パーザの縮小法と自動生成システム
に関する研究

1979年11月

青　江　順　一

パーザの縮小法と自動生成システムに関する研究

1979年11月

青　江　順　一

内　容　梗　概

本論文は，パーザの縮小法と自動生成システムに関する研究の成果をまとめたものであり，次の6章で構成されている。

第1章緒論においては，本研究の目的ならびにその工学上の意義およびこの分野での研究の現状について述べ，本研究の位置づけを行い，本研究によって得られた諸成果を概説している。

第2章では，本論文を通じて用いる記号および基本的な諸概念を定義するとともに，本論文における研究対象である従来の弱順位パーザ，LR(k)パーザに対する解析表と解析手順を説明する。

第3章では，従来の弱順位パーザの連続還元動作に対する性質を与え，これらの性質を利用した弱順位パーザの新しい解析表，解析手順を定義する。この新しい解析表は縮小化が容易なように二つの行列に分割されており，また新しい解析手順は従来の弱順位パーザの誤り検出位置を保存する特長をもつ。更に，連続還元動作を能率的に実行する新しい還元機械を提案し，この還元機械を行列の縮小化と解析速度の改善に利用する。行列の縮小化には，この還元機械のほかに行列の行ラベル（列ラベル）と行番号（列番号）を対応させる対応関数も有効に利用される。従って，本縮小法によって得られた弱順位パーザは，コンパクト性，高速性，有効な誤り検出能力を十分満足し，その縮小法の適用範囲もすべての弱順位文法にわたる。

第4章では，従来のLR (k)パーザの還元動作における行先状態の一意性の条件を求め，この条件を利用したLR (k)パーザの新しい解析表，解析手順を定義する。この新しい解析手順は，従来のLR (k)パーザの誤り検出位置を保存し，還元動作における冗長なスタック操作を除去している。新しい解析表で特に大きくなるのはシフト行列，参照行列と呼ばれる二つの行列であるが，本縮小法では，行列の配列の特徴を見出し，その特徴により行列の効率的な縮小を行う。従って，本縮小法によって得られたLR (k)パーザは，コンパクト性，高速性，有効なエラー検出能力を十分満足し，その縮小法の適用範囲もすべてのLR (k)文法にわたる。

第5章では，新しい弱順位パーザ，LR (k)パーザを得るための自動生成システムについて述べる。この自動生成システムは，パーザの解析表を能率的に構成す

るために，与えられた文法に対して解析表に対応するグラフを作成し，このグラフよりパーザの解析表を構成する。また，このグラフは文法の検定，修正にも利用される。この自動生成システムにより，実用プログラミング言語EULER，XPL，ALGOL 60 に対する新しい弱順位パーザ，LR(k) パーザの自動生成，解析シミュレーション結果を示す。この結果より，新しい弱順位パーザ，LR(k) パーザのコンパクト性，高速性に対する有効性を確かめる。

第6章は結論であって，本論文で得た緒結果を総括的に述べるとともに，今後の課題について述べる。

関　連　発　表　論　文

(1) 青江，山本，原田：“誤り検出遅れのない弱順位関数について”，信学会オートマトンと言語研資，AL 75－19 (1975－07)．

(2) 青江，山本，原田，島田：“弱順位関数の構成法について”信学会オートマトンと言語研資，AL 75－46 (1975－10)．

(3) 青江，山本，原田，島田：“弱順位 parser の一構成法”，電気関係学会連合大会，13－13 (1975－10)．

(4) 青江，山本，原田，島田：“誤り検出場所を保存した弱順位パーザについて”，信学会オートマトンと言語研資，AL 75－75 (1976－01)．

(5) 青江，山本：“パージングテーブルによる弱順位パーザの構成法について（その1）”，電気関係学会連合大会，13－17 (1976－10)．

(6) 青江，山本：“パージングテーブルによる弱順位パーザの構成法について（その2）”，電気関係学会連合大会，13－18 (1976－10)．

(7) 青江，山本，原田，島田：“弱順位関数の一構成法”，信学論(D)，J60-D，1（昭52-01）

(8) 山本，青江：“パージングオートマトンによる弱順位パーザ”，徳島大学工学部研究報告，22 (1977－03)．

(9) 青江，山本，島田：“GOTOグラフによるLR(1) パーザの簡単化”，信学会オートマトンと言語研資，AL 77－1 (1977－05)．

(10) 青江，山本，島田：“GO TO グラフによるLR (1) パーザの拡張”，信学会オートマトンと言語研資，AL 77－2 (1977－05)．

(11) 青江，山本，原田，島田：“パージングテーブルによる弱順位パーザの構成法”，情報処理，18, 5 (昭52－05)．

(12) 青江，山本：“GO TO グラフによる LR (1) パーザの簡単化”，信学論 (D)，J60－D，8 (昭52－08)．

(13) 青江，山本，島田：“LR (k) パーザに対する簡単化”，電気学会情報処理研究会研資，IP－77 (1977－10)．

(14) 青江，山本，島田：“拡張還元によるLR (k) パーザの簡単化（その1）”，電気関係学会連合大会，13－10 (1977－10)．

(15) 青江，山本，島田：“拡張還元によるLR（k）パーザの簡単化（その2）”，電気関係学会連合大会，13－11（1977－10）．

(16) 青江，山本，島田，原田：“弱順位パーザに対する解析シミュレーションとその評価”，信学論（D），J60－D，10（昭52－10）．

(17) 青江，山本，島田：“右順位関係の一義化について”，信学論（D），J60－D，10（昭52－10）．

(18) 青江，山本，島田：“文脈自由言語に対する構文解析部の自動生成”，徳島大学工学部研究報告，23（1978－03）．

(19) 青江，山本，秋友，島田：“LR（1）パーザの解析シミュレーション”，信学会オートマトンと言語研資，AL 78－10（1978－05）．

(20) 青江，山本，島田：“順位パーザの実際的構成法”，信学会オートマトンと言語研資，AL 78－9（1978－05）．

(21) 青江，山本，島田：“順位パーザの実際的縮小法”，信学論（D），J61－D，6（昭53－06）．

(22) 山本，青江，島田：“パージングテーブルによる弱順位パーザの補足”，情報処理，19,8（昭53－08）．

(23) 青江，山本，島田：“2組の弱順位関数の存在性”，信学論（D），J61－D，9（昭53－09）．

(24) 青江，山本，島田：“順位パーザの実際的構成法（その1）”，電気関係学会連合大会，13－4（1978－10）．

(25) 青江，山本，島田：“順位パーザの実際的構成法（その2）”，電気関係学会連合大会，13－5（1978－10）．

(26) 秋友，青江，山本，島田：“LR（1）パーザの解析シミュレーション”，電気関係学会連合大会，13－6（1978－10）．

(27) 青江，山本，島田：“LL（1）解析表の一構成法”，信学論（D），J61－D，12（昭53－12）．

(28) J. Aoe, Y. Yamamoto and R. Shimada: "A technique for generating optimal parsers for precedence grammars", Bulletin of Faculty of Engineering, Tokushima University, 15 (1978).

(29) 青江，山本，島田：“順位パーザのエラー回復”，信学会オートマトンと言

語研資，AL 79－16（1979－05）．

(30) 青江，山本，島田：“行列形式によるLR（k）解析表の縮小法”，信学会オートマトンと言語研資，AL 79－17（1979－05）．

(31) 秋友，青江，山本，島田：“エラー検出遅れのあるLR（k）パーザのエラー訂正”，信学会オートマトンと言語研資，AL 79－18（1979－05）．

(32) 青江，山本，島田：“行列形式を使用したLR（k）パーザの実際的最適化”，信学論（D），J62－D，7（昭54－07）．

(33) 青江，秋友，山本，島田：“ALGOL（JIS3000）に対するGO TOグラフによるLR（1）パーザの構成と解析シミュレーション”，信学論（D），J62－D，8（昭54－08）．

(34) 青江，山本，島田：“順位パーザの実際的縮小法に対する補そく”，信学論（D），J62－D，9（昭54－09），

(35) 青江，山本，島田：“縮小された順位パーザに対するエラー回復とシミュレーション”，信学論（D），J62－D，10（昭54－10）．

(36) 福岡，青江，山本，島田：“スパース行列の縮小化”，電気関係学会連合大会，1－6（1979－11）．

(37) 青江，秋友，山本，島田：“行列形式のデータ構造によるLR（k）パーザの縮小法（その1）”，電気関係学会連合大会，12－18（1979－11）．

(38) 青江，秋友，山本，島田：“行列形式のデータ構造によるLR（k）パーザの縮小法（その2），”電気関係学会連合大会，12－19（1979－11）．

(39) 秋友，青江，山本，島田：“縮小されたLR（k）パーザのエラー訂正Ⅰ”，電気関係学会連合大会，12－20（1979－11）．

(40) 秋友，青江，山本，島田：“縮小されたLR（k）パーザのエラー訂正Ⅱ，”電気関係学会連合大会，12－21（1979－11）．

目　　次

第1章　緒　　論

近年，電子計算機はますます多様化，大型化してきており，これ以上人力によって計算機のハードウェア，ソフトウェアを作成することは困難になりつつある。そこで，自動機械による計算機の作成が新しい課題として研究されている。ソフトウェアにおいてもコンパイラ，オペレーティング・システム等の膨大化に伴い，これらを自動的に作成する方法が考えられている。中でも，コンパイラを自動生成するコンパイラ・コンパイラ，コンパイラ・ジェネレータの研究は特に活発である。(6),(18),(20),(27),(70),(74)　しかし，自動生成されたソフトウェアは処理速度，その大きさが手作りのものより劣る傾向があるので，自動生成されたソフトウェアの最適化が重要な問題となってくる。この問題に対処するためには，手作りの場合のような試行錯誤的手法は適用できないので，理論的に統一された手法が必要となってくる。従って，言語理論の立場からは自動生成に適する新しい文法の研究が行われ，オートマトン理論の立場からは その文法に対応する処理系の最適化の研究が行われている。(6) その一つとして，コンパイラの構文解析部の理論的モデルであるパーザの研究は重要である。パーザの最適化を考える場合，次の4点を満足する必要がある。

(1)　解析速度の高速性

(2)　パーザ自身のコンパクト性

(3)　適用性（受理する言語のクラスの広さ）

(4)　有効な誤り検出能力

現在のプログラミング言語は，ほぼCFG(6)(context free grammar)で記述できるので，CFGに基づくパーザが 自動生成され，上記の(1)～(4)の条件を満足すれば申し分ないが，CFGに基づくパーザは入力言語の解析が不確定的であるので，解析速度は非常に遅くなる。この解析速度を上げるために弱順位パーザ(25),(27)，LR(k) パーザ(55)の確定的構文解析法が考え出された。これらのパーザは一意的に解析を実行できるので，その解析速度は速く，入力例の長さの線形に比例する時間で解析を終了することができる。しかし，これらのパーザは，解析を実行するための情報を解析表に蓄えて置かなければならず，この表の大きさは文法の要素数をnとした場合，弱順位パーザでn^2のオーダ，LR(k)

パーザで$n! \times n^k$のオーダとなる。実用のプログラミング言語のnは数百のオーダとなるので，これらのパーザの解析表の記憶量は膨大なものとなり，コンパクト性の条件を満足しなくなる。

以上の理由により，パーザの縮小法が数多く提案されており，弱順位パーザに対しては，Floyd[1],Wirth[2],Bell[3]，Aho[5],[6],浅井[7],[8],海尻[9],[10],[11]により，弱順位関数による方法が提案された。弱順位関数はnの線形のオーダの大きさで弱順位パーザの解析表を置換えるので，その縮小率は非常に優れたものである。しかし，この方法により得られたパーザは従来の弱順位パーザの誤り検出位置を保存しない。しかも，この関数による縮小法はすべての弱順位パーザに対して適用できるとは限らない。従って，弱順位関数による縮小法はパーザの誤り検出能力，適用性に対して大きな欠点をもつ。[12],[16]

LR(k)パーザに対しては，Korenjak[56]，Deremer[58]がLR(k)解析表の縮小法を与えたが，これらの方法はすべてのLR(k)文法に対して適用できないものである。Parger[57]の縮小法はすべてのLR(k)文法に対して適用可能なアルゴリズムであるが，使用する手順が非常に複雑であり，また得られたパーザの誤り検出能力が十分でない。関本[79],[80],大倉[81]の縮小法はかなりコンパクトな解析表を作成するが，得られたパーザの解析手順が複雑になる。これらの縮小法を使用するとき，その解析表のデータ構造は行列形式のものより表形式（リスト形式とも呼ばれる）のものを使用する場合が多い。この理由は表形式の解析表が高速性の点で行列形式のものより劣るが，縮小率の点で表形式の方が行列形式のものより優れているからである。[65]

このように従来の縮小法によるパーザでは，高速性，コンパクト性，適用性，有効な誤り検出能力のいずれかに欠点をもつ。従って，本論文ではこの四つの条件を実用的な範囲で十分満足するパーザの縮小法を提案し，縮小されたパーザを得るための自動生成システムを開発する。そして，実用のプログラミング言語EULER[2]，XPL[20]，ALGOL 60[77]に対してこのシステムを適用し，本論文で提案した新しいパーザのコンパクト性，高速性を確かめる。本縮小法は，従来の縮小法のように解析表を直接縮小するのではなく，次の順序で解析表の縮小化を行う。

（a） 従来のパーザの誤り検出位置を保存する新しい解析手順を定義し，解析表

のデータ構造を行列形式とする。

（b） 行列は，縮小化が容易になるように，かつ解析手順が能率的になるように分割される。

（c） 弱順位パーザの行列の縮小法では，連続還元動作を実行する還元機械と行列の行ラベル（列ラベル）を行番号（列番号）に対応させる対応関数に行列の一部の情報を移す。

（d） LR（k）パーザの行列の縮小法では，行列の配列の特徴を利用して，行列の要素を表現するビット数を軽減する。

（c），（d）の縮小法は，解析表の情報を欠落させないので，弱順位関数あるいはKorenjak, Deremerの縮小法のような適用性の低下なしに効率的な縮小を行う。従って，本縮小法で得られるパーザは高速性，コンパクト性，適用性，有効な誤り検出能力を十分満足するものとなる。

第2章では，本論文を通じて用いる記号および基本的な諸概念を定義するとともに，本論文における研究対象である従来の弱順位パーザ，LR（k）パーザの解析表，解析手順を説明する。

第3章では，従来の弱順位パーザの連続還元動作に対する性質を与え，この性質により従来の弱順位パーザの検出位置を保存する新しい弱順位パーザの解析手順とその解析表として縮小化が容易な二つの行列の構成法を提案する。また，連続還元動作を能率的に実行する還元機械を構成し，この還元機械を行列の冗長な行の除去，スタック操作の簡単化に利用する。更に，行列の行，列ラベルをそれぞれの行，列番号に対応させる対応関数により，行列のスパースな行列の要素を表現し，それらの行列を除去する。

第4章では，従来のLR（k）パーザの還元動作における行先状態の一意性の条件を求め，その条件を利用して従来のLR（k）パーザの誤り検出位置を保存する新しいLR（k）パーザの解析手順，解析表を提案する。この新しいパーザで特に大きくなる二つの行列に対しては，それぞれの配列の特徴を利用して行列要素を表現するビット数を軽減し，行列の縮小化を行う。また，新しいパーザの解析手順では，従来のパーザの冗長なスタック操作が除去されるので，高速性に対しても有効なものとなる。

第5章では，新しい弱順位パーザ，LR（k）パーザの自動生成システムを説明

し，実用プログラミング言語EULER, XPL, ALGOL 60に対する新しいパーザの自動生成，解析シミュレーションを行う。その結果より，新しい弱順位パーザの記憶量は 優れた縮小率を与えているAho[5]，海尻[9] の2対の弱順位関数と同程度となり，解析速度は従来の弱順位のパーザより約3倍高速になることが分かった。また，新しいLR(k)パーザの記憶量は従来の縮小法のものより約26%～33% 改善され，解析速度は表形式のLR(k)パーザより約1.8倍高速になることが分かった。

第6章では，本研究で得られた結果をまとめ，今後の課題について述べる。

第2章　パーザの解析表と解析手順

2．1　諸　　言

本章では，本論文を通じて用いる記号および基本的な諸概念を定義するとともに，本論文における研究対象である弱順位パーザ，LR(k)パーザを説明する。

2．2節では，文脈自由文法，導出，文脈自由言語についての用語を説明する。2．3節では，Aho[6]による弱順位行列を使用した従来の弱順位パーザの解析手順を説明し，2．4節では，Knuth[55]による表形式の解析表を使用した従来のLR(k)解析表の構成法，解析手順について説明する。

2．2　文脈自由言語

文脈自由文法(CFG: context free grammar)を次のように表わす。

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

(1) V_N は非終端語(nonterminal symbols)の有限集合を表わす。

(2) V_T は終端語(terminal symbols)の有限集合を表わす。

(3) Pは$A\rightarrow\alpha$なる生成規則(productions)の有限集合を表わす。

(4) SはV_Nの要素で始記号(start symbol)を表わす。

$V=V_N\cup V_T$ なる V に対して，すべての有限長の記号列からなる集合をV^*で表わし，V^+ はV^*から空記号 εを除いた集合を表わす。$ を端記号(end-marker)として使用し，$\$\notin V$ とする。以後，次の記号を特に断わらずに使用する。

$$A, B, C, \cdots \in V_N$$
$$a, b, c, \cdots \in V_T\cup\{\$\}$$
$$\cdots, X, Y, Z \in V\cup\{\$\}$$
$$\cdots, x, y, z \in (V_T\cup\{\$\})^*$$
$$\alpha, \beta, \gamma, \cdots \in (V\cup\{\$\})^*$$

記号列 $BA\gamma$ に対して，$A\rightarrow\alpha\in P$ ならば，導出(derivation) $\Rightarrow$ を

$$BA\gamma \Rightarrow B\alpha\gamma$$

で表わす。そして，$\gamma\in V_T^*$ ならば，この導出は最右導出(rightmost derivation)と呼ばれる，以後，導出$\Rightarrow$はすべてこの最右導出を表わすものとする。

$\Rightarrow$ の反射推移閉包，推移閉包をそれぞれ $\overset{*}{\Rightarrow}$，$\overset{+}{\Rightarrow}$ で表わし，一般に R^*, R^+ は関係 R の反射推移閉包，推移閉包をそれぞれ表わすものとして使用する。このとき CFG に対する文脈自由言語（context free language）は次のように表わされる。

$$L(G)=\{w \mid S \overset{*}{\Rightarrow} w,\ w \in V_T^*\}$$

$S \overset{*}{\Rightarrow} w$ に使用した生成規則の番号列を t とするとき，t を w に対する右解析列（right parse）と呼ぶ。そして，入力として w が与えられたとき，この t を出力するのがパーザ（parser）である。

