形 式 言 語 に 関 す る 研 究

１ ９ ６ ７ 年 １ 月

鳥　居　宏　次

形式言語に関する研究

目　次

内　容　梗　概

本論文は筆者が大阪大学大学院工学研究科博士課程（電子工学専攻）の学生として尾崎研究室において行なつた形式言語に関する研究のうち、有限状態言語間の翻訳に関する研究を第1編六章に、コンテクスト・フリー（Context－free）言語の認識と構文解析に関する研究を第2編六章にまとめたものである。

緒論および各編第1章では研究の現状、その工学上の意義、本研究の新しい諸成果について概説している。

各編第2章では定義をまとめており、第6章では各編で得られた主な結果がまとめられている。さらに全体の結論では今後に残された問題について述べている。

一般に言語を生成する規則が定められているとき、その言語は形式言語と呼ばれているが、なかでも句構造（phrase structure）を持つ言語がある程度理論的に体型づけられ、また計算機のプログラミング言語がそれに含まれることなどから種々の立場から研究されている。一般の句構造言語に対して、構造に何らかの制限を付加した、いくつかの部分族をなす言語が知られている。その代表的なものは、コンテクスト・センシティブ（Context－sensitive）言語、コンテクスト・フリー（Context－free）言語、リニア（linear）言語および有限状態言語である。本論文はこれらの言語の一側面について研究した結果である。

第1編「有限状態言語間の翻訳」では、句構造をもつ形式言語のうち、もつとも基本的な有限状態言語が取り扱われている。言語の翻訳とは、文章ごとの1対1対応であるものとして、第3章では普通の順序回路（csmと略記する）

による写像（翻訳）が、第4章では一般化された順序回路（gsmと略記する）による写像（翻訳）が取り扱われている。また第5章では第4章における種々の結果を導く際に使われる補題ならびに定理の証明が述べられている。以下各章の内容をもう少し詳説する。

第3章「csmによる写像」では

二つの有限状態言語L_A，L_Bが与えられたとき、

条件(1)　：　写像MはL_AからL_Bの上への1対1対応でM^{-1}はMの逆写像である。

条件(2)　：　MおよびM^{-1}はcsmにより実現される。

をみたす写像が存在するか否かを判定し、存在するならば一つの翻訳機を与える能率的なアルゴリズムが示される。形式言語の翻訳をオートマトン理論の立場から眺めれば、上の結果を二つの順序回路の間の関係に拡張することが出来、上の意味で翻訳可能かどうかで有限状態言語全体を類別できる。そのとき、各類を代表する状態数最小の標準言語も求められている。

csmでは入力シンボルに出力シンボルが対応するのに対し、gsmでは入力系列（空系列を含む）に出力系列（空系列を含む）が対応する。したがつてgsmは計算機の翻訳プログラムの一つの妥当なモデルと考えられている。第4章「gsmによる写像」では

二つの有限状態言語L_A，L_Bが与えられたとき、

条件(1)　：　写像MはL_AからL_Bの上への1対1対応で、M^{-1}はMの逆写像である。

条件(1')：　写像MはL_AからL_Bの上への写像である。

条件(2')：　M（およびM^{-1}）はgsmにより実現される。

の条件(1)（または(1')）、(2')をみたす写像が存在するための必要十分条件が

与えられる。またGinsburgなどによりrecursively solvable であることが示されたにすぎない、gsmによる中への写像問題で、L_A，L_B が無限個の文章を含むとき、$M(L_A) \subseteqq L_B$でかつ$M(L_A)$が無限であるようなMがつねに存在することも示されている。これらの結果は、つぎに示される標準言語に対する考察の結果として導かれている。

条件(1)および(2')をみたすgsmの存在を二つの順序回路の関係に拡張すれば、csmの場合と同様に状態数最小で構造のきわめて単純な標準言語が、能率的に求められることが示される。

第2編「CFLの認識と構文解析」では、現在のプログラミン言語は句構造言語のうち、コンテクスト・フリー言語(Context-free phrase structure language)（CFLと略記する）であることが知られているが、そのCFLについて、コンパイラなど計算機の翻訳プログラムの作製の際に重要な役割を占める言語の認識と構文解析が取り扱われている。

