_{20}\to a_{51}a_{81}A_{82}$, $A_{21}\to a_{51}a_{81}A_{74}$, $A_{49}\to a_{51}\#^2$, $A_{64}\to a_{51}\#^3$, $A_{73}\to a_{51}\#^5$, $A_{79}\to a_{51}\#^{10}$, $A_{86}\to a_{51}\#^1$, $A_{87}\to a_{51}\#^7$), ③′ ($A_{24}\to a_6a_{76}a_{52}a_{74}a_{53}a_{77}A_{75}$, $A_{25}\to a_7a_{76}a_{52}a_{74}a_{53}a_{77}A_{75}$, $A_{49}\to a_{53}\#^2$, $A_{53}\to a_{53}\#^8$, $A_{57}\to a_{53}\#^6$, $A_{60}\to a_{53}\#^4$, $A_{73}\to a_{53}\#^5$, $A_{87}\to a_{53}\#^7$(, ④′($A_{26}\to a_{55}\#^9a_{76}A_{27}a_{77}a_{81}A_{82}$, $A_{26}\to a_{55}\#^9a_{76}A_{27}a_{77}a_{81}A_{74}$, $A_{57}\to a_{55}\#^6$), ⑤′($A_{54}\to A_{54}a_{74}a_{63}\#^6$, $A_{54}\to a_{17}a_{63}\#^6$, $A_{60}\to a_{63}\#^4$), ⑥′($A_{50}\to a_{58}a_{73}a_{57}\#^2$, $A_{73}\to a_{57}\#^5$, $A_{86}\to a_{57}\#^1$), ⑦′($A_{55}\to a_{18}a_{61}$, $A_{73}\to a_{61}\#^5$), ⑧′($A_{50}\to a_{58}a_{73}a_{59}\#^2$, $A_{73}\to a_{59}\#^5$). このとき ①′〜⑧′ までの組はもはや定義に反していない．⑨に関しては $A_{89}\to A_{89}A_{90}A_{91}$, $A_{89}\to A_{90}A_{91}$, $A_{90}\to a_{75}$ の生成規則の組を $A_{89}\to A_{89}a_{75}A_{91}$, $A_{89}\to a_{75}A_{91}$ の組に書き換える．

4.4 **4.1** で与えた文法 G_{F_1} に **4.3** で述べた修正を加えた文法 $G_{F_2}=(V_N', V_T', P', A_1)$ を考えると，G_{F_2} は rank (26) の正則的順序文法である．ただし $V_N'=V_N$, $V_T'=V_T\cup${コントロール・シンボル}, P' は P のうち **4.2** でリストしたものを **4.3** のように修正したものである．

4.5 **4.4** で与えた文法 G_{F_2} をもとにして FORTRAN IV の文法を亜順序文法 G_F で記述するとつぎのようになる．$G_F=(V_N'', V_T'', P'', A_1)$, $V_N''=V_N'\cup\{A_{103}, \cdots, A_{107}\}$, $V_T''=V_T'\cup\{a_{83}, a_{84}\}-\{A_{103}, \cdots, A_{107}\}$, $P''=P'\cup\{P_{256}, \cdots, P_{260}\}$. このとき G_F は