2．3　弱順位パーザ

文法 G に対して，次の増大文法（augment grammar）を常に使用する。

$$G=(V_N \cup \{S_0\}, V_T \cup \{\$\}, P \cup \{S_0 \rightarrow S\}, S_0)$$

CFG が次の条件を満足するとき，正当な文法 G_p と呼ぶ。

(1) V_N のすべての A に対して，$S_0 \overset{*}{\Rightarrow} xAy \overset{*}{\Rightarrow} xzy$ が存在する。

(2) $A \rightarrow \varepsilon \notin P$。

(3) $A \overset{+}{\Rightarrow} A$ なる導出が存在しない。

V の要素に対して，次の三つの対関係 μ，λ，ρ を定義する。

$A \rightarrow \alpha XY\beta \in P$ ならば，$X \mu Y$ である。

$A \rightarrow X\alpha \in P$ ならば，$A \lambda X$ である。

$A \rightarrow \alpha X \in P$ ならば，$A \rho X$ である。

このとき，順位関係 $\doteq$，$\lessdot$，$\gtrdot$ は次のように表わされる。(17),(19),(21)～(26),(29)

$$\doteq \ = \mu$$

$$\lessdot \ = \mu\lambda^+$$

$$\gtrdot \ = \rho^+\mu\lambda^*$$

〔定義２．１〕

文法 G_p において弱順位文法（weak precedence grammar）は次の条件を満足する。(25)

(1) $\leqq \cap \gtrdot = \emptyset$，$\emptyset$ は空集合，$\leqq$ は $\doteq \cap \lessdot$ を表わす。

(2) $A \rightarrow \alpha X \beta$，$B \rightarrow \beta \in P$ ならば，$X \leqq B$ が成立しない。

以後，弱順位文法は UI（uniquely invertible ；$A \rightarrow \alpha$，$B \rightarrow \alpha \in P$ ならば，$A=B$ である）の条件も満足するものとする。

$V \cup \{\$\} \times V_T \cup \{\$\}$ 上の順位関係を蓄える行列を弱順位行列Mと呼び，行ラベルX，列ラベルaに対応する要素を $M(X, a)$ で表わす。そして，この行列Mを解析表として使用する弱順位パーザをパーザ ξ_P と呼ぶ。次に，パーザ ξ_P の解析手順を説明する。(6)

弱順位パーザ ξ_P に対する計算状況（configuration）を次のように表わす。

$$C\xi_P = (\$\alpha\beta X,\ a x\$,\ t)$$

(1) $\$\alpha\beta X$はプッシュダウンスタックの内容を表わす。

(2) $a x\$$は入力記号列を表わす。

(3) t は出力を表わす。

パーザ ξ_P の計算状況の変化を $\vdash_{\xi_P}$で表わす。

初期状況 $(\$, w\$, \varepsilon)$ が状況 C_{ξ_P} にまで変化しているものと仮定すれば，パーザ ξ_P の解析手順は図２．１のようになる。ただし，図中の第p番目の生成規則 $A_p \rightarrow \beta X$ は最長適用生成規則（longest applicable production）である。

REPEAT

　　IF $M(X,a) = \lessdot$　　**THEN**　$C_{\xi_P} \vdash_{\xi_P} (\$\alpha\beta X a, x\$, t)$

　　ELSE

　　　　IF　$M(X,a) = \gtrdot$ **AND**　$A_p \rightarrow \beta X$

　　　　THEN　$C_{\xi_P} \vdash_{\xi_P} (\$\alpha A_p, a x\$, tp)$

　　　　ELSE　*error*

UNTIL　$(\$So, \$, t')$

図２．１　弱順位パーザ ξ_P の解析手順

〔例２．１〕

次の生成規則をもつ弱順位文法G_1 を以後の例として使用する。

0．$S_0 \rightarrow S$	1．$S \rightarrow BAelse$
2．$S \rightarrow A$	3．$A \rightarrow AGC$
4．$A \rightarrow GC$	5．$A \rightarrow C$
6．$C \rightarrow CFD$	7．$C \rightarrow D$
8．$D \rightarrow D \uparrow E$	9．$D \rightarrow E$
10．$E \rightarrow (A)$	11．$E \rightarrow b$
12．$B \rightarrow if\ a\ then$	13．$F \rightarrow \div$
14．$G \rightarrow +$	

文法G_1に対する弱順位行列を図２．２に示す。

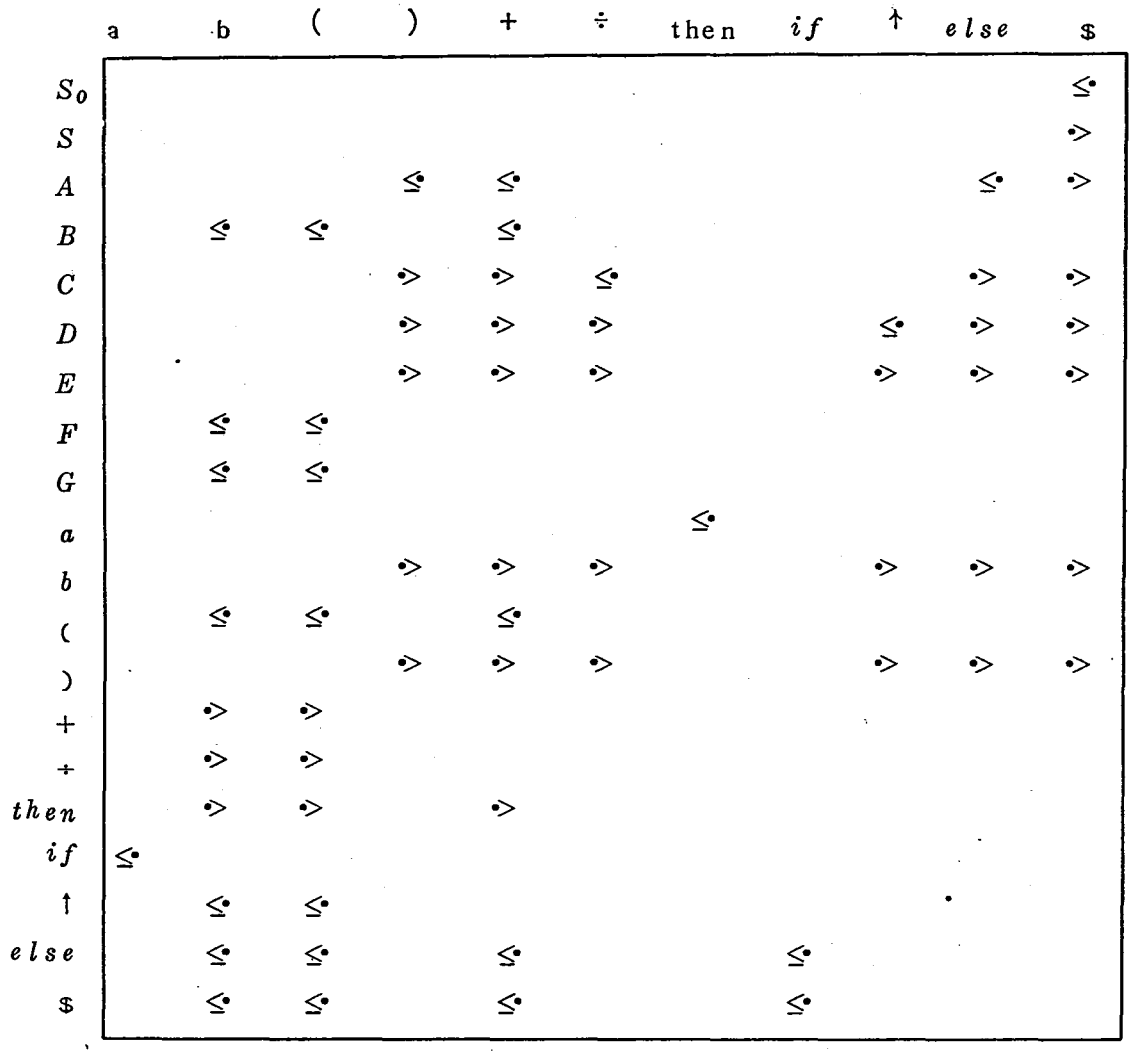


図２．２　文法G_1に対する弱順位行列

文法G_1による正しい入力文

if a then b else b

に対するパーザξ_Pの解析例を以下に示す，ただし；*if*,*then*,*else* はそれぞれ*i*,*t*,*e*と書く。また，状況の変化は単に記号⊢で表わす。

(＄，*i a t b e b*＄，ε)

⊢(＄*i*，*a t b e b*＄，ε)

⊢(＄*i a*，*t b e b*＄，ε)

⊢(＄*i a t*，*b e b*＄，ε)

⊢(＄*B*，*b e b*＄，12)

⊢(＄*B b*，*e b*＄，12)

⊢(＄*B E*，*e b*＄，12 11)

⊢(＄*B D*，*e b*＄，12 11 9)

⊢(＄*B C*，*e b*＄，12 11 9 7)

⊢(＄*B A*，*e b*＄，12 11 9 7 5)

⊢(＄*B A e*，*b*＄，12 11 9 7 5)

⊢(＄*B A e E*，＄，12 11 9 7 5 11)

⊢(＄*B A e* D，＄，12 11 9 7 5 11 9)

⊢(＄*B A e C*，＄，12 11 9 7 5 11 9 7)

⊢(＄*B A e A*，＄，12 11 9 7 5 11 9 7 5)

⊢(＄*B A e S*，＄，12 11 9 7 5 11 9 7 5 2)

⊢(＄*S*，＄，12 11 9 7 5 11 9 7 5 2 1)

⊢(＄S_0，＄，12 11 9 7 5 11 9 7 5 2 1 0)

＝受理

2．4　LR（k）パーザ

Pの要素は$S_0 \rightarrow S\$^k$を第0番目として，適当に順序づけられているものとし，第p番目の要素を

$$A_p \rightarrow X_{p_1} X_{p_2} \cdots X_{pn_p} \quad (n_p \geqq 0)$$

で表わし，p の最大値はuとする。以後，p，qは生成規則番号を表わすものとする。

文法GがLR（k）ならば，文法GについてLR（k）パーザを記述する解析表を作り出すことができる。解析表はパーザの状態（state）にそれぞれ対応する有限個の項からなる。状態sに対応する解析表の項の内容は表2．1 のようになっている。[60),(69)]

表2．1　表形式によるLR（K）解析表

状　態	先読み記号	動　作	スタック記号	行先状態
s	y_1	$shift$	Y_1	s_1
	⋮	⋮	⋮	⋮
	y_m	$reduce\ p$	Y_n	s_n

先読み記号 $y_1, \cdots, y_m$ はいずれもV_T^k（V_T^*で長さkの記号列を表わす集合である）の要素で互いに異なる記号列，またスタック記号$Y_1, \cdots, Y_n$ はいずれもVの要素で互いに異なる記号からなる。状態Sのときに入力として現れうる先読み記号の集合を$L(s)$，また $y \in L(s)$ なるy に対する動作を$A_s(y)$で表す。スタック記号の集合は$S(s)$ と表わされ，そのおのおのの要素$Y \in S(s)$に対して，行先状態$G_s(Y)$が唯一に対応している。動作の種類には$shift$, $reduce\ p$ および$accept$があって，表中では $y_1, \cdots, y_m$に対して，それぞれ唯一にどれかの動作が対応している。

解析表の作り方について述べる前に，V^* の要素に対してV_T^kの部分集合または空集合を対応させる二つの関数を定義する。

〔定義２．２〕

文法G，$\alpha \in V^+$ および $k \geqq 0$に対して，$H(\alpha)$，$H'(\alpha)$ を次のように定める。

(1) $H(\alpha)=\{ t \mid \alpha \overset{*}{\Rightarrow} t\beta,\ t \in V_T^k \}$

(2) $H'(\alpha)=\{ t \mid \alpha \overset{*}{\Rightarrow} t\beta,\ t \in V_T^k$，ただし，$\alpha \overset{*}{\Rightarrow} t\beta$には$A \rightarrow \varepsilon$による導出を含まない$\}$

Knuth[55]による解析表での状態 sとは〔p, j, w〕なる部分状態(partial state)の集まりであって，その状態の集合をKで表わすとき，LR(k)解析表の構成手順は図２．３のようになる。[70]～[76]

```
PROCEDURE CLOSURE(s);
BEGIN
      REPEAT
        FOR  〔p,j,w〕∈s, A_q → X_q1 X_q2 ··· X_qn_q ∈P
             and x∈H(X_p(j+2) ··· X_pn_p w)
        DO   add 〔q, 0, x〕 to s;
      UNTIL  no more partial states can be added to s;
      RETURN s
END
PROCEDURE GOTO(s,Y)
BEGIN
     let s' be the set of partial state 〔p,j+1,w〕
     such that 〔p,j,w〕∈s and X_pj=Y;
     RETURN  CLOSURE(s')
END;
BEGIN
     K:={CLOSURE({S_0 → ·S,$})};
     REPEAT
              FOR  s∈K and Y∈V
              DO   add GOTO(s,Y) to K
     UNTIL  no more sets of partial states can be added to K;
     FOR  〔p,j,w〕∈s, j<n_p and y∈H'(X_p(j+1) ··· X_pn_p w)
     DO   add y to L(s) and A_s(y)=shift;
     FOR  〔p,n_p,w〕∈s  DO  add w to L(s) and A_s(y)=reduce p;
     IF   GOTO(s,Y)=s' THEN add Y to S(s) and G_s(Y)=s'
END
```

図２．３　LR(k)解析表の構成法

ただし，$S_I=\{[0,0,\$^k]\}$，$S_F=\{[0,2,\$^k]\}$なる状態をそれぞれ初期状態，最終状態とし，$A_{s_F}(\$^k)=accept$とする。

表２．１の解析表に従うLR(k)パーザをパーザξ_Lと呼ぶ，パーザξ_Lの計算状況を

$$C_{\xi_L}=(s_I X_1 s_1 \cdots X_n s_n, a_1 \cdots a_k x\$, t)$$

で表わす。( )内の第１～第４要素はそれぞれスタックの内容，入力記号列，出力を表わす。入力wに対して，$C_{\xi_L}=(s_I, w\$, \varepsilon)$より解析を始め，状況$C_{\xi_L}$まで計算が進んでいるとき，パーザ$\xi_L$の解析手順は図２．４のようになる。

REPEAT

　IF　$a_1 a_2 \cdots a_k \in L(s_n)$ **THEN**

　　IF　$A_{s_n}(a_1 a_2 \cdots a_k)=shift$

　　THEN　$C_{\xi_L} \vdash_{\xi_L} (s_1 X_1 s_1 \cdots X_n s_n a_1 G_{s_n}(a_1), a_2 \cdots a_k x\$, t)$

　　ELSE

　　　COMMENT　$A_{s_n}(a_1 a_2 \cdots a_k)=reduce\ p$

　　　$C_{\xi_L} \vdash_{\xi_L} (s_1 X_1 s_1 \cdots X_{n-n_p} s_{n-n_p} A_p G_{s_{n-n_p}}(A_p), a_1 a_2 \cdots a_k x\$, tp)$

　ELSE　error

UNTIL　$(s_I S_0, \$, t')$

図２．４　LR(k) パーザξ_Lの解析手順

〔例２．２〕

次の生成規則をもつLR(1) 文法G_2を以後の例として使用する。

０．　$S_0 \rightarrow S\$$　　１．　$S \rightarrow S+T$

２．　$S \rightarrow T$　　３．　$T \rightarrow T*F$

４．　$T \rightarrow F$　　５．　$F \rightarrow (S)$

６．　$F \rightarrow a$

文法G_2に対するLR(1)解析表を表2．2に示し，入力文$(a+a)*a$に対する解析例を示す。

$\vdash (s_I\ ,(a+a)*a\$\ ,\varepsilon)$

$\vdash (s_I\ (s_5\ ,a+a)*a\$\ ,\varepsilon)$

$\vdash (s_I\ (s_5\ a\ s_4\ ,+a)*a\$\ ,\varepsilon)$

$\vdash (s_I\ (s_5\ F\ s_8\ ,+a)*a\$\ ,6)$

$\vdash (s_I\ (s_5\ T\ s_2\ ,+a)*a\$\ ,64)$

$\vdash (s_I\ (s_5\ S\ s_8\ ,+a)*a\$\ ,642)$

$\vdash (s_I\ (s_5\ S\ s_8\ +s_6\ ,a)*a\$\ ,642)$

$\vdash (s_I\ (s_5\ S\ s_8\ +s_6\ a\ s_4\ ,)*a\$\ ,642)$

$\vdash (s_I\ (s_5\ S\ s_8\ +s_6\ F\ s_3\ ,)*a\$\ ,6426)$

$\vdash (s_I\ (s_5\ S\ s_8\ +s_6\ T\ s_9\ ,)*a\$\ ,64264)$

$\vdash (s_I\ (s_5\ S\ s_8\ ,)*a\$\ ,642641)$

$\vdash (s_I\ (s_5\ S\ s_8\ )s_{11}\ ,*,a\$\ ,642641)$

$\vdash (s_I\ F\ s_3\ ,*a\$\ ,6426415)$

$\vdash (s_I\ T\ s_2\ ,*a\$\ ,64264154)$

$\vdash (s_I\ T\ s_2\ ,*s_7\ ,a\$\ ,64264154)$

$\vdash (s_I\ T\ s_2*s_7\ a\ s_4\ ,\$\ ,64264154)$

$\vdash (s_I\ T\ s_2*s_7\ F\ s_{10}\ ,\$\ ,6426415463)$

$\vdash (s_I\ T\ s_2\ ,\$\ ,6426415463)$

$\vdash (s_I\ S\ s_1\ ,\$\ ,64264154632)$

＝受理

表２．２　文法 G_2 に対するＬＲ(k)解析表

状態	先読み記号	動作	スタック記号	行先状態
s_I	a (	*shift* *shift*	E T F a (	s_1 s_2 s_3 s_4 s_5
s_1	+ $	*shift* *accept*	+	s_6
s_2	* +) $	*shift* *reduce* 2 *reduce* 2 *reduce* 2	*	s_7
s_3	* +) $	*reduce* 4 *reduce* 4 *reduce* 4 *reduce* 4		
s_4	* +) $	*reduce* 6 *reduce* 6 *reduce* 6 *reduce* 6		
s_5	a (	*shift* *shift*	E T F a (	s_8 s_2 s_3 s_4 s_5
s_6	a (	*shift* *shift*	T F a (	s_9 s_3 s_4 s_5
s_7	a (	*shift* *shift*	F a (	s_{10} s_4 s_5
s_8	+)	*shift* *shift*	+)	s_6 s_{11}
s_9	* +) $	*shift* *reduce* 1 *reduce* 1 *reduce* 1	*	s_7
s_{10}	* +) $	*reduce* 3 *reduce* 3 *reduce* 3 *reduce* 3		
s_{11}	* +) $	*reduce* 5 *reduce* 5 *reduce* 5 *reduce* 5		