第3章では一般のCFLの能率的な認識アルゴリズムが、第4章では第3章で示される認識の方法にもとずいた構文解析の方法が、そして第5章ではCFLのある部分族の認識アルゴリズムが示されている。以下各章の内容をもう少し詳説する。

第3章　「CFLの認識」では、CFLの認識とCSLの認識への拡張が述べられる。ある標準のチューリング機械によつて、$T(n)$以下の操作数で言語が認識されるとき、$T(n)$-recognizable であるという認識アルゴリズムの評価の方法が、Hartmanis と Stearns により導入された。こゝでnは文章の長さ、すなわち系列のシンボルの数を表わすものとする。本章ではその概念にしたがつて、上述のnに対して、一般のＦLが2本のワーキングテープを持つオンラインのチューリング機械により、n^3-recognizable で

あることを、ノーマル文法を仲介として示している。

第４章「ＣＦＬの構文解析」では、第３章で示された認識の方法にもとずき、能率的な構文解析のアルゴリズムを示している。この方法では、あいまいさのある言語についても全てのデイリベーション（ derivation）を見出すことが可能である。

第５章「ＣＦＬのある部分族の認識」では、リニア言語、およびメタ・リニア言語をその一部として含むＣＦＬのある部分族Ｌについて、任意のＬに含まれる言語は、１本のワーキングテープを持つオンラインのチューリング機械により、Hartmanis と Stearns の意味において、n^2-recognizable であることを示している。こゝでｎは文章の長さである。

緒　　　　論

自然言語に対して、形式言語とは、その言語を生成する規則が定められているものと考えることができるが、本論文では句構造言語（Phrase structure language）を形式言語とよぶことにする。こう呼んでも妥当と考えられる一つの理由は、言語の研究の現状において知られているいくつかのモデルのうち、理論的な展開が期待できる唯一つのモデルだと思われることである。

第１編では文献(1)〜(3)として公表した有限状態言語間の翻訳に関する研究が述べられている。２つの形式言語L_A，L_Bが与えられたとき、一定の条件をみたすL_AからL_Bの上（中）への文章ごとの写像が存在するかどうかの問題が、言語の翻訳あるいは原理的な興味からGinsburgなどにより研究されている。計算機とくに翻訳プログラムのモデルとして、"普通の順序回路（csmと略記する）"および"一般化された順序回路（gsmと略記する）"による写像が妥当な一つと考えられるが、Ginsburgらにより、L_A，L_Bを有限状態言語とするとき、csmによる写像問題およびgsmによる中への写像問題が、recursively solvable であることが示された。第３章では、L_A，L_Bが有限状態言語であるとき、つぎの二つの条件

条件１　：　写像MはL_AからL_Bの上への１対１写像で、M^{-1}はMの逆写像である。

条件２　：　MおよびM^{-1}はcsmにより実現される。

をみたす写像問題が取り扱われる。また、第４章では、次の条件

条件1′　：　写像MはL_AからL_Bの上（中）への１対１写像でM^{-1}はMの逆写像である。

条件2′　：　M（およびM^{-1}）はgsmにより実現される。

をみたす写像問題について解決されていなかつたgsmによる上への写像問題、およびrecursively solvable であることが示されていたにすぎない中への写像問題などが取扱われる。

第2編ではコンテクスト・フリー(Context-free)言語の認識と構文解析のアルゴリズムを中心に、文献(4)～(6)にもとずいて述べられる。一般に形式言語の研究としては、それを識別あるいは発生するオートマトンの研究、逆に一定のオートマトンで識別できる入力系列全体の構造や性質の研究などオートマトン理論の一側面と見られる研究が考えられるが、与えられた文章が文法に適つた文章か否かを判定し、適つている場合にはその文章構造の解析を行うことは、言語の研究のうちでも基本的な問題の一つである。

有限状態言語が有限オートマトンで受理されることを始めとして、句構造の複雑さに従つて、その言語を受理するのはどのようなオートマトンであるかが、いくつかの言語について知られている。これは句構造言語を記憶の形から分類したものといえる。一方、言語の認識問題は、アルゴリズムの能率がとくに重要であることから、最近記憶のみからでなく計算時間についての議論が問題として取上げられている。たとえばHartmanis と Stearns は標準のチューリング機械を考え、入力系列の長さをnとしたとき、その機械による言語の認識について、$T(n)$ 以下の操作数で、ある言語が認識されるとき、その言語は $T(n)$-recognizable であるという概念を導入した。それは認識に必要な計算時間の面から形式言語の複雑さを分類しようとする興味ある試みである。この概念にもとずき、嵩は一般のコンテクスト・フリー言語が2本のワーキングテープを持つオンラインのチューリング機械でn^3-recognizable であることを示したが、第3章ではそれのより簡単な別証が与えられる。第4章ではコンテクスト・フリー言語の能率的な構文解析の方法が導かれ、その方法のコ