表 2 FORTRAN IV の構成要素

Tablə 2 Syntactic elements of FORTRAN IV

符号	構成要素	符号	構成要素	符号	構成要素	符号	構成要素
A_{1}	〈実行可能プログラム〉	A_{49}	〈DATA 名〉	A_{97}	〈L format code〉	a_{38}	.LE.
A_{2}	〈主プログラム〉	A_{50}	〈DATA〉	A_{98}	〈A format code〉	a_{39}	.NE.
A_{3}	〈副プログラム>〉	A_{51}	〈型宣言文〉	A_{99}	〈H format code〉	a_{40}	F
A_{4}	〈SUBROUTINE 副プログラム〉	A_{52}	〈仮引数並び〉	A_{100}	〈X format code〉	a_{41}	E
A_{5}	〈関数副プログラム〉	A_{53}	〈仮引数〉	A_{101}	〈ケタ移動子〉	a_{42}	G
A_{6}	〈初期値設定副プログラム〉	A_{54}	〈DIMENSION 文〉	A_{102}	〈書式仕様要素群〉	a_{43}	C
A_{7}	〈プログラム本体〉	A_{55}	〈EXTERNAL 文〉	A_{103}	〈論理式埋め込み〉	a_{44}	I
A_{8}	〈プログラム部分〉	A_{56}	〈形容詞〉	A_{104}	〈算術式埋め込み〉	a_{45}	L
A_{9}	〈宣言部 I〉	A_{57}	〈型宣言要素〉	A_{105}	〈入出力並び埋め込み I〉	a_{46}	A
A_{10}	〈宣言部〉	A_{58}	〈COMMON 文〉	A_{106}	〈入出力並び埋め込み II〉	a_{47}	H
A_{11}	〈初期値設定副プログラム部分〉	A_{59}	〈COMMON 要素〉	A_{107}	〈書式仕様I埋め込み〉	a_{48}	X
A_{12}	〈初期値設定副プログラム宣言部〉	A_{60}	〈ブロック要素〉	a_{1}	END	a_{49}	P
A_{13}	〈宣言行〉	A_{61}	〈EQUIVALENCE 文〉	a_{2}	BLOCK DATA	a_{50}	〈変数名〉
A_{14}	〈実行行〉	A_{62}	〈EQUIVALENCE 要素〉	a_{3}	IF	a_{51}	〈配列要素名〉
A_{15}	〈実行文〉	A_{63}	〈EQUIVALENCE 要素前部〉	a_{4}	RETURN	a_{52}	〈I/O 装置 NO.〉
A_{16}	〈実行文 I〉	A_{64}	〈EQUIVALENCE 並び要素〉	a_{5}	CONTINUE	a_{53}	〈配列名〉
A_{17}	〈論理 IF 文〉	A_{65}	〈DO 制御部 I〉	a_{6}	READ	a_{54}	〈FORMAT 文 NO.〉
A_{18}	〈実行文 II〉	A_{66}	〈DO 制御部 II〉	a_{7}	WRITE	a_{55}	〈文関数名〉
A_{19}	〈代入文〉	A_{67}	〈DO 制御部 III〉	a_{8}	FORMAT	a_{56}	〈磁気テープ装置 NO.〉
A_{20}	〈算術代入文〉	A_{68}	〈無条件 GO TO 文〉	a_{9}	DATA	a_{57}	〈定数〉
A_{21}	〈論理代入文〉	A_{69}	〈計算型 GO TO 文〉	a_{10}	SUBROUTINE	a_{58}	〈反復数〉
A_{22}	〈算術 IF 文〉	A_{70}	〈割当て型 GO TO 文〉	a_{11}	FUNCTION	a_{59}	〈文字定数〉