2.5 結　言

本章では，本論文を通じて用いる記号および基本的な諸概念を定義するとともに，本論文における研究対象である弱順位パーザ，LR(k)パーザの解析表，解析手順を説明した。

第３章　弱順位パーザの縮小法

３．１　緒　　言

Floyd[1]，Wirth[2]，Bell[3]，Martin[4]，Aho[5],[6]，浅井[7],[8]，海尻[9],[10],[11]によって提案された弱順位関数による弱順位行列の縮小法はコンパクト性が非常に優れている反面，誤り検出能力と適用上の広さが劣る。従って，本章では，有効な誤り検出能力をもちかつすべての弱順位文法に対して適用可能な弱順位行列の効率的な縮小法を提案する。

３．２節では，従来の弱順位パーザ ξp の性質，特に連続還元動作に対する性質を調べ，連続還元動作の開始と終了の条件を明らかにする。３．３節では，これらの性質を利用することにより従来のパーザ ξp の誤り検出位置を保存する新しい弱順位パーザの解析表，解析手順を定義する。この新しい弱順位パーザの行列は縮小化が容易ように二つに分割されている。また，連続還元動作を能率的に実行する還元機械ＲＭを提案する。このＲＭを使用する解析手順は還元動作における冗長なスタック操作を必要としないので，解析速度の改善にも役立つ。３．４節では，この新しいパーザに対する次の縮小法を提案する。

(1)　ＲＭを使用した行列の冗長な行の除去

(2)　対応関数による順位関係の表現

(1)は，一部の順位関係をＲＭで判定できるように解析手順を変更することで実現される。対応関数とは行列の行ラベル（列ラベル）を行番号（列番号）に対応させる関数のことであるが，(2)では，順位関係の一部をこの対応関数で表現することにより，行列の行と列を除去する。

３．２　弱順位パーザの性質

まず，文法 G に対して次の集合を定義する。

〔定義３．１〕

(1)　$\mathrm{LEFT}(B)=\{a \mid B \leqq a\}$

(2)　$\mathrm{RIGHT}(B)=\{b \mid B \geqq b\}$

(3)　$V_1=\{B \mid \mathrm{LEFT}(B)=\phi\}$

(4) $V_2 = \{C \mid A \in V_N$ なるすべての A に対して，$A \rho C$ が成立しない $\}$

(5) $V_3 = \{X_k \mid X_k \in V_2$ なる X_k に対して，$\mathrm{LEFT}(X_k) = \mathrm{RIGHT}(X_1) - \bigcup_{i=2}^{k-1} \mathrm{LEFT}(X_i), X_i \rho X_{i-1}, 2 \leqq i \leqq k, X_1 \in V_T$ が成立する $\}$

〔定理 3 . 1 〕

パーザ ξ_P において，スタック記号 a ，入力記号 b で連続還元動作が始まれば，

$B \rho^+ a$, $B \leqq b$

なる記号 B がスタックのトップに現れるまで，連続還元動作が続き，その後必ずシフト動作が起る。

（証明）

定義 2 . 1 より $B\rho^+C, C\rho^+a$, $C \leqq b$ なる C は存在しない。故に，定理の成立は明らかである。

（証明終）

〔定理 3 . 2 〕

パーザ ξ_P の計算状況が $B \in V_1$ なるスタック記号 B をもつならば，次の動作は必ず還元動作となる。

（証明）

$B \in V_1$ なる B は $\mathrm{LEFT}(B) = \varnothing$ であるから，$B \leqq a$ なる終端語 a は存在しない。故に，定理の成立は明らかである。

（証明終）

〔定理 3 . 3 〕

パーザ ξ_P の計算状況が $C \in V_2$ なるスタック記号 C をもつならば，次の動作は必ずシフト動作となる。

（証明）

$C \in V_2$ なる C は $A \rho C$ なる A をもたないので，次の動作が還元動作となることはない。故に，定理の成立は明らかである。

（証明終）

以上の定理は，パーザ ξ_P が誤りを検出する場合を除くが，次の定理はその場合でも成立する。

〔定理３．４〕

パーザ ξ_P の計算状況が $D \in V_i$ なるスタック記号 D をもつならば，次の動作は必ずシフト動作となる。

（証明）

記号 D を定義３．１の(5)の X_k とおき，入力記号を a とすると，$X_k \leqq a$ が成立することは $X_k \in V_2$（$V_2 \supseteq V_3$）より明らかである。また，

$$a \notin \bigcup_{i=2}^{k-1} \mathrm{LEFT}(X_i),\quad a \in \mathrm{RIGHT}(X_1)$$

であるから，この状況でパーザ ξ_P が誤りを検出することはない。故に，定理の成立は明らかである。

（証明終）

$X_k \in V_2$ なる X_k に対して，一般に成立する式は次のようになる。

$$\mathrm{LEFT}(X_k) \subseteq \mathrm{RIGHT}(X_1) - \bigcup_{i=2}^{k-1} \mathrm{LEFT}(X_i)$$
$$X_i \rho X_{i-1},\quad 2 \leqq i \leqq k$$

例として，図３．１のグラフを考えよう。このグラフは $X \rho Y$ ならばノード X からノード Y へアークが引かれ，$\mathrm{LEFT}(X) \neq \phi$ ならば，ノード X の右隣に $\mathrm{LEFT}(X)$ の内容が示されている。この例では，

$$\mathrm{LEFT}(X_k) = \{a\}$$
$$\mathrm{RIGHT}(X_1) = \{a, b\}$$
$$\bigcup_{i=2}^{k-1} \mathrm{LEFT}(X_i) = \phi$$

であるから

$$\mathrm{LEFT}(X_k) \subseteq \mathrm{RIGHT}(X_1) - \bigcup_{i=2}^{k-1} \mathrm{LEFT}(X_i)$$

が成立する。従って，パーザ ξ_P の計算状況が X_k なるスタック記号をもつ場合，次の入力が a である保証はないので，パーザ ξ_P は誤りを検出する可能性がある。

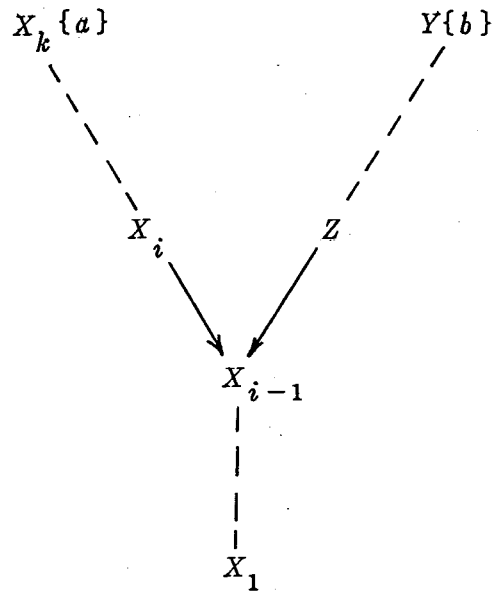


図　3。1　　説明のためのグラフ

3．3　新しい弱順位パーザ

本節で提案する新しい弱順位パーザをπ_P ではシフト動作を決定する行列として，シフト行列M_S を使用し，連続還元動作の開始と終了を決定する行列として還元行列M_r を使用する。また，連続還元動作を能率的に実行する機械として，還元機械RM を使用する。

3．3．1　新しい弱順位パーザの解析表

弱順位文法Gに対して，行列M_S ,M_r，還元機械RMを次のように定義する。

〔定義3．2〕

$V_T \cup \{\$\} \times V_T$ 上のシフト行列M_S と$V \cup \{S_0\} \times V_T \cup \{\$\}$上の還元行列$M_r$ の要素{1，0}を次のように決定する。

(1)　$b \leq a$ ならば，$M_S(b, a)=1$ とする。

(2)　$b > a$ ならば，$M_r(b, a)=1$ とする。

(3)　$B \leq a$ ならば，$M_r(B, a)=1$ とする。

(4)　1以外の要素をすべて0とする。

〔定義３．３〕

還元機械RMを次のように表わす。

$$RM=(Q, \Gamma, V_P, V_L, \delta)$$

(1) QはVの要素に対応する状態であって，記号Xに対してR_Xで表わす。

(2) Γはスタック記号の集合であって，$\Gamma=V\cup\{\$\}$である。

(3) V_Pは生成規則番号の集合を表わす。

(4) V_Lは生成規則の左辺の記号の集合であって，$V_L=V_N$である。

(5) δは$Q\times\Gamma^*$から$V_P\times V_L\times Q$への写像を表わす関数であって，

$A_p\rightarrow\alpha_p X_p$ なるp番目の生成規則に対して，$A_p\in V_2$ならば，

$$\delta(R_{X_p}, \alpha_p)=[p, \varepsilon, R_{A_p}]$$

とし，それ以外のA_pに対して，

$$\delta(R_{X_p}, \alpha_p)=[p, A_p, R_{A_p}]$$

とし，α_pをRHS(right-hand side)，εあるいはA_pをLHS(left-hand side)と呼ぶ。

３．３．２　新しい弱順位パーザの解析手順

まず，SHIFT，REDUCEの動作を説明しておく。

SHIFTは，従来のシフト動作と同様であって，入力記号をスタックにプッシュダウンすることである。

REDUCEは，RMにおける連続した還元動作に関係する次のような１回の動作である。RMを使用しているときのパーザπ_Pの計算状況は，パーザξ_Pの状況の第４項にRMの状態を加えて表わされる。いま，

$$C_{\pi_P}=(\$\alpha, ax\$, t, R_X)$$

なる状況に対して，

$$\delta(R_X, \beta_1), \delta(R_X, \beta_2), \cdots, \delta(R_X, \beta n); n\geqq 1$$

が定義されており，αの最長の接頭語（suffix）βj，$1\leqq j\leqq n$が存在すると仮定する。このとき

$$\delta(R_X, \beta j)=[p, \varepsilon, R_{A_p}], \alpha=r\beta j$$

ならば，パーザπ_Pは

$$C_{\pi_P}\vdash_{\pi_P} (\$r, ax\$, tp, R_{A_p})$$

と変化し，

$$\delta(R_X, \beta j) = [p, Ap, R_{A_p}]$$

ならば，パーザπ_Pは

$$C\pi_P \vdash_{\pi_P} (\$r, a x \$, tp, R_{A_p})$$

と変化した後，$M_r(A_p, a)=0$ならば，このポップアップ動作を継続し，$M(A_p, a)=1$ならば，RMでの連続ポップアップ動作を終了する。但し，このような接頭語βjが存在しなければ，パーザπ_Pの計算状況C_{π_P}は誤りとなる。このほか，$A_p \in V_t$かつ$M_r(A_p, a)=0$の場合もポップアップ動作が不可能となるので，誤りとなる。

パーザπ_Pでは以上のようにして終了した連続ポップアップ動作を正式な連続還元動作の終了とするために，生成規則のLHSをスタックにプッシュダウンする。

左端の入力記号をIN，スタックのトップ記号をTOS，RMの入口状態をR_{ENT}，現在使用中の状態に対するLHSをLHS_{CR}と表わすとき，パーザπ_Pの解析手順は図3．2のようになる。

```
REPEAT
        REPEAT  SHIFT  UNTIL  M_s(TOS,IN)=0
        IF  M_r(TOS,IN)=0  THEN  error
        ELSE
                BEGIN
                        R_ENT := R_TOS;
                        pop up TOS;
                        REPEAT  REDUCE  UNTIL M_r(LHS_CR,IN)=1;
                        push down LHS_CR
                END
UNTIL  ($S_0, $, t)
```

図3　2　　パーザπ_Pの解析手順

〔定理３．５〕

パーザπ_Pは，正当な弱順位パーザであり，パーザπ_Pの誤り検出位置はパーザξ_Pと等価である。

（証明）

パーザπ_Pの$V_T \cup \{\$\} \times V_T \cup \{\$\}$上のシフト，還元の決定および誤りの検出は定義３．２と図３．２の解析手順によりパーザξ_Pと等価であることが分かる。次に，パーザξ_Pにおける連続還元動作と１回のシフト動作を考える。

$$
\begin{aligned}
&(\$\alpha_1 X_1, a x \$, t_1) \\
\vdash_{\xi_P}\ &(\$\alpha_2 X_2, a x \$, t_2) \\
&\vdots \\
\vdash_{\xi_P}\ &(\$\alpha_k X_k, a x \$, t_k) \\
\vdash_{\xi_P}\ &(\$\alpha_k X_k a, x \$, t_k)
\end{aligned}
$$

ここで，$X_1 \in V_T$ ；$X_2, \cdots, X_k \in V_N$ ；$k \geqq 2$とする。このとき，$X_k \leqq a$かつ$1 \leqq i \leqq k-1$なるすべてのiに対して$X \gtrdot a$であるから，行列M_rでは$M_r(X_i, a) = M_r(X_k, a) = 1$かつ$2 \leqq j \leqq k-1$なるすべての$j$に対して，$M_r(X_j, a) = 0$が成立する。また，$t_k = t_{k-1}$ pとすれば，$\delta(R_{X_{k-1}}, a_p) = [p, A_p, R_{X_k}]$かつ$A_p \neq \varepsilon$であるから（$t_k = t_{k-1}$ pとしたので，$X_{k-1} = X_p$，$X_k = A_p$となる），パーザπ_Pでは次の計算状況の変化に対応できる。

$$
\begin{aligned}
&(\$\alpha_1 x_1, a x \$, t_1, R_{X_1}) \\
\vdash_{\pi_P}\ &(\$\alpha_1, a x \$, t_1, R_{X_1}) \\
\vdash_{\pi_P}\ &(\$\alpha_2, a x \$, t_2, R_{X_2}) \\
&\vdots \\
\vdash_{\pi_P}\ &(\$\alpha_k, a x \$, t_k, R_{X_k})
\end{aligned}
$$

以上より，正しい入力文に対してパーザξ_Pとパーザπ_Pは等価であるといえる

次に，パーザπ_Pの誤り検出がパーザξ_Pより遅れる場合を考える。この状況に対応するグラフとして図３．１のものを使用する。パーザξ_Pは次のように誤りを検出する。

$$
\begin{aligned}
&(\$\beta_1 X_1, b x \$, t_1') \\
\vdash_{\xi_P}\ &(\$\beta_2 X_2, b x \$, t_2') \\
&\vdots
\end{aligned}
$$

$\vdash_{\varepsilon_R}$ （$\$\beta_i x_i$, $b x \$$, t_i ）

＝誤り

すなわち，$1 \leqq m \leqq i-1$, $2 \leqq i \leqq k$ なるすべてのmに対して，$X_m \geqslant b$ であるが $X_i > b$ かつ$X_i \leqq b$が成立しない。これに対して，パーザπ_Pでは$M_r(X_r, b) \neq 1$よりシフト動作が起ることはないので，誤り検出位置は保存される。この点を除けば，誤り検出についてもパーザε_Pとパーザπ_Pは等価である。故に，定理の成立は明らかである。

（証明終）

（例３．１）

文法G_1に対するV_1 , V_2 , V_3 ,は次のようになる

$$V_1 = \{S, E\}$$

$$V_2 = \{S_0, B, F, G\}$$

$$V_3 = \{S_0, B, F, G\}$$

このように，文法G_1に対しては$V_2 = V_3$となる。次に，文法G_1に対するRMの内容を示す。

$\delta(R_S, \varepsilon) = [0, So, R_{So}]$

$\delta(R_S, BA\ else) = [1, \varepsilon, R_S]$

$\delta(R_A, \varepsilon) = [2, \varepsilon, R_S]$

$\delta(R_C, AG) = [3, A, R_A]$

$\delta(R_C, A) = [4, A, R_A]$

$\delta(R_C, \varepsilon) = [5, A, R_A]$

$\delta(R_D, CF) = [6, C, R_C]$

$\delta(R_D, \varepsilon) = [7, C, R_C]$

$\delta(R_E, D\uparrow) = [8, D, R_D]$

$\delta(R_E, \varepsilon) = [9, D, R_D]$

$\delta(R), (A) = [10, \varepsilon, R_E]$

$\delta(R_b, \varepsilon) = [11, \varepsilon, R_E]$

$\delta(R_{then\ if\ a}) = [12, B, R_B]$

$\delta(R_{\div}, \varepsilon) = [13, F, R_F]$

$\delta(R_{+}, \varepsilon) = [14, G, R_G]$

入力文 $if\ a\ then\ b\ else\ b$ に対するパーザ π_P の解析例を以下に示す。

$\vdash$(\$, $i\ a\ t\ b\ e\ b$ \$, ε)

$\vdash$(\$ i, $a\ t\ b\ e\ b$ \$, ε)

$\vdash$(\$ $i\ a$, $t\ b\ e\ b$ \$, ε)

$\vdash$(\$ $i\ a\ t$, $b\ e\ b$ \$, ε)

$\vdash$(\$ $i\ a$, $b\ e\ b$ \$, ε, R_{then})

$\vdash$(\$, $b\ e\ b$ \$, 12, R_B)

$\vdash$(\$ $B\ b$, $e\ b$ \$, 12)

$\vdash$(\$ B, $e\ b$ \$, 12, R_b)

$\vdash$(\$ B, $e\ b$ \$, 12 11, R_E)

$\vdash$(\$ B, $e\ b$ \$, 12 11 9, R_D)

$\vdash$(\$ B, $e\ b$ \$, 12 11 9 7, R_C)

$\vdash$(\$ B, $e\ b$ \$, 12 11 9 7 5, R_A)

$\vdash$(\$ $B\ A\ e$, b \$, 12 11 9 7 5)

$\vdash$(\$ $B\ A\ e\ b$, \$, 12 11 9 7 5)

$\vdash$(\$ $B\ A\ e$, \$, 12 11 9 7 5, R_b)

$\vdash$(\$ $B\ A\ e$, \$, 12 11 9 7 5, R_E)

$\vdash$(\$ $B\ A\ e$, \$, 12 11 9 7 5 11 9, R_D)

$\vdash$(\$ $B\ A\ e$, \$, 12 11 9 7 5 11 9 7, R_C)

$\vdash$(\$ $B\ A\ e$, \$, 12 11 9 7 5 11 9 7 5, R_A)

$\vdash$(\$ $B\ A\ e$, \$, 12 11 9 7 5 11 9 7 5 2, R_S)

$\vdash$(\$, \$, 12 11 9 7 5 11 9 7 5 2, R_S)

$\vdash$(\$, \$, 12 11 9 7 5 11 9 7 5 2 1, R_{So})

$\vdash$(\$ So, \$, 12 11 9 7 5 11 9 7 5 2 1 0)

= 受　理

3．4　解析表の縮小法

まず，3．2節の性質を利用した行列M_rの縮小法を示す。

定理3．2とRMの操作法により，行列M_rの$B \in V_1$に対応するラベルBの行は参照されないので，除去してもよい。また，定理3．4により，$A_p \in V_3$に対応するLHS A_pはRMでもし検出できるならば，行列M_rのラベルA_pの行は除去可能となる。LHS A_pは以後数値化された整数値として考える。