ンテクスト・センシテイブ (Context-sensitive) 言語への拡張について述べられている。第5章には、リニア言語、さらにメタ・リニア言語をその一部として含み得る或るコンテクスト・フリー言語の部分族について、1本のワーキングテープを持つオンラインのチューリング機械で、n^2-recognizable であることが示されている。

関 連 発 表 論 文

(1) 鳥居、嵩、尾崎

“有限状態言語間の翻訳についての一考察” オートマトン研究会資料

(昭和39年1月)

(2) T.Kasami , K.Torii and H.Ozaki

“Translation of finite state languages by a sequential machine”

presented at ICMCI.Tokyo (Sept. 1964)

(3) 嵩、鳥居、尾崎

“一般化された順序回路による正規集合間の写像について”

信学誌 (昭和40年7月)

(4) 鳥居、嵩、尾崎

“Context-free 言語の Recognition と Parsing の一方法”

オートマトン研究会資料 (昭和41年6月)

(5) 鳥居、尾崎

“Context-Sensitive 言語の認識に関する一考察”

電気通信学会全国大会 (昭和41年11月)

(6) 鳥居、嵩、尾崎

“Context-free 言語の認識に関する一考察” 信学誌載録決定済

第１編　有限状態言語間の翻訳

第１章　緒　　論

２つの形式言語L_A，L_Bが与えられたとき、一定の条件をみたすL_AからL_Bの上（中）への文章ごとの写像が存在するかどうかの問題が、最近研究されている。[(7)〜(10)] 計算機とくに翻訳プログラムの形式的モデルとしては、"普通の順序回路（csmと略記する）"あるいは"一般化された順序回路（gsmと略記する）"による写像が妥当な一つと考えられる。[(7)] 中への写像問題では、問題を無意味にしないためににL_A，L_Bが無限個の文章を含むとき、写像MによるL_Aの写像像$M(L_A)$もまた無限個の文章を含むものとする。[(7)] 写像Mがgsmにより実現されるものとすると、L_Aがコンテクスト・フリー（Context-free）言語（CFLを略記する）または有限状態言語（FSLと略記する）のとき、$M(L_A)$はそれぞれCFL、FSLである。[(7)] Ginsburgらにより、L_A，L_BがCFLのとき、$M(L_A)=L_B$または$M(L_A)\subseteqq L_B$をみたすcsmまたはgsmによる写像Mが存在するかどうかの決定問題は一般にrecursively unsolvable[(5)(6)]であることが示された。[(7)] したがつて当然よりせまいクラスの言語が問題となる。そのなかでもつとも基本的なものはFSLである。たとえば、任意のCFLはそれにより定まる一つのFSLと特定のもつとも単純なCFLによつて表現できることが知られている。[(1)〜(3)] FSLは自然言語などのモデルとしては一般に不充分であるが、ある観点❀から一つのFSLモデルが提案されている。[(11)] またCFLを任意の構造の深さでFSLにより近似できる。[(2)]

❀ Depth Hypothesis

言語の翻訳とは両言語間の１対１対応と考えるのがほぼ妥当であろう。しかし、単に１対１対応ならば、両言語とも無限個の系列を含むときは、文章の長さの順に、また同じ長さの文章同志は、その中で任意の順に並べてそれぞれ番号をつけ、両言語の同じ番号の文章同志を対応させて、１対１対応が得られるが、このような対応は意味がない。したがつて翻訳はさらに“何らかの性質”を保存した写像である，その“性質”として構文論、あるいは意味論に関する性質が考えられるが、ここでは全く形式的に許される写像について条件を加え、逆にその条件のもとに保存される性質が何であるかを探る立場をとろうとするのである。