A_{23}	〈入出力文〉	A_{71}	〈文 NO. 並び〉	a_{12}	INTEGER	a_{60}	〈サブルーチン名〉
A_{24}	〈READ 文〉	A_{72}	〈実引数並び〉	a_{13}	REAL	a_{61}	〈外部手続き名〉
A_{25}	〈WRITE 文〉	A_{73}	〈実引数〉	a_{14}	DOUBLE PLECISION	a_{62}	〈関数名〉
A_{26}	〈文関数定義行〉	A_{74}	〈論理式〉	a_{15}	COMPLEX	a_{63}	〈配列宣言子〉
A_{27}	〈変数名並び〉	A_{75}	〈入出力並び〉	a_{16}	LOGICAL	a_{64}	〈端末文 NO〉
A_{28}	〈FORMAT 行〉	A_{76}	〈書式仕様 I〉	a_{17}	DIMENSION	a_{65}	〈整定数〉
A_{29}	〈FORMAT 文〉	A_{77}	〈論理項〉	a_{18}	EXTERNAL	a_{66}	〈整数型変数名〉
A_{30}	〈DATA 行〉	A_{78}	〈論理因子〉	a_{19}	COMMON	a_{67}	〈8進数〉
A_{31}	〈SUBROUTINE 行〉	A_{79}	〈論理一次子〉	a_{20}	EQUIVALENCE	a_{68}	〈文 NO.〉
A_{32}	〈FUNCTION 行〉	A_{80}	〈関係式〉	a_{21}	DO	a_{69}	〈ブロック名〉
A_{33}	〈DIMENSION 行〉	A_{81}	〈関係演算子〉	a_{22}	GO TO	a_{70}	〈文字欄〉
A_{34}	〈型宣言行〉	A_{82}	〈算術式〉	a_{23}	STOP	a_{71}	〈関数の引用〉
A_{35}	〈COMMON 行〉	A_{83}	〈符号〉	a_{24}	ASSIGN	a_{72}	〈論理定数〉
A_{36}	〈EQUIVALENCE 行〉	A_{84}	〈項〉	a_{25}	TO	a_{73}	*
A_{37}	〈EXTERNAL 行〉	A_{85}	〈因子〉	a_{26}	REWIND	a_{74}	,
A_{38}	〈DO 文〉	A_{86}	〈一次子〉	a_{27}	BACKSPACE	a_{75}	/
A_{39}	〈GO TO文〉	A_{87}	〈入出力並び要素〉	a_{28}	END FILE	a_{76}	(
A_{40}	〈ASSIGN 文〉	A_{88}	〈DO 型並び〉	a_{29}	CALL	a_{77}	)
A_{41}	〈STOP 文〉	A_{89}	〈書式仕様〉	a_{30}	PAUSE	a_{78}	—
A_{42}	〈PAUSE 文〉	A_{90}	〈Slush 列〉	a_{31}	.OR.	a_{79}	**
A_{43}	〈補助入出力文〉	A_{91}	〈書式仕様要素〉	a_{32}	.AND.	a_{80}	.
A_{44}	〈CALL 文〉	A_{92}	〈F format code〉	a_{33}	.NOT.	a_{81}	=
A_{45}	〈DATA 文〉	A_{93}	〈E format code〉	a_{34}	.GT.	a_{82}	+
A_{46}	〈DATA 要素〉	A_{94}	〈G format code〉	a_{35}	.LT.	a_{83}	⊢
A_{47}	〈DATA 要素前部〉	A_{95}	〈D format code〉	a_{36}	.EQ.	a_{84}	⊣
A_{48}	〈DATA 名並び〉	A_{96}	〈I format code〉	a_{37}	.GE.		