＜手順3．1＞　行列M_rの冗長な行の除去

（3．1－1）　$B \in V_1 \cup V_3$なるラベルBをもつ行を行列列M_rより除去する。

（3．1－2）　RMの$\delta(R_{X_p}, \alpha_p) = [p, A_p, R_{A_p}]$に対して，

$A_p \in V_1$ならば，LHS＝0

$A_p \in V_3$ならば，LHS＝$-A_p$

それ以外はLHS＝A_p

とする。

（手順終）

（例3．2）

手順3．1の縮小法により，行列M_rのラベルS_0，S，E，B，F，Gの行が除去され，行列M_s，M_rは図3．3のようになる。ただし，同一行，同一列はまとめ，0要素のみをもつ行，列は省く。

	(				
	b	+	*if*	*then*	*a*
else, $	1	1	1	0	0
(	1	1	0	0	0
↑	1	0	0	0	0
a	0	0	0	1	0
if	0	0	0	0	1

(a)　M_s

	)				(
	$	+	*else*	÷	↑	*b*
b,)	1	1	1	1	1	0
D	0	0	0	0	1	0
C	0	0	0	1	0	0
A	0	1	1	0	0	0
then	0	1	0	0	0	1
+, ÷	0	0	0	0	0	1

(b)　M_r

図3．3　　文法G_1に対する行列M_s, M_r

弱順位行列は行列M_s, M_rに分割，縮小してもなお非常にスパースなものであり，特に1の要素を1個だけしかもたない行，列（UN行，UN列と呼ぶ）が数多く存在する。従って，このUN行，UN列が行列M_s ,M_r上から消去できれば行列の縮小率は更に高くなる。行列M_s ,M_rの同一行，同一列は可能な限り併合されるので，行ラベル（列ラベル）を行番号（列番号）に対応させる対応関数$f(g)$が必要となる。この対応関数を利用することによって，行列M_s ,M_r上のUN行，UN列は消去可能となる。次に，対応関数の構成法を示す。

＜手順３．２＞　対応関数の構成法

（３．２－１）　UN行（UN列）でない行（列）を第1行（第1列）より並べ，引き続いてUN行（UN列）を並べる。

（３．２－２）　$M(a_i, b_j)=M(a_m, b_n)=1$なるUN行$a_i$,UN列b_nに対して，$j=m$が成立するならば，$j\neq m$となるように行あるいは列を並び換える。ただし，それが不可能ならば，a_i行あるいはb_n列をUN行，UN列でないものとして取扱う。

（３．２－３）　$M(a_i, b_j)=1$なるUN行a_iに対して，

$$f=(a_i)=-j$$

とし，$M(a_m, b_n)=1$なるUN列b_nに対して，

$$g=(b_n)=-m$$

とする。

（３．２－４）　0の要素のみをもつ行a_i ,列b_jに対しては行列の大きさを$n\times m$ $(n\leqq m)$とするとき，

$$f(a_i)=-(m+1)$$

$$g(b_j)=-(m+2)$$

とする。

（３．２－５）　UN行（UN列）でない行ラベル（列ラベル）についてはその行番号，列番号を対応させる。

（手順終）

手順（３．２）の後，行列M_s ,M_rのUN行，UN列をすべて消去する。従って，パーザπ_Pの解析手順では$f_s(a)<0$あるいは$g_s(b)<0$であって，

$$|f_s(a)|=|g_s(b)|$$

ならば，$M_{si}(a, b)=1$と判断し，

$$|f_s(a)| \neq |g_s(b)|$$

ならば，$M_s(a, b)=0$と判断する操作を追加する。

手順（3．2－2）で行あるいは列の並び換えを行ったのは

$$M_s(a_i, b_j)=M(a_m, b_n)=1, j=m$$

を満足するUN行a_i，UN列b_nを手順（3．2－3）で関数化すれば，

$$M_s(a_i, b_n)=0$$

であるにもかかわらず，

$$f_s(a_i)=g_s(b_n)=-j(=-n)$$

となるからである。例3．1の行列M_sの

$$M_s(\uparrow, b)=M_s(\$, if)=1$$

なるUN行↑，UN列ifがこの場合に相当する。

以上より，手順3．2の正当性は明らかである。

（例3．3）

例3．2の行列M_s，M_rは手順3．2により，それぞれ2×2，3×5の大きさに縮小される。縮小された行列の対応関数を図3．4に示す。

	A	*C*	*D*	*a*	*b*	(	)	+	÷	*then*	*if*	↑	*else*	$
f_s				-4	-6	2	-6	-6	-6	-6	-5	-1	1	1
g_s				-5	1	1	-7	2	-7	-3	-1	-7	-7	-7
f_r	4	-4	-5	-7	1	-7	1	-6	-6	5	-7	-7	-7	-7
g_r				-8	6	6	3	2	4	-8	-8	5	3	-1

図3．4　文法G_1に対する行列M_s，M_rの対応関数

$f(a)<0$あるいは$g(b)<0$であって，$|f(a)|=|g(b)|$ならば，$a \# b$と書くとき，パーザπ_pの最終的な解析手順は図3．5のようになる。

```
    REPEAT
out :       REPEAT  SHIFT  UNTIL  TOS≠IN or M_s(f_s(TOS), g_s(IN))=0;
            IF  TOS≠IN or M_r(f_r(TOS), g_r(IN))=0 THEN  error
            ELSE
                BEGIN
                    R_ENT := R_TOS;
                    pop up TOS ;
loop:               REDUCE;
                    IF LHS_CR=0   THEN GOTO  loop
                    ELSE IF  LHS_CR < 0  THEN
                      BEGIN
                          push down  -LHS_CR;
                          GOTO out
                      END
                        ELSE  IF  M_r(f_r(LHS_CR), g_r(IN))=0
                              THEN  GOTO  loop
                              ELSE  push  down  LHS_CR
                END
    UNTIL   ($S_0, $, t)
```

図３．５　パーザ π_P の修正された解析手順

3.5 結　言

本章では，従来の弱順位パーザξ_Pの連続還元動作に対する性質を明らかにし，それらの性質を利用した新しい弱順位パーザπ_Pの解析表，解析手順を定義した。そして，このパーザπ_Pに対して効率的な縮小法を提案した。パーザπ_Pは，従来のパーザξ_Pの誤り検出位置を保存するので，従来の縮小法のような誤り検出能力の低下は起らない。

本縮小法による有効性は，次の四つの特長で示される。

(1) 連続還元動作を能率的に実行する還元機械ＲＭを導入し，行列の冗長な行の除去に利用した。

(2) $B \leqq a$なる順位関係を行列M_sではなく，行列M_rで表現し，行列M_s，M_rの要素の総数を軽減した。

(3) 行ラベル（列ラベル）と行番号（列番号）を対応させる対応関数で一部の順位関係を表現した。

(4) $A_p \rightarrow \alpha_p X_p$に対応するＲＭのＲＨＳを$\alpha_p$とし，パーザ$\pi_P$の連続還元動作におけるスタック操作を簡単にした。

以上より，本章の縮小法で得られたパーザπ_Pは，コンパクト性，適用性はもちろん誤り検出能力，高速性に対しても十分有効なものであると思われる。

第4章　LR(k)パーザの縮小法

4.1　緒　　言

Krenjak[56]によって提案されたLR(k)パーザの縮小法は，その適用性が十分でなく，Parger[57]による縮小法で得られたパーザは有効な誤り検出能力をもたない。また，関本[79]，[86]，大倉[81]によるパーザは解析手順が複雑であるため，解析速度が低下する欠点を持っている。これに対して，Deremer[58]はLR(k)文法のサブクラスSLR(k)文法を定義し，SLR(k)文法に対するパーザのコンパクト性を示した。実用プログラミング言語に対するSLR(k)パーザの具体例はLalonde[59]，Anderson[62]が表形式の解析表として与え，Joliat[64]が行列形式の解析表として与えた。しかし，表形式の解析表を使用するパーザは，行列形式の解析表を使用するパーザよりコンパクト性の点で優れている反面，高速性の点で劣る。このように従来のLR(k)パーザの縮小法では，適用性，有効な誤り検出能力，高速性をすべて満足するものはない。

本章では，有効な誤り検出能力と適用性を満足する縮小法を与えるために，従来のパーザξ_Lの誤り検出位置を保存しかつすべてのLR(k)，LALR(k)[59]，SLR(k)文法に対して適用可能な方法を提案する。また，この新しい縮小法によるパーザは，行列形式の解析表を使用し，その解析手順では還元動作における冗長なスタック操作が除去されるので，高速性に対しても有効なものとなる。4.2節では，従来のLR(k)パーザξ_Lの性質，特に還元動作における行先状態の一意性について調べ，その条件を明らかにする。4.3節では，この性質を利用した新しいLR(k)パーザの解析表，解析手順を定義する。新しいパーザの解析表で特に大きくなるのはシフト行列N_s，参照行列N_rと呼ばれる二つの行列であるが，4.4節では，行列の配列の特徴を利用したこれら二つの行列の縮小法を提案する。

4.2　LR(k)パーザの性質

実際の場合，$L(s)$の要素の記号長はすべてkに統一しなくてもよいので，以後$L(s)$の各要素は状態sからの各動作を一意的に決定するのに必要な最小長(1以上)の記号列とする。また，シフト動作の起こる状態をシフト状態，還元動作の起こる状態を還元状態と呼び，それらの両方の動作が起こる状態をシフト・還元状態と呼ぶ。状態sがシフト・還元状態であるかあるいは2種類以上の還元動作の起こる還元状態であるならば，状態s

を不適合状態と呼ぶ。$L(s)=\{a, y\}$ かつ $A_s(a)=shift$, $A_s(y)=reduce\ p$ なる条件を満足しない不適合状態のみを先読み状態と呼ぶ。この条件を満足する不適合状態は入力が a でないことを確認すれば，次の動作を $reduce\ p$ とできる(ただし，これは誤り検出遅れを考慮した場合である)ので，後で提案するパーザではこの解析手順を使用する。

重要な定理を説明する前に次の集合を定義しておく。

〔定義4．1〕

(1) $G_s^{-1}(X_1, X_2, \cdots, X_n)=\{s_0 \mid Gs_0(X_1)=s_1, Gs_1(X_2)=s_2, \cdots, Gs_{n-1}(X_n)=s_n; s=s_n, n\geqq 0\}$

ただし，$G_s^{-1}(\varepsilon)=\{s\}$ とする。

(2) $NEXT(s, p)=\{s' \mid s''\in G_s^{-1}(Xp_1 \cdots Xpnp), G_{s''}(Ap)=s'\}$

〔定理4．1〕

$s_1, s_2 \in NEXT(s, p)$ なる s_1, s_2 が $s_1 \gtrless s_2$ ならば，

$$s_1', s_2' \in G_s^{-1}(Xp_1 \cdots Xpnp)$$

$$Gs_1'(Ap)=s_1, \quad Gs_2'(Ap)=s_2$$

$$s_1' \gtrless s_2'$$

なる s_1', s_2' が必ず存在する。

(証明)

$s_1'=s_2'$ と仮定すると $G_{s1'}(Ap)=G_{s2'}(Ap)$ となり矛盾が生じる。

逆は明らかである。故に，定理の成立は明らかである。

(証明終)

4．3 新しい $LR(k)$ パーザ

本節で提案する新しいLR(k)パーザをパーザ π_L と呼ぶ。パーザ π_L ではシフト行列 N_s，参照行列 N_r，先読み行列 N_l，還元表 T を使用する。シフト行列 N_s はシフト，還元動作の決定と還元ループから出るタイミングの決定を行い，参照行列 N_s は還元動作後の行先状態を決定する。還元表Tはパーザ π_L の個々の還元動作において必要なスタックからのポップアップ数，出力等の情報をもつ。

4・3・1　新しいLR(k)パーザの解析表

本項では3つの行列N_s, N_r, N_l と還元表Tの定義と構成法について述べる。

〔定義4・2〕

行列N_s, N_r, N_l はLR(k)解析表から次のように構成される。

(1) 先読み状態でないシフト状態 s に対して，

$$A_s(a)=shift,\ G_s(a)=s'$$

ならば，$N_s(s, a)=s'$ とする。ただし，$As(\$)=accept$ なる状態 s に対しては$N_s(s, \$)=s_F$ とする。

(2) NEXT(s, p)がシングルトンでないとき，

$$s'\in G_s^{-1}\ (Xp_1 \cdots Xpnp),\ G_s'(Ap)=s''$$

なる s, s', s'' に対して，$N_r(s, s')=s''$ とする。

この参照行列Nrの列ラベルに対応する状態を参照状態と呼ぶ。

(3) 先読み状態 s に対して，$A_s(y)=reduce\ p$ ならば，

$N_l(s, y)=p$ とし，$A_s(y)=shift$ かつ$G_s(a)=s'$, $y=ay'$ ならば，

$N_l(s, y)=s'$ とする。

(4) 以上で決定した要素以外の要素をすべて0とする。

〔定義4・3〕

還元表Tは各還元状態 s に対して，EX_s, OP_s, PU_s, GT_s の項の内容をもつ。

(1) EX_s は還元状態 s が還元ループの出口となるか否かを表わすものであって，$EX_s=0$ ならば，出口とならないことを表わし，$EX_s=1$ ならば，シフト行列N_s により出口となるか否かを決定する。

(2) OP_s は出力すべき生成規則の番号を表わす。

(3) PU_s はスタックからポップアップするべき参照状態の個数を表わす。

(4) GT_s は行先状態を決定するものであって，$GT_s=0$ ならば，参照行列N_r によって行先状態を決定し，そうでなければGT_s 自身が行先状態となる。

次に，LR(k)解析表から還元表Tを構成する手順を与える。

<手順4・1> 還元表Tの構成法

(4・1－1) 還元状態 s がシフト・還元状態ならば，$EX_s=1$ とし，そうでなければ，$EX_s=0$ とする。

（４．１－２）　*reduce* pが起こる還元状態sに対して，

$G_{sp}(j-1)(X_{pj})=s_{pj}$, $1\leqq j\leqq n_p$, $s_{pnp}=s$

なる状態列$s\,p_1$, $s\,p_2$ … s_{pnp}中に含まれる参照状態の数を

PU_sとし，$OP_s=p$とする。

（４．１－３）　NEXT(s, p)=$\{s'\}$なる還元状態sに対して，$GT_s=s'$とし，

それ以外の還元状態sに対しては$GT_s=0$とする。

（手順終）

ただし，手順４．１において還元状態sから２種類以上の還元動作が起こるならば，各還元動作についてPU_s, OP_s, GT_sを求める。この場合は先読み行列N_lによって各還元動作が一意的に決定される。また，手順（４．１－２）でPU_sを決定する状態列が２種類以上存在しかつそれぞれのPU_sの値が異なる場合は，ポップアップするべきそれぞれの参照状態名の列を別の表に蓄える必要がある。しかし，このような場合は後述のプログラミング言語EULER, XPL, ALGOL 60に対しては存在せず，またLalondeらも取り上げていないので，今後この場合は議論しないものとする。

（例４．１）

例２．２の文法G_2に対する解析表は図４．１のようになる。参照状態はs_1, s_5, s_6, s_7であり，シフト・還元状態はs_2, s_9である。また，文法G_2の還元表TのGT_sはすべて０となる。

	a	+	*	(	)	$
s_I, s_5, s_6, s_7	s_4	0	0	s_5	0	0
s_8	0	s_6	0	0	s_{11}	0
s_2, s_9	0	0	s_7	0	0	0
s_1	0	s_6	0	0	0	s_F

(a) シフト行列N_s

	s_I	s_5	s_6	s_7
s_4, s_{11}	s_3	s_3	s_3	s_{10}
s_3, s_{10}	s_2	s_2	s_9	0
s_2, s_9	s_1	s_8	0	0

(b) 参照行列N_r

	EX_s	OP_s	PU_s	GT_s
s_4	0	6	0	0
s_{11}	0	5	1	0
s_3	0	4	0	0
s_{10}	0	3	1	0
s_2	1	2	0	0
s_9	1	1	1	0

(c) 還元表T

図4．1　文法G_1に対する新しいLR(k)解析表

4．3．2　新しいLR(k)パーザの解析手順

現在使用中の状態をs_{CT}，任意長の入力記号列を$I\tilde{N}$で表わし，最初のs_{CT}を初期状態s_Iとするとき，パーザπ_Lの解析手順は次のようになる。ただし，$Nl(s_{CT}, \tilde{IN}) = p$は還元状態$s_{CT}$の$p$番目の還元動作を決定する。

＜手順4．2＞　パーザπ_Lの解析手順

（4．2－1）　s_{CT}が先読み状態ならば手順（4．2－13）へ，そうでなければ次へ進む。

（4．2－2）　$N_s(s_{CT}, IN)$がsならば手順（4．2－3）へ，s_Fならば受理となり停止する。また，0ならば手順（4．2－5）へ進む。

（4．2－3）　s_{CT}が参照状態ならばs_{CT}をプッシュダウンし，そうでなければ次へ進む。

（4．2－4）　INを取り除き，$s_{CT}=s$とする。手順（4．2－1）へ帰る。

（4．2－5）　s_{CT}が還元状態ならば次へ進むが，そうでなければ誤りとなり停止する。

（4．2－6）　PU_s個の参照状態をポップアップし，OPs_{CT}を出力する。

（4．2－7）　$GTs_{CT}=0$ならば$s_{CT}=Nr(s_{CT}, s_{TOS})$とし，$GTs_{CT}\neq 0$ならば$s_{CT}=GTs_{CT}$とする。

（4．2－8）　s_{CT}が先読み状態ならば手順（4．2－13）へ，そうでなければ次へ進む。

（4．2－9）　s_{CT}が還元状態ならば手順（4．2－11）へ，そうでなければ次へ進む。

（4．2－10）　$N_s(s_{CT}, IN)$がs，s_Fならば手順（4．2－2）と同じ，0ならば誤りとなり停止する。

（4．2－11）　$EXs_{CT}=0$ならば手順（4．2－6）へ，$EXs_{CT}=1$ならば次へ進む。

（4．2－12）　$N_s(s_{CT}, IN)$がs，s_Fならば手順（4．2－2）と同じく，0ならば手順（4．2－6）へ進む。

（4．2－13）　$N_l(s_{CT}, \tilde{IN}) = p$ならば手順（4．2－6）へ，$s$ならば手順（4．2－3）へ，0ならば誤りとなり停止する。