本編第３章では、二つのＦＳＬ，L_A，L_Bが与えられたとき

条件(1)✿ ： 写像ＭはL_AからL_Bの上への１対１対応でM^{-1}はＭの逆写像である。

条件(2) ： ＭおよびM^{-1}はcsmにより実現される。

のもとに、また第４章では、L_A，L_BがＦＳＬのとき、

条件(1)✿ ： 写像ＭはL_AからL_Bの上への１対１対応で、M^{-1}はＭの逆写像である。

条件(1)' ： 写像ＭはL_AからL_Bの上（中）への写像である。

条件(2)' ： Ｍ（およびM^{-1}）はgsmにより実現される。

のもとに写像問題が文献(15)～(21)にもとづいて取扱われている。

✿ Ginsburg，Rose，Hibbard(7)～(10)らは写像として条件１の１対１対応を要請していない。

第 2 章　諸 定 義

本章ではいくつかの記号の説明と定義をする。

有限個のシンボルの集合をアルファベットAとするとき、Aに属するシンボルの有限長の系列※全体をA^*で表わす。一般に系列を小文字の英字で、シンボルを大文字の英字で表わすことが多い。また系列uのシンボルの数を系列uの長さといい、#(u)と書く。長さ0の系列(空系列)をΛで表わし、さらに

$$u=u_1u_2\cdots\cdots u_k\ (u_i\leqq A\ ;\ 1\leqq i\leqq k)$$

$$v=v_1v_2\cdots\cdots v_h\ (v_i\leqq A\quad 1\leqq i\leqq h)$$

とするとき

$$uv=u_1u_2\cdots\cdots u_k v_1v_2\cdots\cdots v_h$$

と書く。※※

有限オートマトンとは、次の性質をもつ、順序づけられた5組の文字(Σ, X, δ, So, F)のことである。

(1)　Σ　は空でない有限個の状態の集合

(2)　X　は空でない有限個の入力シンボルの集合

(3)　δ　は次の状態を示す関数

(4)　So　は最初の状態

(5)　F　はΣの部分集合(最終状態の集合)

ここでδは、$\Sigma\times X^*$からΣへの写像として、任意の$u.v\leqq X^*$について

$$\delta(S, uv)=\delta(\delta(S, u), v)$$

と拡張される。

※　このことをA上で作られる有限長の系列ということもある。(1)

※※　このときuvはuとvとのconcatenationといわれる。(1)

$$L = \{ u \mid \delta((S_0 , u)) \leqq F \}$$

であるような有限オートマトンA＝（Σ, X, δ, S_0, F）が存在するとき、Lを有限状態言語（FSLと略記する）という。すなわちFSL　Lは、任意の系列uが与えられた時、初期状態がS_0の有限オートマトンAにuを入力系列として入れたとき、最後の状態がFに含まれるかどうかにより、uがLに属するか否かが決定できるような言語である。なお有限状態言語については第2編第2章で別の表現による定義をする。

一般化された順序回路（generalized seguential machine , gsmと略記する）はつぎの性質をもつ、順序づけられた6組の文字（Σ, X, Δ, σ, δ, S_0）と定義する。ここでΣ, X, δ, S_0 は、前述のオートマトンの場合と同じ意味であり、

(1)　Δは、空でない有限個の出力シンボルの集合

(2)　δは、$\Sigma \times X$からΔ^*への写像（出力関数）を示す。

δが、$\Sigma \times X$をXへの写像であるとき、gsmは普通の順序回路（complete sequential machine , csmと略記する）である。

ここで、上述の二つの順序回路において、σはオートマトンの場合と同様にまた、δはつぎのように拡張される。

$$\delta(S, uv) = \delta(S, u)\,\delta(\sigma(S, u), v)$$

gsmとcsmの違いは、csmによる写像においては、系列の長さが入力側と出力側とで保存されることである。逆にいえば、gsmでは空系列Λをも含めて有限長の入力系列に対して、出力で空系列Λをも含めて、有限長の系列が対応する。

csmにおいて、出力関数δが入力シンボルに直接依存せず、状態のみに依存し、かつ出力シンボルが2種たとえば“0”、“1”のみとするとき、δのかわ

りに出力が“1”である状態の集合をFと指定すれば、あきらかに、csmは前述の有限オートマトンになる。

定義から、有限状態言語Lは有限オートマトンを指定することにより定まる。さらにオートマトンは、状態図Gが与えられれば定まる。以下の議論では、状態図Gは簡約形であるものとし、決定性(deterministic)オートマトンを考える。したがつて一つの状態から他の状態に移るときの入力シンボルは互いにことならねばならない。