表 3
Table 3 FORTAN IV の Production Rules of FORTRAN IV

符号	生成規則	符号	生成規則	符号	生成規則	符号	生成規則
P_{1}	$A_{1}\rightarrow A_{1}A_{3}$	P_{66}	$A_{24}\rightarrow a_{6}a_{76}a_{52}a_{74}a_{54}a_{77}A_{75}$	P_{131}	$A_{57}\rightarrow a_{53}$	P_{196}	$A_{83}\rightarrow a_{82}$
P_{2}	$A_{1}\rightarrow A_{2}$	P_{67}	$A_{25}\rightarrow a_{7}a_{76}a_{52}a_{77}$	P_{132}	$A_{57}\rightarrow a_{55}$	P_{197}	$A_{83}\rightarrow a_{78}$
P_{3}	$A_{2}\rightarrow A_{7}$	P_{68}	$A_{25}\rightarrow a_{7}a_{76}a_{52}a_{74}a_{53}a_{77}A_{75}$	P_{133}	$A_{57}\rightarrow a_{63}$	P_{198}	$A_{84}\rightarrow A_{84}a_{73}A_{85}$
P_{4}	$A_{3}\rightarrow A_{4}$	P_{69}	$A_{25}\rightarrow a_{7}a_{76}a_{52}a_{74}a_{54}a_{77}A_{75}$	P_{134}	$A_{58}\rightarrow A_{58}A_{59}$	P_{199}	$A_{84}\rightarrow A_{84}a_{75}A_{85}$
P_{5}	$A_{3}\rightarrow A_{5}$	P_{70}	$A_{26}\rightarrow a_{55}a_{76}A_{27}a_{77}a_{81}A_{82}$	P_{135}	$A_{58}\rightarrow a_{19}A_{59}$	P_{200}	$A_{84}\rightarrow A_{85}$
P_{6}	$A_{3}\rightarrow A_{6}$	P_{71}	$A_{26}\rightarrow a_{55}a_{76}A_{27}a_{77}a_{81}A_{74}$	P_{136}	$A_{59}\rightarrow A_{59}a_{74}A_{60}$	P_{201}	$A_{85}\rightarrow A_{85}a_{79}A_{86}$
P_{7}	$A_{4}\rightarrow A_{31}A_{7}$	P_{72}	$A_{27}\rightarrow A_{27}a_{74}a_{50}$	P_{137}	$A_{59}\rightarrow a_{75}a_{69}a_{75}A_{60}$	P_{202}	$A_{85}\rightarrow A_{86}$
P_{8}	$A_{5}\rightarrow A_{32}A_{7}$	P_{73}	$A_{27}\rightarrow a_{50}$	P_{138}	$A_{59}\rightarrow a_{75}a_{75}A_{60}$	P_{203}	$A_{86}\rightarrow a_{57}$
P_{9}	$A_{6}\rightarrow A_{11}a_{1}$	P_{74}	$A_{28}\rightarrow a_{54}A_{29}$	P_{139}	$A_{59}\rightarrow A_{60}$	P_{204}	$A_{86}\rightarrow a_{80}$
P_{10}	$A_{7}\rightarrow A_{8}a_{1}$	P_{75}	$A_{29}\rightarrow a_{8}a_{76}A_{76}a_{77}$	P_{140}	$A_{60}\rightarrow a_{50}$	P_{205}	$A_{86}\rightarrow a_{51}$
P_{11}	$A_{8}\rightarrow A_{8}A_{14}$	P_{76}	$A_{30}\rightarrow A_{45}$	P_{141}	$A_{60}\rightarrow a_{52}$	P_{206}	$A_{86}\rightarrow a_{71}$
P_{12}	$A_{8}\rightarrow A_{8}A_{28}$	P_{77}	$A_{31}\rightarrow a_{10}a_{60}a_{76}A_{52}a_{77}$	P_{142}	$A_{60}\rightarrow a_{63}$	P_{207}	$A_{86}\rightarrow a_{76}A_{104}a_{77}$
P_{13}	$A_{8}\rightarrow A_{8}A_{30}$	P_{78}	$A_{31}\rightarrow a_{10}a_{60}$	P_{143}	$A_{61}\rightarrow A_{61}a_{74}A_{62}$	P_{208}	$A_{87}\rightarrow a_{50}$
P_{14}	$A_{8}\rightarrow A_{9}A_{14}$	P_{79}	$A_{32}\rightarrow a_{11}a_{62}a_{76}A_{52}a_{77}$	P_{144}	$A_{61}\rightarrow a_{20}A_{62}$	P_{209}	$A_{87}\rightarrow a_{53}$
P_{15}	$A_{8}\rightarrow A_{10}A_{14}$	P_{80}	$A_{32}\rightarrow A_{56}a_{11}a_{62}a_{76}A_{52}a_{77}$	P_{145}	$A_{62}\rightarrow A_{63}a_{77}$	P_{210}	$A_{87}\rightarrow a_{51}$
P_{16}	$A_{9}\rightarrow A_{9}A_{26}$	P_{81}	$A_{33}\rightarrow A_{54}$	P_{146}	$A_{63}\rightarrow A_{63}a_{74}A_{64}$	P_{211}	$A_{87}\rightarrow A_{88}$