（手順終）

〔定理４．２〕

パーザπ_Lは，正当なＬＲ(k)パーザであり，パーザπ_Lの誤り検出位置はパーザξ_Lと等価である。

（証明）

$y \in L(s)$, $A_S(y)=shift$, $G_S(a)=s'$, $y=ay'$

なるy'に対して，$y'=\varepsilon$ならば$N_S(s, a)=s'$であり，$y' \neq \varepsilon$ならば$N_l(s, y)=s'$であるから（定義４．２），入力文が正しければシフト先読み状態におけるパーザπ_Lのシフト，還元動作の決定はパーザξ_Lと明らかに等価である（手順４．２）

ただし，

$$L(s)=\{a, y\},\ A_S(a)=shift,\ A_S(y)=reduce\ p$$

なるシフト・還元状態sに対して，パーザπ_Lでは入力がaでなければ，yに関係なく次の動作を還元動作とする（手順４．２－5)。従って，この状況において入力がyでもないとすれば，パーザπ_Lは手順（４．２－１０）で誤りを検出するまでに何回かの還元動作を実行し，パーザπ_Lより誤り検出が遅れる可能性がある。しかし，この場合でもパーザπ_Lが誤り検出までにシフト動作を実行することはないので，パーザπ_Lとパーザξ_Lの誤り検出位置は等価である。また，文法の要素はスタックに入れなくてもよいので[6]，パーザπ_Lとパーザξ_Lのシフト動作も等価であるといえる。

ＬＲ(k)パーザの還元動作は次の行先状態を決定するためにだけスタック操作を行うのであるから，パーザπ_Lの還元動作が正当な行先状態を決定しかつ正当なスタック操作を実行することを証明する。

$$s_I X_1 s_1 \cdots\cdots X_n s_n$$

なるスタックの内容をもつパーザξ_Lが$reduce\ p$を実行すれば，その内容は

$$s_I X_1 s_1 \cdots\cdots X_{n-n_p}\ s_{n-n_p} A_p s'$$

$$s \leftarrow G s_{n-n_p}(A_p)$$

と変化する。パーザπ_Lでは還元状態s_nの$reduce\ p$に対する行先状態をＮＥＸＴ(s_n, p)であらかじめ決定しているので，ＮＥＸＴ(s_n, p)がシングルトンの場合は$GT\,s_n=s'$により行先状態はs'となり，またＮＥＸＴ(s_n, p)がシングルトンでない場合も$GT\,s_n=0$かつ$N_r(s_n, s_{n-n_p})$により行先状態はs'となる（定義４．２，手順（４．１－３），手順（４．２－7））。従って，パーザπ_Lの行先状態の決定は正当である。パーザπ_Lは参照状態のみをスタックに入れるので，スタック操作の

正当性は参照状態のみに着目すればよい。まず，参照状態のプッシュダウンのタイミングについては明らかである。次は，参照済の参照状態の除去についてはパーザξ_Lがポップアップした。

$$X_{n-n_p+1}{}^{s}{}_{n-n_p+1}\cdots X_n{}^{s}{}_n$$

なる状態列中の参照状態数をPU_{s_n}として還元表Tに与えてあるので（手順４．１－２））, この除去も明らかに正当である （手順（４．２－６））。

連続還元動作の終了，続行の正当性についても手順（４．１－１），手順（４．２－８）から手順（４．２－１３）より明らかである。また，π_Lとξ_Lは受理についても明らかに等価であるから，定理の成立は明らかである。

（証明終）

（例４．２）

文法G_2の入力（$a+a$）$*$ aに対するパーザπ_Lの解析例を以下に示す。ただし，計算状況の第４項目は使用中の状態を表わす。

$$(s_I, (a+a)*a\ \$, \varepsilon, s_I)$$

$$\vdash (s_I\ s_3, a+a)*a\ \$, \varepsilon, s_5)$$

$$\vdash (s_I\ s_3, +a)*a\ \$, \varepsilon, s_4)$$

$$\vdash (s_I\ s_5, +a)*a\ \$, 6, s_8)$$

$$\vdash (s_I\ s_5, +a)*a\ \$, 64, s_2)$$

$$\vdash (s_I\ s_5, +a)*a\ \$, 642, s_8)$$

$$\vdash (s_I\ s_5\ s_6, a)*a\ \$, 642, s_6)$$

$$\vdash (s_I\ s_5\ s_6,)*a\ \$, 642, s_4)$$

$$\vdash (s_I\ s_5\ s_6,)*a\ \$, 6426, s_3)$$

$$\vdash (s_I\ s_5\ s_6,)*a\ \$, 64264, s_9)$$

$\vdash$ (s_I s_5 ,) * a \$, 6 4 2 6 4 1 , s_8)

$\vdash$ (s_I s_5 , * a \$, 6 4 2 6 4 1 , s_{11})

$\vdash$ (s_I , * a \$, 6 4 2 6 4 1 5 , s_3)

$\vdash$ (s_I , * a \$, 6 4 2 6 4 1 5 4 , s_2)

$\vdash$ (s_I , a \$, 6 4 2 6 4 1 5 4 , s_7)

$\vdash$ (s_I , \$, 6 4 2 6 4 1 5 4 , s_4)

$\vdash$ (s_I , \$, 6 4 2 6 4 1 5 4 6 , s_{10})

$\vdash$ (s_I , \$, 6 4 2 6 4 1 5 4 6 3 , s_2)

$\vdash$ (s_I , \$, 6 4 2 6 4 1 5 4 6 3 2 , s_1)

$\vdash$ (s_I , \$, 6 4 2 6 4 1 5 4 6 3 2 , s_F)

= 受　理

4．4　解析表の縮小法

パーザ π_L の解析表で特に大きくなるのは，行列 N_s，N_r であるので，本節ではこの二つの行列の縮小法を中心に説明する。

行列 N_s の縮小化に対して，次の特徴が利用される。ただし，ラベル s の行（ラベル a の列）を s 行（a 列）と呼び，状態名 s の要素を s 要素と呼ぶ。

（特徴1）

$N_s(s, a)=s'$ なる s' 要素を1個だけもちほかの要素がすべて0である s 行。

（特徴2）

$N_s(s, a)=s$ なる要素の種類が1個である a 列。

例4．1の行列 Ns では s_2，s_9 行が特徴1を満足し，またすべての列が特徴2を満足する。

＜手順4．3＞　行列 N_s の縮小法

（4．3－1）　特徴1を満足する s 行を行列 N_s より除く。ただし，この手順で失われる $G_s(a)=s'$ の情報は別の表 T_s に蓄える。

（４．３－２）　特徴２を満足するa列とそれ以外の列により行列を二つに分割し，それぞれを行列N_{s1}，N_{s2}と呼ぶ。

（４．３－３）　行列N_{s1}，N_{s2}から０の要素のみをもつ行（例）を除く。

（４．３－４）　$N_{s1}(s, a)=s'$なるa列に状態sを対応させ，s'要素を1に変換する。

（４．３－５）　行列N_{s1}，N_{s2}に対して，同一行（同一列）を併合する。

（手順終）

手順４．３が行列N_sに対する等価な変換であることは明らかである。

（例４．３）

例４．１の行列N_s（この行列は行列N_{s2}をもたない）は手順４．３により図４．２のように変換される。ただし，行列N_{s1}の列ラベルには対応する状態名を与えてある。

	((s_5) $a(s_4)$	$)(s_{11})$	$+(s_6)$	$\$(s_F)$
s_{Π}, s_5, s_6, s_7	1	0	0	0
s_8	0	1	1	0
s_1	0	0	1	1

(a)　N_{s_1}

s_2, s_9	*	s_7

(b)　T_s

図４．２　縮小された行列N_s

図４．１の行列N_sでは，(列，a列は併合不可能であるが，手順４．３によりこれらの列は併合可能となる。また，手順（４．３－１）により＊列は除去可能となる。なお，縮小手順（以後の縮小手順も含む）の有効性は５．４節でまとめて説明する。

次に，行列N_rの縮小法を説明する。

行列N_rは還元動作からの行先状態が一意的でない場合に使用されるものであるから，行列N_rの各行には２種類以上のs要素が含まれる。従って，特徴２は使用できない。しかし，行列N_rの一つの行に含まれるs要素の種類は数種に限定される（特徴３）。

また，行列N_rの0要素は組合せ禁止としてもよいので[5]，次の縮小化が可能である。

＜手順４．４＞　行列N_rの縮小法

（４．４－１）　s行のs要素をその行に現れる回数の多い順に並べたものをs_1，s_2，…，s_n（$n \geqq 2$）とするとき，それぞれの状態名を１，２，…，nに対応させ，その対応表を作成する。

（４．４－２）　各s要素を対応する整数値に変換する。

（４．４－３）　同一行（同一列）を併合する。

（４．４－４）　組合せ禁止要素を利用して，同一行（同一列）を併合する。ただし，0の要素の最も多い行（列）を併合可能な0の要素の最も少ない行（列）に併合する方法で行う。

（手順終）

（例４．４）

例４．１の行列N_rは，同一行（同一列）の併合による縮小結果を確かめるほど大きくないが，手順４．４により行列N_rの要素の種類は図４．３のように６種類から３種類に減少する。

s_4, s_{11}	1	1	1	2	$s_3 \cdots 1$, $s_{10} \cdots 2$
s_3, s_{10}	1	1	2	0	$s_2 \cdots 1$, $s_9 \cdots 2$
s_2, s_9	1	2	0	0	$s_1 \cdots 1$, $s_8 \cdots 2$

図４．３　縮小された行列N_r

次に，還元表Tの縮小法について述べる。

還元表Tも同一行があれば，併合した方が空間的に有利である。しかし，この同一行はたとえ意味処理に関係しない$A \rightarrow B$なる単一生成規則（unit production）番号を0としてもさほど多くは存在しない。従って，出力項だけを別の表に蓄え，残った還元表Tの同一行を併合する縮小化を行う。

（例4．5）

例4．1の還元表Tは，出力項を除くことによって，図4．4のように縮少される。

	EX_s	PU_s	GT_s
s_4, s_3	0	0	0
s_{11}, s_{10}	0	1	0
s_2	1	0	0
s_9	1	1	0

図 4．4　縮小された還元表T

4．5 結　　言

本章では，従来のLR（k）パーザξ_Lの還元動作における行先状態の一意性についての条件を求め，その条件を利用した新しいLR（k）パーザの解析表，解析手順を定義した。そして，このパーザπ_Lで特に大きくなるシフト行列N_S，参照行列N_Tの二つの行列に対して，効率的な縮小法を提案した。パーザπ_Lはパーザξ_Lの誤り検出位置を保存するので，従来の縮小法のような誤り検出能力の低下は起らない。

本縮小法による有効性は，次の二つの特長で示される。

(1) 解析表として，行列形式のものを使用し，更にパーザπ_Lの解析手順では還元動作における冗長なスタック操作を除去した。

(2) 行列の配列の特徴を利用して縮小化を行い，特に行列要素を表現するビット数を軽減した。

以上より，本章の縮小法で得られたパーザπ_Lはコンパクト性，適用性，誤り検出能力，高速性に対しても十分有効なものであると思われる。

第5章　パーザの自動生成システム

5・1　諸　　言

本章では，パーザπ_P，π_Lの自動生成システムを実用プログラミング言語EULER[(2)]，XPL[(20)]，ALGOL 60[(77)]に対して適用し，パーザπ_P，π_Lのコンパクト性，高速性に対する実証を行う。

5・2節では，パーザの解析表を能率的に構成するために，解析表に対応するグラフを与えられた文法に対して作成する。このグラフは文法の検定，修正にも利用できる。弱順位パーザに対してはグラフRG，LG，LR(k)パーザに対してはGOTOグラフを定義し，それぞれのグラフを使用したパーザπ_P，π_Lの自動生成システムを開発する。5・3節では，EULER，XPL，ALGOL 60に対するパーザπ_P，π_Lの自動生成シミュレーションを行い，各文法に対するパーザπ_P，π_Lの記憶量を計算する。また，XPLに対して，パーザπ_P，π_Lの解析シミュレーションを行い，パーザπ_P，π_Lの解析速度の高速性を確かめる。更に，パーザπ_Pについては，従来のパーザξ_Pの縮小法である1対，2対，2組の関数に対する記憶量と比較し，そのコンパクト性を確かめる。パーザπ_Lについては，従来のパーザξ_Pの縮小法であるLalonde[(59)]，Anderson[(62)]，Jeliat[(64)]，によるパーザの記憶量と比較し，そのコンパクト性を確かめる。5・4節では，第3章，第4章で与えたパーザξ_P，ξ_Lの縮小法に対する評価，特に縮小手順の理論的有効性を確かめる。

5・2　自動生成システム

パーザの解析表を能率的に構成するために解析表に対応するグラフを与えられた文法に対して作成する。このグラフは文法の検定，修正にも利用される。本節では，弱順位パーザに対してグラフRG，LG，LR(k)パーザに対してGOTOグラフを提案し，それぞれのグラフを使用した自動生成システムを説明する。

5・2・1　弱順位パーザの自動生成システム[(39)～(54)]

グラフKを$K=(N, R)$で表わす。ここで，Nはノードの有限集合を表わし，Rは集合N上の関係を表わす。$a, b \in N$，$(a, b) \in R$ならば，ノードaか

らノードbにアークが引かれているものとする。

＜手順５．１＞　ＲＧ，ＬＧの構成法

（５．１－１）　文法Gp＝（V_N，V_T，P，So）に対して，グラフ（V_1，ρ），（V_2，λ）を構成し，以下の手順を実行したものをそれぞれＲＧ，ＬＧ，と呼ぶ。ここで，V_1，V_2は次のような集合である。

$$V_1 = \{X \mid A \rho^* X,\ A \in V_N\}$$

$$V_2 = \{Y \mid A \lambda^* Y,\ A \in V_N\}$$

（５．１－２）　$a \notin V_1$，$b \notin V_2$，$a\ \mu\ b$なるbをグラフ（V_2，λ）のノードとする。

（５．１－３）　グラフ（V_1，ρ）のノード　とグラフ（V_2，λ）のノードYにそれぞれμ_R（X），μ_R（Y）を隣接集合として加える。ここで，

$$\mu_R(X) = \{Y \mid X \mu Y\}$$

$$\mu_L(Y) = \{X \mid X \mu Y\}$$

である。ただし，μ_R（So）＝ μ_R（So）＝｛ $ ｝である。

（手順終）

以後，ＲＧのノードAから第k番目にあるすべての子孫を要素とする集合$R^k(A)$で表わし，

$$R_N^k(A) = R^k(A) \cap V_N$$

$$R_T^k(A) = R^k(A) \cap V_T$$

とする。また，ＬＧについても集合$L^k(A)$，$L_N^k(A)$，$L_T^k(A)$を同様に定義する。

＜手順５．２＞　ＲＧ，ＬＧによる順位関係の決定

（５．２－１）　ＲＧのすべてのノードXと$Y \in \mu_R(X)$なるすべてのYに対して$X \doteq Y$とし，ＬＧのすべてのノードYと$X \in \mu_L(Y)$なるすべてのXに対して$X \doteq Y$とする。

（５．２－２）　ＲＧのすべてのノードAによる$X \in R^+(A)$なるすべてのXと$Z \in \mu_R(A)$，$Y \in L^*(Z)$なるすべてのYに対して

$X \geqslant Y$とする。

（５．２－３）　ＬＧのすべてのノードAによる$X \in L(A)$なるすべてのXと$Y \in L^{+}(A)$なるYに対して$X \leqslant Y$とする。

（手順終）

〔定理５．１〕

手順５．２は，文法 G_p のすべての順位関係を決定する。

（証明）

ＲＧは$V_1 \times V$上のすべての$\doteq$関係，またＬＧは$V \times V_2$と$\overline{V}_1 \times \overline{V}_2$上（手順（５．１－２））のすべての$\doteq$関係を手順（５．２－１）で与える。よって，ＲＧ，ＬＧが手順（５．２－１）において$V \times V$上のすべての$\doteq$関係を与えることは明らかである。ここで，$\overline{V}_1$，$\overline{V}_2$はそれぞれVに対するV_1，V_2の補集合を表わす。また，ＲＧ，ＬＧのすべてのノードAに対して

$$R^k(A) = \{X \mid A \rho^k X\}$$
$$L^k(A) = \{Y \mid A \lambda^k Y\}$$

が成立するので，ＲＧ，ＬＧが手順（５．２．２），（５．２－３）において$V \times V$上のすべての　$\geqslant$，$\leqslant$関係を与えることは明らかである。

（証明終）

＜手順５．３＞　ＲＧ，ＬＧによる弱順位文法の検定

（５．３－１）　$X \in R^{+}(A)$，$Y_1 \in \mu_R(A)$，$Y_2 \in \mu_R(X)$なるすべてのY_1，Y_2に対して

$$L_T^{*}(Y_1) \cap L_T^{*}(Y_2) = \phi$$

が成立するかどうかを調べ，もし成立しなければ，対応する文法は弱順位文法でない，そうでなければ，次へ進む。

（５．３－２）　$A \rightarrow \alpha X \beta$，$B \rightarrow \beta \in P$なるすべての$X$，$B$に対して，

$$X \in \mu_L(C),\ B \in L_N^{*}(C)$$

なるCがLGに存在するかどうかを調べ，もし存在しなければ，対応する文法は弱順位文法である。そうでなければ，対応する文法は弱順位文法でない。

（手順終）

〔定理５．２〕

手順５．３は，文法 G_p が弱順位文法かどうかを決定する。

（証明）

必要：手順（５・３－１）において，

$$a \in (L_T^*(Y_1) \cap L_T^*(Y_2))$$

なるaが存在するとき（図５・１－（ａ））, $X \leqq a$かつ$X > a$となるので，定義２・１の(1)に反する。手順（５・３－２）において，

$$X \in \mu_L(C),\ B \in L_N^*(C)$$

なるノードCが存在するとき（図５・１－（ｂ））, $X \leqq B$となるので，定義２・１の(2)に反する。

十分：　$X < a$が成立するのは

$$X \in R^+(A),\ Y_1 \in \mu_R(A),\ a \in L_T^*(Y_1)$$

なるA，Y_1が存在する場合に限る。$X \leqq a$が成立するのは

$$X \in \mu_L(Y_2),\ a \in L_T^*(Y_2)$$

なるY_2が存在する場合に限る。よって，手順（５・３－１）の条件が成立する場合は明らかに

$$\leqq \cap > = \phi$$

である。また，手順（５・３－２）のノードCがＬＧに存在しなければ，$X \leqq B$なる関係が存在しないことも明らかである。故に，手順５・３は文法Gpが弱順位文法かどうかを決定する。

（証明終）

（ａ）　ＲＧ　　（ｂ）　ＬＧ

図５・１　証明のためのＲＧ，ＬＧ

（例５．１）

例２．１の文法 G_1 に対するＲＧ，ＬＧを図５．２に示す。図５．２において，ＲＧのノードＣとＦ∈$\mu_R(C)$より，

$$R^+(C)=\{D,\ E,\ b,\ )\}$$

$$L^*(F)=\{F,\ \div\}$$

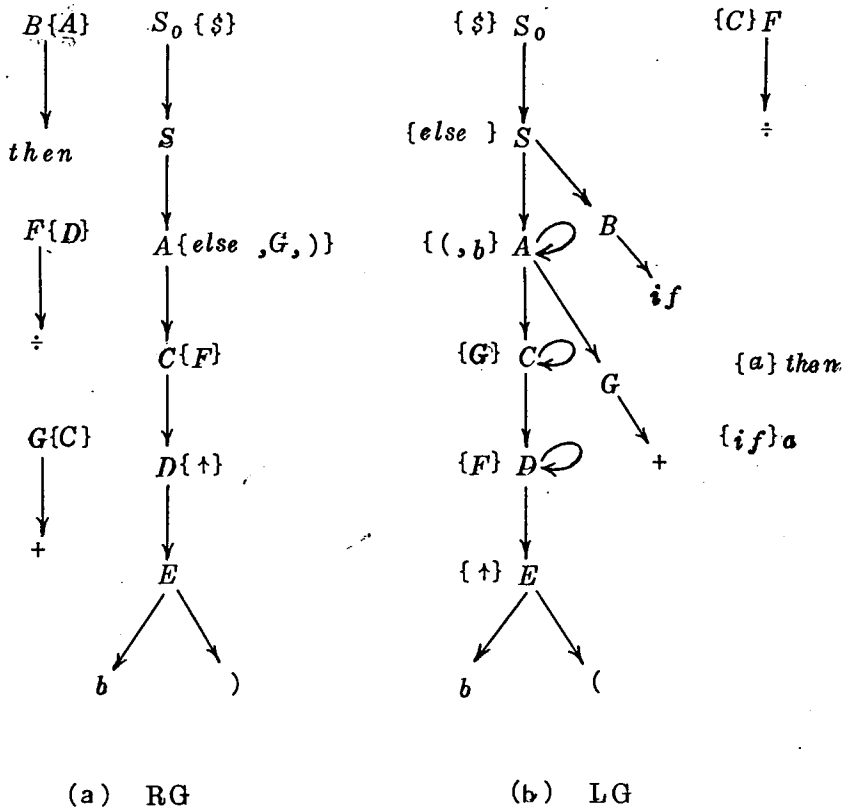
の要素はすべて≫関係をもつ。また，このＲＧ，ＬＧは

$$L_T^*(else,\ G,\ )\cap L_T^*(F)\cap L_T^*(\uparrow)$$

$$=\{else,\ +,\ )\}\cap\{\div\}\cap\{\uparrow\}$$

$$=\phi$$

であり，手順（５．３－１）の条件を満足する。更に，$C\to D$，$C\to CFD$なる生成規則に対して，$F\in\mu_L(D)$であるが，$C\notin L_T^*(D)$となり，手順（５．３－１）の条件を満足しない。また，ほかの生成規則についても同様であるので，文法 G_1 は弱順位文法である。



(a) RG　　(b) LG

図５．２　文法 G_1 に対するRG，LG

以上のＲＧ，ＬＧの性質を利用したパーザπ_p の自動生成システムのフローチャートを図５．３に示す。

パーザπ_p の自動生成システムのプログラムステップ数は７１１ステップであり，まず入力文法G_p が与えられれば，グラフＲＧ，ＬＧを作成する。そして，手順５．３により入力文法G_p が弱順位文法か否かを調べ，もし弱順位文法でなければ，Mcaffee[28] による方法に従って文法修正を行う。この繰り返し操作により，文法が弱順位文法となれば，第３章で定義した集合 V_1 ， V_2 ， V_3 を手順５．２を利用して求め，これらの集合に従って，行列M_S ， M_r を構成する。更に，行列 M_S ， M_r については同一行（同一列）の併合を行う。そして，最後に行列 M_S ，M_r に対して手順（３．２）により対応関数を構成し，行列 M_S ， M_r を縮小する。

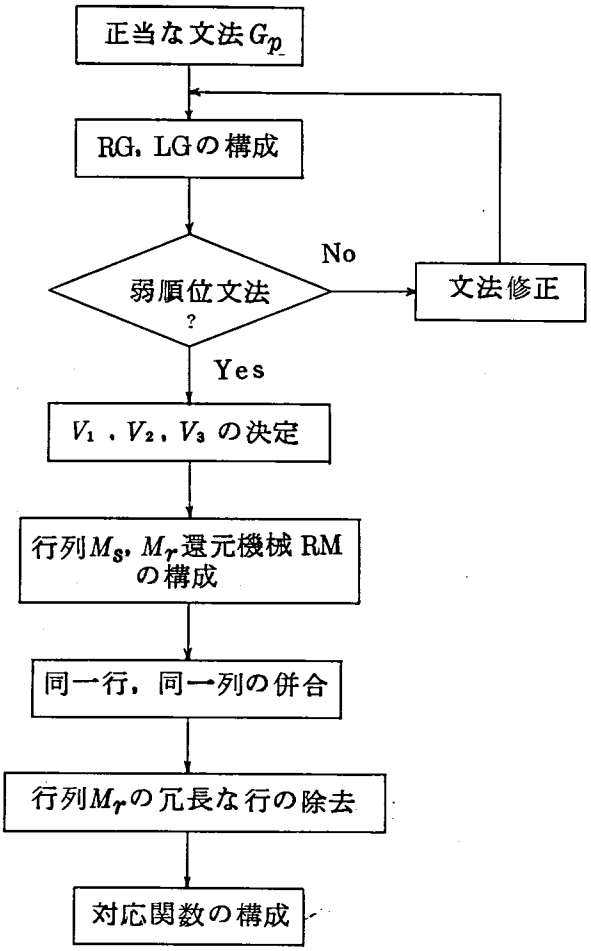


図５．３　パーザπ_pの自動生成のフローチャート

5・2・2　LR(k)パーザの自動生成システム[85)～(100)]

グラフ$K=(N, R)$に対して，アークラベルをもつグラフを$\vec{K}=(N, R, g)$で表わす。ここでgはRからアークラベルへの関数であって，アーク(s, s')がラベルXをもつとき，$g((s, s'))=X$と書く。

<手順5・4>　GOTOグラフの構成法

(5・4－1)　LR(k)解析表のすべての状態sに対するノードsを作る。

(5・4－2)　$G_s(X)=s'$ならば，$g((s, s'))=X$とする。

(5・4－3)　$s\in$NEXT(s, p)ならば，$g((s, s'))=p$とする。

(5・4－4)　$g((s, s'))=X$，$X\in V_N$なるアーク(s, s')を消去する。

(手順終)

$g((s_i, s_j))=a$なるアーク(s_i, s_j)をシフトアーク，$g((s_n, s_m))=p$なるアーク(s_n, s_m)を還元マークと呼ぶとき，次の定理が容易に導かれる。

〔定理5・3〕

(1)　シフトアークと還元アークをもつノードsあるいは2種類以上の還元アークをもつノードsは不適合状態である。

(2)　2本以上の同一還元アークラベルpをもつノードsのNEXT(s, p)はシングルトンでない。

(証明)　$g((s_i, s_j))=a$ならばあるxに対して

$$As_i(ax)=shift$$

となり，$g((s_n, s_m))=p$ならば先読み記号yに対して

$$As_n(y)=reduce\ p$$

となる。故に(1)は成立する。また，(2)は自明である。

(証明終)

〔定理5・4〕

状態sの先読み記号$y\in L(s)$はGOTOグラフのノードsからのパス上のシフトアークラベル列として求まる。

（証明）

明らかである。

（証明終）

このように，状態 s をＧＯＴＯグラフに対応するデータ構造で表わすことによって，シフト状態，還元状態，シフト・還元状態，不適合状態，先読み状態，更には参照状態と還元表 T の各情報を容易に求めることができる。(78)，(82)～(84)

（例５．２）

例２．２の文法 G_2 に対するＧＯＴＯグラフを図５．４に示す。

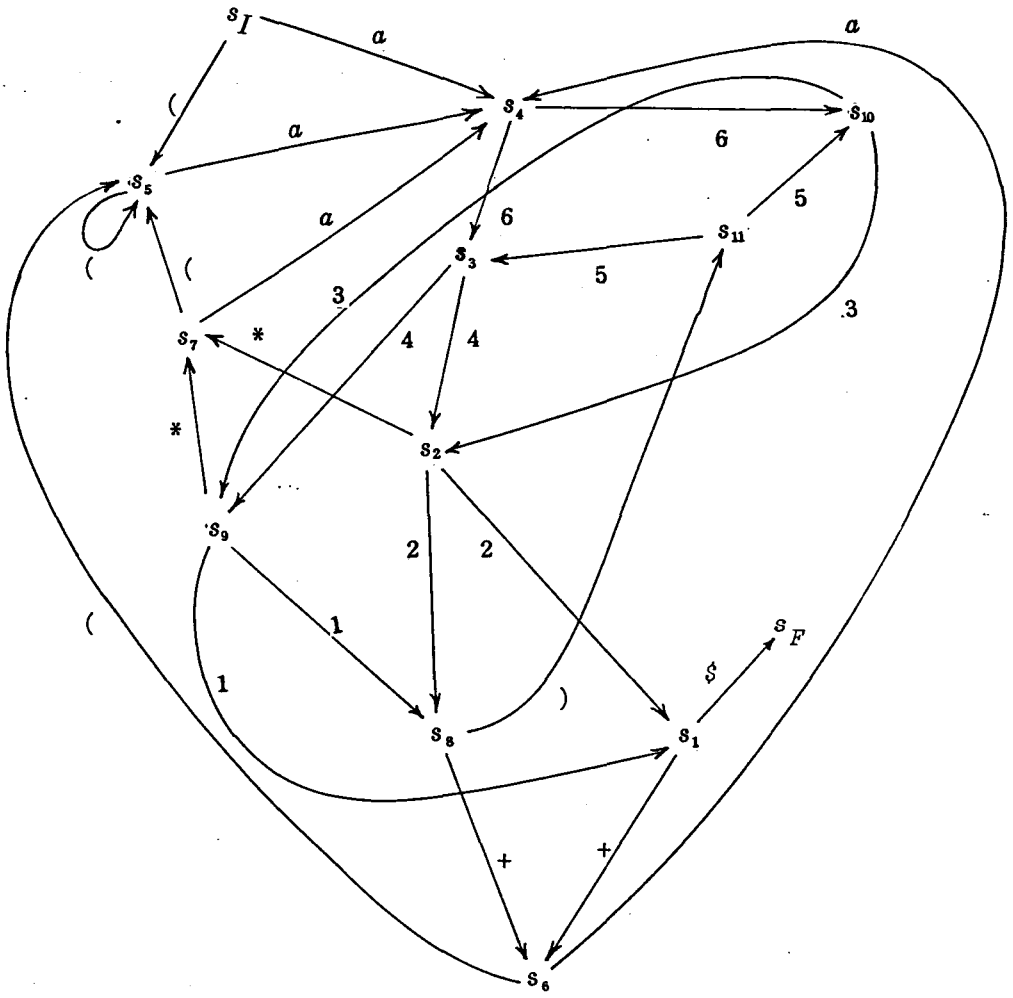


図５．４　文法 G_2 に対するGOTOグラフ

以上のＧＯＴＯのグラフの性質を利用したパーザπ_Lの自動生成システムのフローチャートを図５．５に示す。

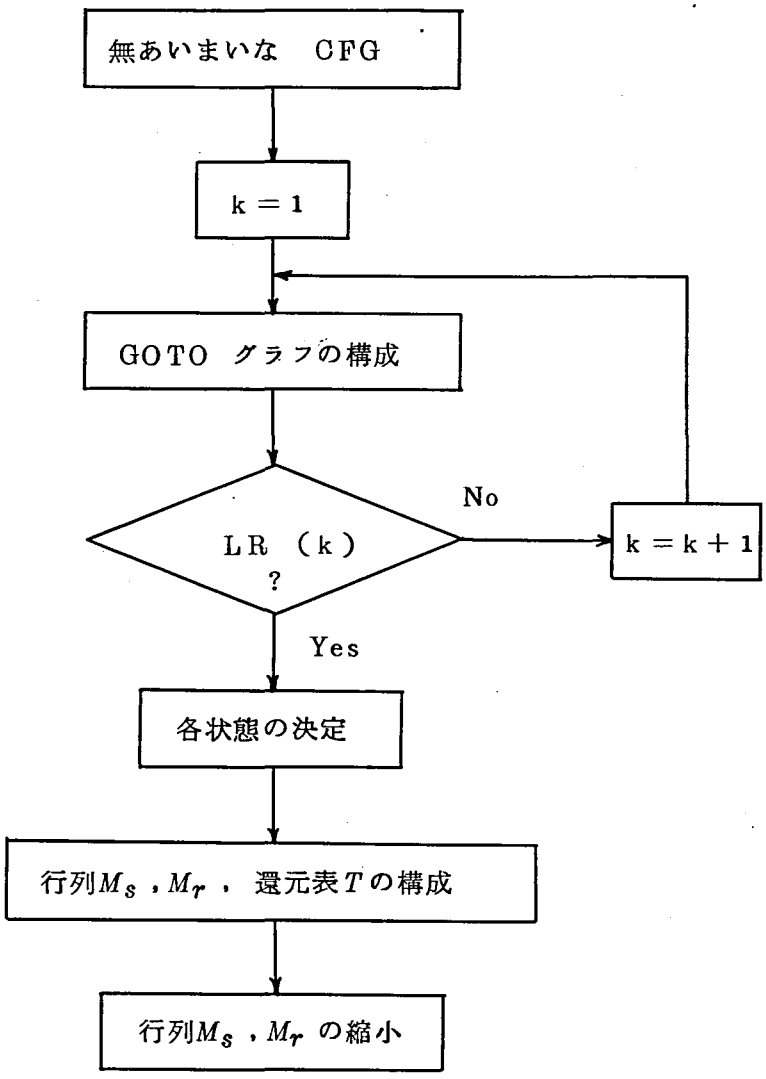


図５．５　パーザπ_Lの自動生成のフローチャート

パーザπ_Lの自動生成システムでは，まず$k=1$として，ＧＯＴＯグラフにより各状態の動作を決定する。そこで，もし入力文法がＬＲ(1)文法でなければ，定理５．４に従ってkを増加させる。このループは入力文法が無あいまいならば，必ず終了する[5]。次に，シフト状態，還元状態，シフト・還元状態，参照状態，先読み状態を決定し，解析表の構成へと進む。そして，第４章の縮小手順により解析表を縮小する。

5．3　実用プログラミング言語に対する適用結果 (40), (41), (45), (86)

パーザ π_P に対する適用結果として，文法修正後の各文法の値，自動生成時間，行列の大きさ，パーザ π_P の各部分の記憶量を表5．1に示す。

表5．1　パーザ π_P の自動生成シミュレーション結果

	EULER	XPL	ALGOL 60
文法の要素数			
V_N	44	61	115
V_T	76	47	55
P	120	118	202
自動生成時間 (s)	42	26	108
文法修正数	0	11	37
行列の大きさ			
M	120×76	108×17	170×55
M_s	4×5	8×9	9×9
M_r	14×22	19×20	29×30
各部分の記憶量 (bytes)			
行列	41	57	119
対応関数	329	219	255
RM	494	490	775

表５．１において，行列の各要素は１ビット，対応関数の各要素は８ビットで計算し，特にRHSについては表５．２のデータ構造とした。そして，ＲＨＳの長さの項のみ２ビットとし（EULER，XPL，ALGOL ６０ともRHSの最大長は４である），そのほかの項は８ビットとして計算した。

表５．２　RMのデータ構造

RHSの長さ	RHS	出力	LHS	行先状態

パーザπ_Lに対する適用結果として，表５．３に各文法の値，生成時間，状態数，行列の大きさ，各部分の記憶量を与える。ただし，行列N'を縮小したもので表わす。

図５．６，５．７にはパーザπ_Pとξ_P，パーザπ_Lとξ_Lに対する解析シシューレション結果を与える。これらの結果はXPLに対するパーザξ_P，π_P，ξ_L，π_Lの入力数と解析時間の関係を示したものであって，それぞれのパーザの解析プログラムのステップ数はそれぞれ66，74，101，110である。

表５．１，５．３の結果より，パーザπ_P，π_Lで最も大きくなる行列の記憶量がパーザπ_Pでは全体の約5.0％（EULER），7.4％（XPL），１０.４％（ALGOL ６０），またパーザπ_Lでは約３９％（EULER），３７％（XPL），４０％（ALGOL ６０）にまで縮小されていることが分かる。

図５．６，５．７の結果より，パーザπ_Pはパーザξ_Pより約３倍，パーザπ_Lはパーザξ_Lより約1.8倍速くなることが分かった。

文献(22)ではALGOL（ＪＩＳ3,０００）[38]に対して，２組の関係によるパーザFがパーザξ_Pより約2.7倍速くなることを示した。従って，対象とする文法は異るが，パーザπ_Pの解析速度はパーザFと同等かあるいはそれ以上のものになると思われる。また，パーザπ_Lの解析速度も若干ではあるがJoliat[64]のものより（パーザξ_Lより約1.7倍高速である）優れている。

表５．３　パーザ π_L の自動生成シミュレーション結果

	EULER	XPL	ALGOL 60
文法の要素数			
V_N	44	50	86
V_T	75	48	63
P	120	109	174
自動生成時間〔s〕	215	74	82
シフト状態数	85	97	130
還元状態数	122	112	179
シフト・還元状態数	16	24	25
参照状態数	58	43	77
全状態数	191	185	284
行列の大きさ			
N_s	85×75	97×48	130×63
N'_{s1}	15×15	16×14	22×21
N'_{s2}	9×6	18×8	22×12
N_r	82×58	62×43	136×77
N'_r	14×49	9×33	17×48
各部分の記憶量〔bytes〕			
行列	425	321	730
対応表	581	485	976
還元表	75	57	98