P_{17}	$A_{9}\rightarrow A_{9}A_{30}$	P_{82}	$A_{34}\rightarrow A_{51}$	P_{147}	$A_{63}\rightarrow a_{76}A_{64}a_{74}A_{64}$	P_{212}	$A_{88}\rightarrow a_{76}A_{105}a_{74}A_{106}a_{77}$
P_{18}	$A_{9}\rightarrow A_{9}A_{28}$	P_{83}	$A_{35}\rightarrow A_{58}$	P_{148}	$A_{64}\rightarrow a_{50}$	P_{213}	$A_{89}\rightarrow A_{89}A_{90}A_{91}$
P_{19}	$A_{9}\rightarrow A_{10}A_{26}$	P_{84}	$A_{36}\rightarrow A_{61}$	P_{149}	$A_{64}\rightarrow a_{51}$	P_{214}	$A_{89}\rightarrow A_{89}a_{74}A_{91}$
P_{20}	$A_{9}\rightarrow A_{10}A_{30}$	P_{85}	$A_{37}\rightarrow A_{55}$	P_{150}	$A_{65}\rightarrow A_{66}$	P_{215}	$A_{89}\rightarrow A_{91}$
P_{21}	$A_{9}\rightarrow A_{10}A_{28}$	P_{86}	$A_{38}\rightarrow a_{21}a_{64}A_{65}$	P_{151}	$A_{65}\rightarrow A_{66}a_{74}a_{65}$	P_{216}	$A_{89}\rightarrow A_{90}A_{91}$
P_{22}	$A_{9}\rightarrow A_{26}$	P_{87}	$A_{39}\rightarrow A_{68}$	P_{152}	$A_{65}\rightarrow A_{66}a_{74}a_{66}$	P_{217}	$A_{90}\rightarrow a_{75}$
P_{23}	$A_{9}\rightarrow A_{30}$	P_{88}	$A_{39}\rightarrow A_{69}$	P_{153}	$A_{66}\rightarrow A_{67}a_{74}a_{65}$	P_{218}	$A_{91}\rightarrow A_{102}$
P_{24}	$A_{9}\rightarrow A_{28}$	P_{89}	$A_{39}\rightarrow A_{70}$	P_{154}	$A_{66}\rightarrow A_{67}a_{74}a_{66}$	P_{219}	$A_{91}\rightarrow a_{65}A_{102}$
P_{25}	$A_{10}\rightarrow A_{10}A_{13}$	P_{90}	$A_{40}\rightarrow a_{24}a_{68}a_{25}a_{66}$	P_{155}	$A_{67}\rightarrow a_{66}a_{81}a_{65}$	P_{220}	$A_{91}\rightarrow A_{92}$
P_{26}	$A_{10}\rightarrow A_{13}$	P_{91}	$A_{41}\rightarrow a_{23}$	P_{156}	$A_{67}\rightarrow a_{66}a_{81}a_{66}$	P_{221}	$A_{91}\rightarrow A_{93}$
P_{27}	$A_{11}\rightarrow A_{11}A_{30}$	P_{92}	$A_{41}\rightarrow a_{23}a_{67}$	P_{157}	$A_{68}\rightarrow a_{22}a_{68}$	P_{222}	$A_{91}\rightarrow A_{94}$
P_{28}	$A_{11}\rightarrow A_{12}A_{30}$	P_{93}	$A_{42}\rightarrow a_{30}$	P_{158}	$A_{69}\rightarrow a_{22}a_{76}A_{71}a_{77}a_{74}a_{66}$	P_{223}	$A_{91}\rightarrow A_{95}$
P_{29}	$A_{12}\rightarrow A_{12}A_{13}$	P_{94}	$A_{42}\rightarrow a_{30}a_{67}$	P_{159}	$A_{70}\rightarrow a_{22}a_{66}a_{74}a_{76}A_{71}a_{77}$	P_{224}	$A_{91}\rightarrow A_{96}$
P_{30}	$A_{12}\rightarrow a_{2}$	P_{95}	$A_{43}\rightarrow a_{26}a_{56}$	P_{160}	$A_{71}\rightarrow A_{71}a_{74}a_{68}$	P_{225}	$A_{91}\rightarrow A_{97}$
P_{31}	$A_{13}\rightarrow A_{33}$	P_{96}	$A_{43}\rightarrow a_{27}a_{56}$	P_{161}	$A_{71}\rightarrow a_{68}a_{74}a_{68}$	P_{226}	$A_{91}\rightarrow A_{98}$
P_{32}	$A_{13}\rightarrow A_{34}$	P_{97}	$A_{43}\rightarrow a_{28}a_{56}$	P_{162}	$A_{72}\rightarrow A_{72}a_{74}A_{73}$	P_{227}	$A_{91}\rightarrow A_{99}$
P_{33}	$A_{13}\rightarrow A_{35}$	P_{98}	$A_{44}\rightarrow a_{29}a_{60}$	P_{163}	$A_{72}\rightarrow A_{73}$	P_{228}	$A_{91}\rightarrow A_{100}$