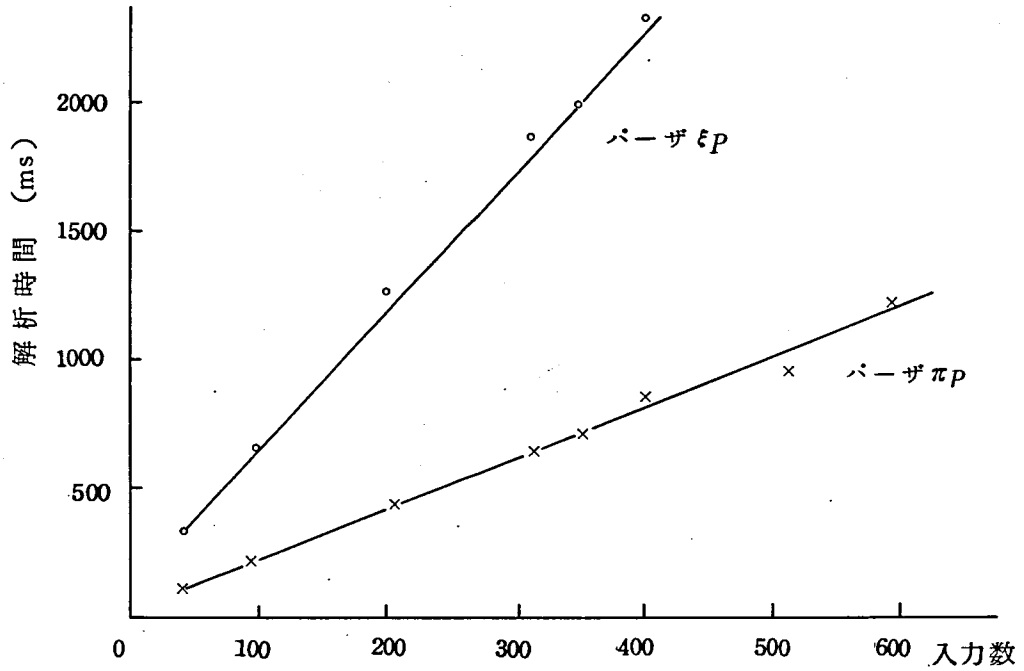


図５．６　弱順位パーザの解析シミュレーション結果

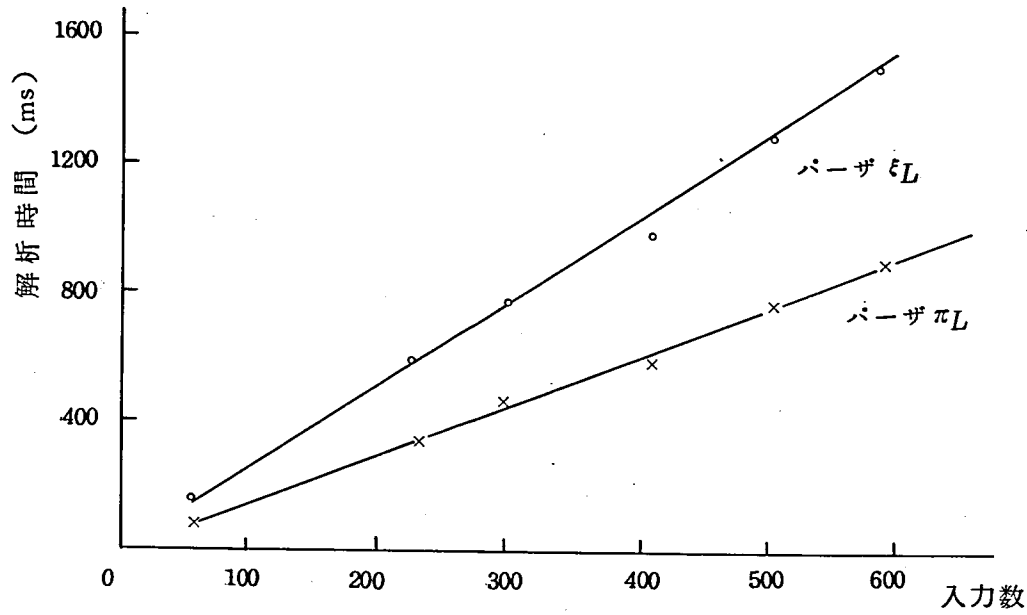


図５．７　ＬＲ(ｋ)パーザの解析シミュレーション結果

すでに提案されている弱順位行例の各縮小法と本方法を比較するために，次の四つのパーザ弱順位行列に対応する記憶量を計算し，その結果を表５．４に示す。ただし，V_N，$V_T \cup \{\$\}$の要素数をそれぞれ n_N，n_T とする。

パーザ ξ_P ： 第２章で説明した弱順位行列Mを使用するパーザであって，その記憶量を（ $n_N + n_T$ ）n_T ／4 バイトで計算した。このパーザは，各縮小法がどの程度となるかの目安となる。

パーザ π_1 ： １対の関数（ f，g ）によるパーザであって，(1)～(11)その記憶量を $n_N + n_T$ バイトで計算した。

パーザ π_2 ： ２対の関数（ f，g ），（ f'，g' ）によるパーザであって(5)，(9)，その記憶量を２（ $n_N + n_T$ ）バイトで計算した。

パーザ π_3 ： ２組の関数（f_r，g_r，δ_r），（f_s，g_s，δ_s ）によるパーザであって，(12)，(15) その記憶量を２（ $n_N + 2\,n_T$）バイトで計算した。

表５．４　パーザ πp の記憶量の比較

	EULER	XPL	ALGOL 60
弱順位行列	2,280	1,269	2,338
$M_s + M_r$ 対応関数	365	276	374
１対の関数	196	155	225
２対の関数	392	310	450
２組の関数	544	404	560

表５．４の結果より，本方法による縮小率は，２対の関数のものと同程度となることが分かる。

パーザ π_L の全体の大きさを比較するための対象として，良い縮小結果を与えている3種類のLR(k)パーザ L，A，J を選ぶ。

パーザ L ： 表形式の解析表を使用するLalonde[40]によるパーザであって，縮小法としては解析表の分割，還元動作の際のスタッフのポップアップ数の簡単化を行っている。

パーザ A ： パーザ L と同じ表形式の解析表を使用するAnderson[59]によるパーザであって，縮小法としては $A \rightarrow B$ なる単一生成規則の除去を中心としている。

パーザ J ： 行列形式の解析表使用するJoliat[64]によるパーザであって，縮小法としてはKnuth の解析表を一つの行列で直接表現し，誤り検出のみを行うBoolean 行列を別に作ることによって，行列を縮小している。

大きさの点では表形式の解析表を使用するパーザ L，A がパーザ J より49%～55%優れているが，解析速度については行列形式を使用するパーザ J がパーザ L，A より優れている。

表5．5に以上の記憶量を示し，パーザ π_L と比較する。なお，表中のパーザ A，J の一印はその値がAnderson[62]，Joliat[64]の論文において与えられていないことを示す。

表5．5　パーザ π_L の記憶量の比較

	EULER	XPL	ALGOL 60
パーザ π_L	1,081	863	1,804
パーザ L	1,606	1,250	2,821
パーザ A	—	1,182	2,434
パーザ J	—	1,836	—

表５．５より，パーザ π_L の大きさが従来の最小のものよりEULERで３３％，XPL で２７％，ALGOL 60で２６％優れていることが分かる。

５．４　縮小法に対する評価 (40), (86)

弱順位パーザの縮小法の特長は，四つあり，その一つはRMを導入することにより，弱順位行列を分割し，冗長な行を除去した点にある。ALGOL 60 の行列 M_r では，V_N に対応する１１５行のうち，５５行が除去された。また，この特長は対応関数 f_r の数の減少にも役立っている。

次の特長は，定義３．２の(3)の $B \lessdot a$ の関係を行列 M_s ではなく行列 M_r 上で表した点にある。例３．２の図３．３の行列 M_s, M_r からも分かるように，この関係をもし行列 M_s で表わしたとすれば，列ラベル

$$\{\$\},\ \{else,\ )\},\ \{\uparrow\},\ \{\div\}$$

が行列 M_r に残ったままで行列 M_s に追加されることになり，行列 M_s，M_r の和は大きくなる。この特長は，次のように理論的に説明できる。

定理３．１より

$$a \gtrdot b$$

ならば，

$$B\rho^{+}a,\ b \in LEFT(B)$$

なる B が在存するので，常に

$$LEFT(B) \subseteq RIGHT(a)$$

が成立する。従って，行列 M_r のラベル B の行とラベル a の行は，必要とする列ラベルを有効に共有することになる。ALGOL 60に対して，この特長を利用しないとすれば，行列 M_r，M_s はそれぞれ４８×３９，１５×３３の大きさになる。

第３番目の特長は，手順３．２の対応関数の構成法にある，この手順により解析手順は若干復雑になるが，行ラベル（列ラベル）を行番号（列番号）にのみ対応される目的の対応関数を順位関係の表現に対しても使用しているので，縮小化に対する有効性は極めて大きい，この対応関数による行列 M_s，M_r の記憶量の改善率は，EULERで約４５％，XPL で約５３％，ALGOL 60で約４１％となった。

パーザ π_P は，パーザ ξ_P の誤り検出位置を保存したパーザであるから，誤り

検出の等価性の点では文献(6),(28)の中等価の条件で弱順位行列を縮小した場合と同様である。しかし，この条件でALGOL 60の行列を縮小してもその大きさは63×39となり，対応関数も含めた弱順位行列の記憶量は6,162バイトにもなる。

最後の特長は，定義3．3により，

$$Ap \rightarrow \alpha pXp$$

に対応するRMのRHSが αp となることである。これはパーザ π_P の連続還元動作のスタック操作が αp をポップアップするだけでよいことを意味する。αpXp をポップアップし，Ap をプッシュダウンしていたパーザ ξp のスタック操作は大幅に簡単化される。ALGOL 60では20回程度の連続還元動作が起こるので，この特長も非常に重要なものである。

以上より，パーザ π_P についての評価をまとめると次のようにする。

(1) コンパクト性は，2対の関数と同程度のものとなる。

(2) 誤り検出能力は，パーザ ξp の誤り検出位置を保存する。

(3) 解析速度は，パーザ ξp より数倍速くなる。

(4) 適用上の広さは，すべての弱順位文法にわたる。

(5) 構成法は，RG，LGを利用しているので，極めて簡単である。

一般に関数による縮小法は，コンパクト性が極端に優れている反面，誤り検出能力と適用性が劣る。従って，第3章縮小法は，欠点の少ないより実際的な弱順位パーザの縮小法であるといえる。

次に，第4章のLR(k)パーザの縮小法に対する評価を与える。まず4．3節の特徴1，2，3の有効性を理論的に考察する。

LR(k)解析表の状態sの構成では，まず〔p，j〕，$1 \leqq j \leqq n_p$ なる部分状態を要素とする基礎集合(basis set)が作られ，次にその基礎集合の

$$〔p, j〕, Xp(j+1) \in V_N, Xp(j+1) \overset{*}{\Rightarrow} Aq\alpha$$

なるすべての部分状態に対して，〔q，0〕になる部分状態を要素にもつ閉包集合(closure set)が追加される。従って，異なる状態 s，s' のそれぞれの基礎集合の要素を

$$〔p, j〕, 1 \leqq j \leqq np$$

$$〔p', j'〕, 1 \leqq j' \leqq np'$$

とすると，

$$Xp(j+1) \overset{*}{\Rightarrow} Xp'(j'+1)\alpha'$$

$$(Xp'(j'+1) \overset{*}{\Rightarrow} Xp(j+1)\alpha \text{でも同じ})$$

あるいは

$$Xp(j+1) \overset{+}{\Rightarrow} Aq\beta\,,\quad Xp'(j'+1) \overset{+}{\Rightarrow} Aq\beta'$$

なる関係が成立するならば，状態 s ，s'のそれぞれの閉包集合は同じ部分状態（共通部分状態と呼ぶ）を要素にもつ。そして，

$$〔A \rightarrow \cdot X\alpha〕$$

なる共通部分状態より作られる

$$G_s(X) = s_n\,,\quad G_{s'}(X) = s'_n$$

なる新しい状態 s_n ，s'_n の基礎集合の要素は

$$〔A \rightarrow X \cdot \alpha〕$$

なる部分状態と一致する（すなわち，$s_n = s'_n$ となる）。ただし，これは閉包集合の共通分状態に対する場合であって，もし状態 s ，s'の基礎集合のいずれかに

$$〔B \rightarrow \beta \cdot X_T〕\,,\quad \beta \neq \varepsilon$$

なる部分状態が存在すれば，状態 s_n ，s'_nの基礎集合は一致しない（すなわち，$s_n \neq s'_n$ なる）。従って，状態 s ，s'の基礎集合の要素数が少ないほど，この状況に対して，$s_n = s'_n$ となる可能性は高くなる。

実用のプログラミング言語の各状態の基礎集合は，大半が数個の部分状態で構成されているので，上記の場合に対して，$s_n = s'_n$ となる可能性は極めて高いといえる。ALGOL 60 では$G_s(X)$がすべての状態 s に対して，常に一意的となるXがVの要素に対して41％となった。この比率はXとVをそれぞれ$X \in V_T$，V_T に限定すれば，57％（この値は後で説明する手順（4．3－1）の特長より更に65％に改善される）にもなるので，特徴2の有効性は高いといえる。以上は$X \in V_N$ になるXに対して，$G_s(X)$が一意的でない場合でもそれが数種類に限定されること，つまり特徴3の有効性も説明している。ALGOL 60 の行列 N_r の各行に現れるS要素の種類の平均数は約2.9であるので，特徴3の有効性も高いといえる。

状態 s に含まれる　〔p，j〕，$Xp(j+1) \in V_T$

なる部分状態のすべての$Xp(j+1)$がある一つの終端語aと一致するならば，状態sの行先状態は1個（$G_{\hat{s}}(a)$ で決定される）だけとなる（特徴1）。このような状態sも基礎集合の要素数の少ない（特に1個）状態に多く存在するので，実用プログラミング言語に対してもこの特徴をもつ状態sの存在性の高さが期待できる。ALGOL 60 の行列 N_Sでは，約38%の行が特徴1を満足した。この値は特徴2ほど多くはないが，特徴1を満足する行は行列 N_S より直接除去（手順（4．3－1）できるので，この程度の値でもその縮小効率は極めて高いといえる。

次に，縮小手順の有効性について述べる。まず，手順4．3は次の特長をもつ。

(1) 手順（4．3－1）の行の削除は，特徴2を満足する列および0の要素のみをもつ列を増加させる。

(2) 手順（4．3－2）の行列の分割は，0の要素のみをもつ行を作り，かつ同一行（同一列）の数も増加させる。

(3) 手順（4．3－4）のs'要素から1の要素への変換は，行列N_{S_1}の要素を2種類に限定し，併合可能な同一行（同一列）の数も増加させる。

(3)の特長により，行列N_{S_1}の各要素は1ビットで表現可能となる。

ALGOL 60 の行列 N_Sでは，手順（4．3－1）で50個の行が除去され，これにより特徴2をもつ列が3個，0の要素のみをもつ列が5個増加した。また，手順（4．3－2）により0の要素のみをもつ行が行列N_{S_1}，N_{S_2}で18個削除され，手順（4．3－2），（4．3－4）により同一行（同一列）が9（2）個増加した。この行列 N_S を単純に同一行（同一列）の併合だけによって縮小すれば，94×44（記憶量は4,653バイト）にしか縮小できないので，表5．3の結果と比較することで手順4．3の有効性は容易にか分る。手順（4．3－1）では，行列 N_S の情報を一部表 T_S に蓄えるが，この操作によって行列の一意的参照の特長が失われることはない。

手順4．4は次の特長をもつ，

手順（4．4－1）で求まる最大の整数lとするとき，行列 N_T の各要素は $l+1 \leqq 2^m$ を満足する最小の整数mによるビット数で表現可能となる。すなわち，手順（4．4－1），（4．4－2）は行列 N_T の各要素を表現するビット数を減少させる特長をもつ。また，これらの手順は行列 N_T の同一

行（同一列）の数を増加させる。

ALGOL　60の行列 N_T では，この l は9であるから，行列 N_T の要素は4ビットで表現可能となり，状態数が284個（この場合，各要素を表現するには9ビット必要となる）もあるALGOL　60に対してこの特長は有効である。また，同一行（同一列）はこれらの手順によって4(4)個増加した。組合せ禁止のみで行列 N_T を縮小すれば，21×52（記憶量は1,229バイト）にしか縮小できないので，表5．3の結果より手順4．4も有効であることが分かる。なお，手順4．4は自動的に縮小を行うために一意的な操作（整数値割当手順（4．4－1），組合せ禁止による同一行（同一列）の併合手順（4．4－4）を導入しているので，最適な縮小手順とはいえないが，結果としては表5．3のような良い結果が得られた。

ALGOL　60の還元表 T では EX_s，PU_s，GT_s が0となる場合が全体のそれぞれ86%，64%，73%もあるので，OP_s の項を除けば，同一行が全体の56%も存在することになり表5．3のような良い結果が得られた。

パーザ π_L の高速化の理由として，次が考えられる。XPL の場合，パーザ ξ_L の状態 s に対する $L(s)$，$S(s)$ の平均数がそれぞれ6.8，7.9であるのに対して，パーザ π_L ではそれらの探索がすべて一意的である。しかも，パーザ π_L の各還元動作では冗長な入力参照，スタック参照がそれぞれパーザ π_L より79%，45%除去されている。

以上より，パーザ π_L についての評価をまとめると次のようになる。

(1) コンパクト性は，従来の縮小法のものより，約26%～33%改善される。

(2) 誤り検出能力は，パーザ ξ_L の誤り検出位置を保存する。

(3) 解析速度は，表形式の解析表をもつパーザ ξ_L より約1.8倍速くなる。

(4) 適用上の広さは，すべてのLR(k)文法にわたる。

以上より，第4章で提案されたLR(k)パーザの縮小法は，十分実用的であるといえる。

5．5 結　　言

本章では，パーザ π_P，π_L の自動生成システムを開発し，実用プログラミング言語EULER，XPL，ALGOL　60に対するシミュレーションを行い，パーザ π_P，

π_L のコンパクト性，高速性に対する実証を行った。

具体的結果としては，次のようにまとめられる。

(1) パーザ π_P の記憶量は，従来の縮小法による2対の関数と同程度になった。

(2) パーザ π_P の解析速度は，パーザ ξ_P より約3倍高速になった。

(3) パーザ π_L の記憶量は，従来の縮小法によるものより約26%～33%改善された。

(4) パーザ π_L の解析速度は，パーザ ξ_L より約1.8倍高速になった。

第6章　結　　論

本論文によって，高速性，コンパクト性，適用性，有効な誤り検出能力を満足するパーザが自動生成可能となった。

第3章では，従来の弱順位パーザ ξ_P の連続還元動作に対する性質を定理3.1に与えたが，この性質はパーザ ξ_P の誤り検出位置を保存する新しい弱順位パーザ π_P の解析手順，および縮小化が容易なシフト行列，還元行列の構成を示唆した。また，連続還元動作を実行するために提案された還元機械RMは，行列の縮小化と解析速度の改善の両方に利用し，行列の行，列ラベルをそれぞれ行，列番号に対応させる対応関数も行列の縮小化に利用した。従って，二つの行列の縮小率は極めて高いものとなった。

第4章では，従来のLR(k)パーザ ξ_L の還元動作における行先状態の一意性の条件を定理4.1で与え，この条件を利用してパーザ ξ_L の誤り検出位置を保存した新しいLR(k)パーザ π_L の解析手順，解析表を定義した。行列の縮小化としては，それぞれの行列の配列の特徴を利用して，行列要素を表現するビット数を軽減したので，行列の大きさの縮小が十分でない場合でも，行列全体が占有する記憶量は極めて少なくなった。

第5章では，パーザ π_P，π_L の自動生成システムを説明し，実用プログラミング言語EULER, XPL, ALGOL 60に対するパーザ π_P，π_L の自動生成，解析シミュレーションを行った。そして，それらの結果より，パーザ π_P，π_L のコンパクト性，高速性に対する有効性を実証した。また，パーザ π_P はパーザ π_L よりコンパクト性が優れており，パーザ π_L はパーザ π_P より高速性，適用性が優れているので，小規模のパーザについてはパーザ π_P が，高速で大規模なパーザについてはパーザ π_L が適していると思われる。

以上の方法で自動生成されたパーザは，従来のパーザの誤り検出位置を保存しているので，誤り訂正，誤り回復能力をもつパーザに拡張可能である。従って，当面の研究課題は，パーザ π_P, π_L に対する有効な誤り処理法を提案することである。[100]～[103] この方法が与えられれば，プログラムの構文誤りの自動訂正，自動回復が可能となるので，プログラマの大きな助けとなる。更には，パーザ π_P，π_L に対する語い解析部，命令語生成部[105]～[107]を開発し，コンパイラ・ジェネレータを完成することが今後の残された課題である。

謝　　辞

本研究の全過程を通じ，直接懇切なる御指導，御鞭撻を賜わった大阪大学工学部通信工学教室手塚慶一教授に心より感謝の意を表する。

本研究にあたって，御指導，御教示を賜わった大阪大学工学部電子工学教室尾崎弘教授，精密工学教室牧之内三郎教授に深く感謝する。

徳島大学大学院工学研究科修士課程および徳島大学工学部情報工学教室において，電子，情報工学一般および各専門分野に関して御指導，御教示を賜わった徳島大学工学部電子工学教室故原田尚文名誉教授，情報工学教室島田良作教授に感謝する。特に，本研究の全過程を通じ，直接御指導をいただいた情報工学教室山本米雄講師に深く感謝する。

有益な御助言，御討論をいただいた大阪大学工学部通信工学教室滑川敏彦教授，熊谷信昭教授，中西義郎教授に対して厚くお礼申し上げる。

大阪大学工学部手塚研究室の真田英彦助教授，打浪清一助手，中西暉助手，大学院生西岡弘明氏，恩地幹雄氏，小玉浩二氏をはじめ，研究室の諸氏には種々の面でお世話になった。また，徳島大学工学部島田研究室の石田富士雄技官，大学院生秋友利彦氏，福岡一郎氏，卒業生麻植稔恵氏，小金由美子氏，芝野良一氏，山下勝則氏をはじめ，研究室の諸氏には熱心な御討論をいただいた。

ここに記して，以上の方々に深く感謝の意を表する。

参　考　文　献

(1) R.W.Floyd:"Syntax analysis and operator precedence", J.ACM, Vol.10, №.3, p.316 (1963).

(2) N.Wirth, H.Weber: "EULER:A generalization of ALGOL, and its formal definition: Part I, II," C.ACM, Vol.11, №.10, p.685 (1968).

(3) J.R.Bell: "A new method for determining precedence functions for precedence grammars," C.ACM, Vol.12, №.10, p.567 (1969).

(4) D.F.Martin: "A boolean matrix method for the computation of linear precedence functions," C.ACM, Vol.15, №.6, p.448 (1972).

(5) A.V.Aho and J.D.Ullman: "Error detection in precedence parsers," Meth.S.T., Vol.7, №.2, p.97 (1973).

(6) A.V.Aho and J.D.Ullman: "The theory of parsing, translation, and compiling," Prentice-Hall, Vol.I (1972), Vol.II (1973).

(7) 浅井清："順位関数をもつ順位文法," 情報処理, Vol.12, №.5, p.264 (1971).

(8) 浅井清："順位関数の存在について," 情報処理, Vol.13, №.4, p.218 (1972).

(9) 海尻，妹尾，打浪，手塚："誤り検出の遅延による順位関数実現法," 信学論 (D), Vol.59—D, №.11, p.778 (1976).

(10) 海尻，打浪，手塚："順位関数の新しい実現法について," 信学論 (D), Vol.59—D, №.11, p.786 (1976).

(11) 海尻，打浪，手塚："拡張弱順位関数," 情報処理, Vol.18, №.6, p.542 (1977).

(12) 青江，山本，原田，島田："弱順位関数の一構成法," 信学論 (D), Vol.60—D, №.1, p.72 (1977).

(13) 青江，山本，原田："誤り検出遅れのない弱順位関数について," 信学会オートマトンと言語研資, AL75—19, p.23 (1975).

(14) 青江，山本，原田，島田："弱順位関数の構成法について," 信学会オートマトンと言語研資, AL75—46, p.99 (1975).

(15) 青江，山本，原田，島田：“誤り検出場所を保存した弱順位パーザについて，”信学会オートマトンと言語研資，AL75−75，p.97 (1976).

(16) 青江，山本，原田，島田：“弱順位 parser の一構成法，”電気関係学会四国支部連合大会，13 − 13，p.253 (1975).

(17) D.F.Martin：“Boolean matrix methods for the delection of simple precedence grammars，” C.ACM, Vol.11, №10, p.685(1968).

(18) J.Teldman and D.Gries: “Translator writing systems，” C.ACM, Vol.11, №2, p.77 (1968).

(19) J.Fischer：“Some properties of precedence languages，” Proc. of ACM Symp. on Theory of Computing, p.181 (1969).

(20) W.M.Mckeeman, J.J.Horning and D.B.Wortman：“A compiler generator，” Prentice−Hall, (1970).

(21) A. Colmerauer：“Total precedence relations，” J.ACM, Vol. 17, №1, p.14 (1970).

(22) D.J.Rosenkrantz and R.E.Stearns：“Properties of determintic topdown grammars，” Information and Control, Vol. 17, №3, p.226 (1970).

(23) A.Learner and A.L.Lim：“A note on transforming context free grammars to Wirth Weber precedence Form，” Computer Journal, Vol.13, №2, p.142 (1970).

(24) J.B.Morris：“A result on the relationship between simple precedence languages and reducing transition languages，” Proc.of 2nd Annual ACM Symp.on Theory of Computing, p.73 (1970).

(25) J.D.Ichbiah and S.P.Marse：“A technigue for generating almost optimal Floyd−Evans productions for precedence grammars, C.ACM, Vol.13, №8, p.501 (1970).

(26) S.L.Graham：“Extended precedence languages, bounded right context languages, and deterministic languages，” Proc. of IEEE 11 th Annual Symp.on Switching and Automata Theory, p.175(1970).

(27) A.V.Aho, P.J.Denning and J.D.Ullman：“Weak and mixed strategy

precedence parsing," J.ACM, Vol.19, №2, p.225 (1972).

(28) J.Mcafee and L.Presser : "An algorithm for the design of simple precedence grammars," J.ACM, Vol.19, №3, p.385 (1972).

(29) J.N.Gray and M.A.Harrison : "Canonical precedence schemes," J.ACM, Vol.20, №2, p.214 (1973).

(30) N.A.Khabbaz : "Multipass precedence analysis," Acta Informatica, Vol.4, №1, p.77 (1974).

(31) R.Haskell : "Symmetrical precedence relations on general phrase structure grammars", Computer Journal, Vol.17, №3, p.234 (1975).

(32) H.B.Hunt, T.G.Sjymanski and J.D.Ullman : "Operations on sparse relations", C.ACM, Vol.20, №3, p.171 (1977).

(33) 井上謙蔵："右順位文法," 情報処理, Vol.11, №8, p.449 (1970).

(34) 井上謙蔵："順位言語の右順位解析," 情報処理, Vol.11, №4, p.231 (1970).

(35) 関本彰次："拡張右順位文法とその解析法," 情報処理, Vol.12, №9, p.543 (1971).

(36) 井上謙蔵："右順位文法と包含関係," 情報処理, Vol.13, №10, p.692 (1972).

(37) 浅井，富山："順位文法によるコンパイラの構成," 情報処理, Vol.14, №7, p.495 (1973).

(38) 日本工業規格："電子計算機プログラム用言語ALGOL," C6214 (1967).

(39) 青江，山本，原田，島田："パージングテーブルによる弱順位パーザの構成法," 情報処理, Vol.18, №5, p.438 (1977).

(40) 青江，山本，島田："順位パーザの実際的構成法," 信学論 (D), Vol. J 61-D, №6, p.387 (1978).

(41) 青江，山本，島田，原田："弱順位パーザに対する解析シミュレーションとその評価," 信学論 (D), Vol.J60-D, №10, p.887 (1977).

(42) 青江，山本，島田："右順位関係の一義化について," 信学論 (D), Vol. J60-D, №10, p.890 (1977).

(43) 山本，青江，島田："パージングテーブルによる弱順位パーザの補足," 情報処理, Vol.19, №8, p.788 (1978).

(44) 青江，山本，島田：“２組の弱順位関数の存在性，”信学論（D），Vol.J61－D，№9，p.722（1978）．

(45) 青江，山本，島田：“順位パーザの実際的構成法に対する補そく，”信学論（D），Vol．J62-D，№9，p.599（昭54－09）．

(46) 山本，青江，島田：“パージングオートマトンによる弱順位パーザ，”徳島大学工学部研究報告，Vol.22，p.124（1977）．

(47) J.Aoe，Y.Yamamoto and R.Shimada：“A technique for generating optimal parsers for precedence grammars，” Bulletin of Faculty of Engineering，Tokushima Uniuersity，Vol.15，p.39（1978）．

(48) 青江，山本，島田：“順位パーザの実際的構成法，”信学会オートマトンと言語研資，AL 78－9，p.77（1978）．

(49) 青江，山本，島田：“順位パーザのエラー回復，”信学会オートマトンと言語研資，ＡＬ79－10，p.83（1979）．

(50) 青江，山本：“パージングテーブルによる弱順位パーザの構成法について（その１），”電気関係学会四国支部連合大会，13－17，p.217（1976）．

(51) 青江，山本：“パージングテーブルによる弱順位パーザの構成法について（その２），”電気関係学会四国支部連合大会，13－18，p.219（1976）．

(52) 青江，山本，島田：“順位パーザの実際的構成法（その１），”電気関係学会四国支部連合大会，13－4，p.195（1978）．

(53) 青江，山本，島田：“順位パーザの実際的構成法（その２），”電気関係学会四国支部連合大会，13－5，p.197（1978）．

(54) 福岡，青江，山本，島田：“スパース行列の縮小化，”電気関係学会四国支部連合大会，1－6，p.11（1979）．

(55) D.E.Knuth：“On the translation of languages from left to right，” Information and Control，Vol.8，p.607（1965）．

(56) A.J.Korenjak：“A practical method for constructing LR (k) processors，” C.ACM，Vol.12，№11，p.613 (1969)．

(57) D.Parger：“A solution to an open problem by Knuth，” Information and Control，Vol.17，№5，p.462（1970）．

(58) F.L.Deremer：“Simple LR(k) grammars，” C.ACM，Vol.14，№7，p.453（1971）．

(59) W.R.Lalonde, E.S.Lee and J.J.Horning: "An LALR (k) parser generator," Information Processing 71, p.513 (1972).

(60) A.V.Aho and J.D.Ullman: "Optimization of LR (k) parsers," J.CSS, Vol.6, p.573 (1972).

(61) A.V.Aho and J.D.Ullman: "A technique for speeding up LR (k) parsers," SIAM J. Computing, Vol.2, No.2, p.102 (1972).

(62) T.Anderson, J.Eve and J.J.Horning: "Efficient LR (1) parsers," Acta Infarmatica, Vol.2, p.12 (1973).

(63) K.Culik and R.Cohen: "LR-regular grammars-on extension of LR grammars," J.CSS, Vol.7, p.66 (1973).

(64) M.L.Joliat: "Practical minimization of LR (k) parser tables," Infermation Processing 74, p.376 (1974).

(65) F.L.Bauer and J.Eickel (Editor): "Compiler construction-on advanced course-," Springer-Verlag, p.85 (1974).

(66) A.V.Aho and S.C.Johnson: "LR parsing," Computing Surveys, Vol. 6, No 2, p.99 (1974).

(67) M.L.Joliat: "A simple technique for partial elimination of unit productions from LR (k) parsers," IEEE Trans. Vol. C-25, No 7 (1976).

(68) A.V.Aho and J.D.Ullman: "Determinnistic parsing of ambiguous grammars," C.ACM, Vol,18, No 8, p.441 (1975).

(69) H.B.Hunt, T.G.Symanski and J.D. Ullman: "On the complexity of LR (k) testing," C.ACM. Vol.18, No 2, p.707 (1975).

(70) P.M.Lewis, D.J.Rosenkrantz and R.E. Stearns: "Compiler design theory," Addison-Wesley (1976).

(71) M.D.Mickunas, R.L.Lancaster and V.B.Schneider: "Transforming LR (k) grammars to LR (1), SLR (1), and (1,1) bounded right-context grammars," J.ACM, Vol.23, No 3, p.511 (1976).

(72) M.D.Mickunas: "On the complete covering problem for LR (k) grammars," J.ACM, Vol.23, No 3, p.534 (1976).

(73) M.M.Geller and M.A.Harrison :"Characterestic parsing:a framework for producing compact deterministic parsers,I and II", J.CSS, Vol,14,p.263 (1977).

(74) A.V.Aho and J.D.Ullman : "Principles of compiler design", Addison-Wesley (1977).

(75) D.Parger :"The lane-trace algorithm for constructing LR (k) parsers and ways of enhancing its efficiency", Information Sciences,Vol.12,p.19 (1977).

(76) J.Cohen and M.S.Roth :"Analysis of deterministic parsing algorithms", C.ACM,Vol.21, No 6,p.448 (1978).

(77) B.A.Wichman:"Modified report on the algorithmic language ALGOL 60", Computer J.,19, No 4, p.364 (1976).

(78) 林達也:" CFG-PL 変換について," 情報処理, Vol.12, No 3,p.145 (1971).

(79) 関本彰次:" LR (1) 文法の諸性質とそれに基づく parser の構成法," 情報処理, Vol.13, No 3,p.170 (1972).

(80) 関本, 向井, 首藤:" LR (k) parser の最小化問題について," 信学論 (D), Vol.J 56-D, No 10,p.599 (193).

(81) 大倉, 打浪, 手塚:" LR (k) パーザ縮小化の一方法," 信学論 (D), Vol. J60-D, No 1,p.141 (1977).

(82) 小島, 加藤, 中田:" LR (k) -parser 生成システムとその FORTRAN コンパイラへの応用," 情報処理, Vol.15, No 2, p.93 (1974).

(83) 牧之内顕文:" LR (k) 言語アナライザのもう一つの実際的構成法-SLR (k) の拡張-," 情報処理, Vol.17, No 8,p.729 (1976).

(84) 関本, 向井, 首藤:" LR (k) パーザの一縮小法," 信学論 (D), Vol. J61-D, No 6,p.403 (1978).

(85) 青江, 山本:" GOTOグラフによるLR(1) パーザの簡単化," 信学論(D), Vol.J60-D, No 8, p.578 (1977).

(86) 青江, 山本, 島田:" 行列形式を使用したLR (k) パーザの実際的最適化," 信学論 (D), J 60-D, No 7,p.443 (1979).

(87) 青江, 秋友, 山本, 島田:" ALGOL (JIS3000) に対するGOTOグラフによ

るLR（1）パーザの構成と解析シミュレーション," Vol.J62−D, №8, p.551 （昭54−08）.

(88) 青江，山本，島田："LL（1）解析表の一構成法," 信学論（D），Vol. J61−D, №12, p.948 (1978).

(89) 青江，山本，島田："文脈自由言語に対する構文解析部の自動生成," 徳島大学工学部研究報告，Vol.23, p.145 (1978).

(90) 青江，山本，島田："GOTOグラフによるLR（1） パーザの簡単化," 信学会オートマトンと言語研資，AL77−1, p.1 (1977).

(91) 青江，山本，島田："GOTOグラフによるLR（1）パーザの拡張," 信学会オートマトンと言語研資，AL 77−2, p.11 (1977).

(92) 青江，山本，島田："LR（k）パーザに対する簡単化," 電気学会情報処理研究会資料，IP 77−45，p.1 (1977).

(93) 青江，山本，秋友，島田："LR（1）パーザ解析シミュレーション," 信学会オートマトンと言語研資，AL 78−10, p.87 (1978).

(94) 青江，山本，島田："行列形式によるLR（k）解析表の縮小法," 信学会オートマトンと言語研資，AL 79−17, p.93 (1979).

(95) 青江，山本，島田："拡張還元によるLR（k）パーザの簡単化（その1）," 電気関係学会四国支部連合大会，13−10，p.185 (1977).

(96) 青江，山本，島田："拡張還元によるLR（k）パーザの簡単化（その2）," 電気関係学会四国支部連合大会，13−11, p.187 (1977).

(97) 秋友，青江，山本，島田："LR（1）パーザの解析シミュレーション," 電気関係学会四国支部連合大会，13−6，p.199 (1978).

(98) 青江，秋友，山本，島田："行列形式のデータ構造によるLR（k）パーザの縮小法（その1）"，電気関係学会四国支部連合大会，12 − 18，p 239 (1979).

(99) 青江，秋友，山本，島田："行列形式のデータ構造によるLR（k）パーザの縮小法（その2）," 電気関係学会四国支部連合大会，12 − 19，p.241 (1979).

(100) 青江，山本，島田："縮小された順位パーザのエラー回復とシミュレーション," 信学論（D），Vol.J62−D, №10, p.681 （昭54 − 10）.

(101) 秋友，青江，山本，島田：“エラー検出遅れのないＬＲ（ｋ）パーザのエラ訂正，”信学会オートマトンと言語研資，AL 79－18，p.103 (1979)．
(102) 秋友，青江，山本，島田：“縮小されたＬＲ（１）パーザのエラー訂正Ⅰ，”電気関係学会四国支部連合大会，12－20，p.243 (1979)．
(103) 秋友，青江，山本，島田：“縮小されたＬＲ（１）パーザのエラー訂正Ⅱ，”電気関係学会四国支部連合大会，12－21，p.245 (1979)．
(104) 田中，青江，山本，島田：“フローグラフの支配関係のための簡単化，”電気関係学会四国支部連合大会，13－3，p.193 (1978)．
(105) 佐藤，青江，山本，田中：“フローグラフの直接支配点を求めるアルゴリズム，”信学会オートマトンと言語研資，AL 79－15，p.73 (1979)．
(106) 佐藤，青江，山本：“可約フローグラフの支配関係を求めるアルゴリズム（その１），”電気関係学会四国支部連合大会，１－７，p.13 (1979)．
(107) 佐藤，青江，山本，“可約フローグラフの支配関係を求めるアルゴリズム（その２），”電気関係学会四国支部連合大会，１－８，p.15 (1979)．