P_{34}	$A_{13}\rightarrow A_{36}$	P_{99}	$A_{44}\rightarrow a_{29}a_{60}a_{76}A_{72}a_{77}$	P_{164}	$A_{73}\rightarrow a_{57}$	P_{229}	$A_{92}\rightarrow A_{101}a_{65}a_{40}a_{65}a_{80}a_{65}$
P_{35}	$A_{13}\rightarrow A_{37}$	P_{100}	$A_{45}\rightarrow A_{45}a_{74}A_{46}$	P_{165}	$A_{73}\rightarrow a_{50}$	P_{230}	$A_{92}\rightarrow A_{101}a_{40}a_{65}a_{80}a_{65}$
P_{36}	$A_{14}\rightarrow A_{15}$	P_{101}	$A_{45}\rightarrow a_{9}A_{46}$	P_{166}	$A_{73}\rightarrow a_{53}$	P_{231}	$A_{92}\rightarrow a_{65}a_{40}a_{65}a_{80}a_{65}$
P_{37}	$A_{14}\rightarrow a_{68}A_{15}$	P_{102}	$A_{46}\rightarrow A_{47}a_{75}$	P_{167}	$A_{73}\rightarrow a_{51}$	P_{232}	$A_{92}\rightarrow a_{40}a_{65}a_{80}a_{65}$
P_{38}	$A_{15}\rightarrow A_{16}$	P_{103}	$A_{47}\rightarrow A_{47}a_{74}A_{50}$	P_{168}	$A_{73}\rightarrow a_{61}$	P_{233}	$A_{93}\rightarrow A_{101}a_{65}a_{41}a_{65}a_{80}a_{65}$
P_{39}	$A_{15}\rightarrow A_{18}$	P_{104}	$A_{47}\rightarrow A_{48}a_{75}A_{50}$	P_{169}	$A_{73}\rightarrow a_{59}$	P_{234}	$A_{93}\rightarrow A_{101}a_{41}a_{65}a_{80}a_{65}$
P_{40}	$A_{16}\rightarrow A_{17}$	P_{105}	$A_{48}\rightarrow A_{48}a_{74}A_{49}$	P_{170}	$A_{74}\rightarrow A_{74}a_{31}A_{77}$	P_{235}	$A_{93}\rightarrow a_{65}a_{41}a_{65}a_{80}a_{65}$
P_{41}	$A_{16}\rightarrow A_{38}$	P_{106}	$A_{48}\rightarrow A_{49}$	P_{171}	$A_{74}\rightarrow A_{77}$	P_{236}	$A_{93}\rightarrow a_{41}a_{65}a_{80}a_{65}$
P_{42}	$A_{17}\rightarrow a_{3}a_{76}A_{74}a_{77}A_{18}$	P_{107}	$A_{49}\rightarrow a_{50}$	P_{172}	$A_{75}\rightarrow A_{75}a_{74}A_{87}$	P_{237}	$A_{94}\rightarrow A_{101}a_{65}a_{42}a_{65}a_{80}a_{65}$
P_{43}	$A_{18}\rightarrow A_{19}$	P_{108}	$A_{49}\rightarrow a_{53}$	P_{173}	$A_{75}\rightarrow A_{87}$	P_{238}	$A_{94}\rightarrow A_{101}a_{42}a_{65}a_{80}a_{65}$
P_{44}	$A_{18}\rightarrow A_{39}$	P_{109}	$A_{49}\rightarrow a_{51}$	P_{174}	$A_{76}\rightarrow A_{76}A_{90}$	P_{239}	$A_{94}\rightarrow a_{65}a_{42}a_{65}a_{80}a_{65}$
P_{45}	$A_{18}\rightarrow A_{22}$	P_{110}	$A_{50}\rightarrow a_{57}$	P_{175}	$A_{76}\rightarrow A_{89}$	P_{240}	$A_{94}\rightarrow a_{42}a_{65}a_{80}a_{65}$
P_{46}	$A_{18}\rightarrow A_{40}$	P_{111}	$A_{50}\rightarrow a_{58}a_{73}a_{57}$	P_{176}	$A_{77}\rightarrow A_{77}a_{32}A_{78}$	P_{241}	$A_{95}\rightarrow A_{101}a_{65}a_{43}a_{65}a_{80}a_{65}$
P_{47}	$A_{18}\rightarrow a_{4}$	P_{112}	$A_{50}\rightarrow a_{59}$	P_{177}	$A_{77}\rightarrow A_{78}$	P_{242}	$A_{95}\rightarrow A_{101}a_{43}a_{65}a_{80}a_{65}$
P_{48}	$A_{18}\rightarrow a_{5}$	P_{113}	$A_{50}\rightarrow a_{58}a_{73}a_{59}$	P_{178}	$A_{78}\rightarrow A_{78}a_{33}A_{79}$	P_{243}	$A_{96}\rightarrow a_{65}a_{43}a_{65}a_{80}a_{65}$
P_{49}	$A_{18}\rightarrow A_{41}$	P_{114}	$A_{51}\rightarrow A_{51}a_{74}A_{57}$	P_{179}	$A_{78}\rightarrow A_{79}$	P_{244}	$A_{95}\rightarrow a_{43}a_{65}a_{80}a_{65}$
P_{50}	$A_{18}\rightarrow A_{42}$	P_{115}	$A_{51}\rightarrow A_{56}A_{57}$	P_{180}	$A_{79}\rightarrow A_{80}$	P_{245}	$A_{96}\rightarrow a_{65}a_{44}a_{65}$
P_{51}	$A_{18}\rightarrow A_{23}$	P_{116}	$A_{52}\rightarrow A_{52}a_{74}A_{53}$	P_{181}	$A_{79}\rightarrow a_{72}$	P_{246}	$A_{96}\rightarrow a_{44}a_{65}$
P_{52}	$A_{18}\rightarrow A_{43}$	P_{117}	$A_{52}\rightarrow A_{53}$	P_{182}	$A_{79}\rightarrow a_{50}$	P_{247}	$A_{97}\rightarrow a_{65}a_{45}a_{65}$
P_{53}	$A_{18}\rightarrow A_{44}$	P_{118}	$A_{53}\rightarrow a_{50}$	P_{183}	$A_{79}\rightarrow a_{51}$	P_{248}	$A_{97}\rightarrow a_{45}a_{65}$
P_{54}	$A_{19}\rightarrow A_{20}$	P_{119}	$A_{53}\rightarrow a_{53}$	P_{184}	$A_{79}\rightarrow a_{71}$	P_{249}	$A_{98}\rightarrow a_{65}a_{46}a_{65}$
P_{55}	$A_{19}\rightarrow A_{21}$	P_{120}	$A_{53}\rightarrow a_{61}$	P_{185}	$A_{79}\rightarrow a_{76}A_{103}a_{77}$	P_{250}	$A_{98}\rightarrow a_{46}a_{65}$
P_{56}	$A_{20}\rightarrow a_{50}a_{81}A_{82}$	P_{121}	$A_{54}\rightarrow A_{54}a_{74}a_{63}$	P_{186}	$A_{80}\rightarrow A_{82}A_{81}A_{82}$	P_{251}	$A_{99}\rightarrow a_{65}a_{47}a_{70}$
P_{57}	$A_{20}\rightarrow a_{51}a_{81}A_{82}$	P_{122}	$A_{54}\rightarrow a_{17}a_{63}$	P_{187}	$A_{81}\rightarrow a_{34}$	P_{252}	$A_{100}\rightarrow a_{65}a_{48}$
P_{58}	$A_{21}\rightarrow a_{50}a_{81}A_{74}$	P_{123}	$A_{55}\rightarrow A_{55}a_{74}a_{61}$	P_{188}	$A_{81}\rightarrow a_{35}$	P_{253}	$A_{101}\rightarrow a_{65}a_{49}$
P_{59}	$A_{21}\rightarrow a_{51}a_{81}A_{74}$	P_{124}	$A_{55}\rightarrow a_{18}a_{61}$	P_{189}	$A_{81}\rightarrow a_{36}$	P_{254}	$A_{101}\rightarrow a_{78}a_{65}a_{59}$
P_{60}	$A_{22}\rightarrow a_{3}a_{76}A_{82}a_{77}a_{68}a_{74}a_{68}a_{74}$ a_{68}	P_{125}	$A_{56}\rightarrow a_{12}$	P_{190}	$A_{81}\rightarrow a_{37}$	P_{255}	$A_{102}\rightarrow a_{76}A_{107}a_{77}$
P_{61}	$A_{23}\rightarrow A_{24}$	P_{126}	$A_{56}\rightarrow a_{13}$	P_{191}	$A_{81}\rightarrow a_{38}$	P_{256}	$A_{103}\rightarrow a_{83}A_{74}a_{84}$
P_{62}	$A_{23}\rightarrow A_{25}$	P_{127}	$A_{56}\rightarrow a_{14}$	P_{192}	$A_{81}\rightarrow a_{39}$	P_{257}	$A_{104}\rightarrow a_{83}A_{82}a_{84}$
P_{63}	$A_{24}\rightarrow a_{6}a_{76}a_{52}a_{77}$	P_{128}	$A_{56}\rightarrow a_{15}$	P_{193}	$A_{82}\rightarrow A_{82}A_{83}A_{84}$	P_{254}	$A_{105}\rightarrow a_{83}A_{75}a_{84}$
P_{64}	$A_{24}\rightarrow a_{6}a_{76}a_{52}a_{77}A_{75}$	P_{129}	$A_{56}\rightarrow a_{16}$	P_{194}	$A_{82}\rightarrow A_{84}$	P_{359}	$A_{106}\rightarrow a_{83}A_{65}a_{84}$
P_{65}	$A_{24}\rightarrow a_{6}a_{76}a_{52}a_{74}a_{53}a_{77}A_{75}$	P_{130}	$A_{57}\rightarrow a_{50}$	P_{195}	$A_{82}\rightarrow A_{83}A_{84}$	P_{260}	$A_{107}\rightarrow a_{83}A_{76}a_{84}$

rank (27) の亜順序文法である.

（注１） 亜順序文法に対する構文解析においては文中に記号⊢と⊣が存在することが前提となっている. しかし, FORTRAN 原始プログラムにはそのような記号は存在しない. そこで構文解析の前処理として以下の基準にしたがって⊢と⊣を挿入する.

1. 論理 IF 文, 算術 IF 文において, IF (〈論理式〉) 〈実行文 II〉, IF (〈算術式〉) 〈文 No.〉, 〈文 No.〉, 〈文 No.〉 なる記号列中の〈論理式〉, 〈算術式〉に出現する（の右隣に⊢,）の左隣に⊣を挿入する. 2. 文関数定義文の右辺に出現する（の右隣に⊢,）の左隣に⊣を挿入する. 3. FORMAT 文, FORMAT (〈書式仕様I〉) において, 〈書式仕様 I〉中に出現する（の右隣に⊢,）の左隣に⊣を挿入する. 4. READ 文, WRITE 文中の〈入出力並び〉の部分に関して, (と, の右隣には⊢を,) と, の左隣には⊣を挿入する. ただし整合する一群の括孤と他の一群を区切っている, には挿入しない. すなわち (((…, …), …), …), ((…, …), …) は (⊢ (⊢ (⊢…⊣, ⊢…⊣) ⊣, ⊢…⊣) ⊣, ⊢…⊣), (⊢ (⊢…⊣, ⊢…⊣) ⊣, ⊢…⊣) となる.

（注２） **4.3** で述べた前処理におけるコントロール・シンボル ($\sharp_1, \cdots, \sharp^{10}, ¢$) の挿入や注１における記号対⊢, ⊣の挿入に関して, 算術文, DATA 文, READ 文などの句識別が前提となっている. このため, 構文解析と一部, 解析が重複するが, 特別なシラブルに注目してステートメントの種類をあらかじめ決定しておかねばならない. また以上述べた前処理部の機能は構文解析部に比してかなり比重の高いものとなるが, このような前処理を施さずに解析法を形式化すれば, 解析は non-deterministic になり, 逆戻り (back-tracking) が生じて, 解析速度を著しく低める. しかも本文中で述べたように, 前処理を施した結果得られるクラスに対する解析は deterministic であり, ⊢と⊣の発見を除いては, 部分既約列の情報の使用も, 先読みも必要としない簡単な形で形式化できる.

5. むすび

本文中で述べたように FORTRAN IV 文法は亜順序文法で近似できることが明らかとなった. 亜順序文法の構文解析法の特徴は他の解析法に比べて, 構文表記号表の大きさがかなり小さくなること, 解析過程が一意であり, 手順が簡単であることなどである. 筆者らの最終的な目標は, FORTRAN 型の文法構造をもつプログラミング言語に対する, コンパイラ・コンパイラ作成への応用であり, 既存のコンパイラ・コンパイラが広範囲のプログラミング言語を対象としているために生じる, コンパイラ・コンパイラ作成上の難点を解消しようとする試みである. しかし, このためには亜順序文法の構文解析法に即した意味づけ法を定式化する必要があり, 現在研究中である.

おわりに, 有益な御助言を戴いた, 本学部, 北橋忠宏氏ならびに卒業研究として御協力を戴いた現在住友金属 K. K の遠藤忠光氏に深甚なる感謝の意を表します. さらに本論文の査読の過程で本論文のもつ種々の不備を御指摘戴き適切な御助言をたまわった査読者に深謝いたします.

7. 付録

参考文献

1) 森口繁一: JIS FORTRAN 入門上, 下巻, 東京大学出版会, 1969.
2) 浅井清: 順位関数をもつ順位文法, 情報処理, VOL. 12, NO. 5, 1971.
3) J. E. Hopcroft and J. D. Ullman: Formal Langnages and their Relation to Antomta, Addison-Wesley, Reading, Mass, 1969,
4) 井上謙蔵: コンパイラ・コンパイラ, 産業図書株式会社, 1970.

(昭和 48 年 6 月 26 日受付)
(昭和 48 年 8 月 18 日再受付)