第 3 章　csmによる写像

§ 3.1 序　　言

Ginsburg , Hibbard[7] により、2つのFSL L_A , L_B が与えられたとき、翻訳機の一つのモデルとしてのcsmによる写像が存在するかどうかの決定問題はrecursively solvableであることが示されていたにすぎない。本章ではこの問題に対して、文章ごとに1対1対応があり、かつ、L_A(L_B)からL_B(L_A)の上への写像および逆写像がcsmにより実現されるための必要十分条件を求め、その結果を利用して、標準言語や翻訳機などを求める。

§ 3.2 準　　備

FSL L_A , L_B に対応するオートマトンを

$$A_A=(\Sigma_A,\ X_A,\ \sigma_A,\ s_{Ao},\ F_A)$$

$$A_B=(\Sigma_B,\ X_B,\ \sigma_B,\ s_{Bo},\ F_B)$$

とする。これらに対応する状態図は簡約化[4]されているという仮定から、任意の系列$u_A \in X_A^*$($u_B \in X_B^*$)に対しても、

$$\sigma_A(s_{Ai},\ u_A) \in F_A$$

$$(\sigma_B(s_{Bi},\ u_B) \in F_B)$$

である $s_{Ai} \in \Sigma_A$($s_{Bi} \in \Sigma_B$) はたかだか1つしか存在しない。存在するとき、これをs_{AD}(s_{BD}) とし、これ以外のΣ_A , Σ_B の状態を

$$\tilde{\Sigma}_A = \{s_{Ao},\ \cdots s_{Ai},\ \cdots,\ s_{An}\}$$

$$\tilde{\Sigma}_B = \{s_{Bo},\ \cdots s_{Bi},\ \cdots,\ s_{Bm}\}$$

とおく。

$$\sigma_A(s_{Ai}, x) = s_{Aj}, x\ X_A, 0 \leqq i, j \leqq n$$

をみたすシンボルの集まりを $X_{ij}^{(A)}$，それに含まれるシンボルの数を $P_{ij}^{(A)}$ とかき、また

$$\sum_{0 \leqq j \leqq n} P_{ij}^{(A)} = P_j^{(A)}$$

$$X_i^{(A)} = \bigcup_{0 \leqq j \leqq n} X_{ij}^{(A)}$$

とおく。同様に、$X_{ij}^{(B)}$，$X_i^{(B)}$，$P_{ij}^{(B)}$，$P_i^{(B)}$ を定義する。

§ 3.3 基本的な関係

以下、L_A から L_B への条件(1)、(2)をみたす翻訳について考える。L_A から L_B の上への翻訳機を

$$M_T = (\Sigma_T, X_A, X_B, \sigma_T, \delta_T, s_{T0})$$

とする。定義から、$u \in L_A$ のとき

$$\delta_T(s_{T0}, u) \in L_B$$

この節では、このような M_T の存在を仮定し、いくつかの必要条件を導く。

あとの取扱いの便宜上、翻訳機 M_T の代わりに、M_T と同じ出力シンボルをだすと同時に、入力系列 u が L_A に属するか否かにより 1，0 を合わせて指示するような翻訳機 M を考える。こゝで、M は $M_T \times M_A$ に等価な csm であることは容易にわかる。ただし、

$$M_A = (\Sigma_A, X_A, \{0, 1\}, \sigma_A, \delta_A, s_{A0})$$

こゝで

$$\sigma_A(s_{Ai}, x) \in F_A, x \in X_A,\ \text{のとき}\ \delta_A(s_{Ai}, x) = 1$$

他のすべてのとき　　　　　　　　　　　$\delta_A(s_{Ai}, x)=0$

である。したがつて、１つのM_Tが存在すれば、１つのＭが求まる。逆に１つの

$$M=(\Sigma, X_A, X_B\times\{0,1\}, \sigma, \delta(=(\delta_1,\delta_2)), s_0)$$

（たゞし、δ_1, δ_2は、夫々X_B，{０，１}を出力シンボルの集合とする。したがつてδ_1とδ_Tとは等しい。）

が存在して、上記の条件を満足すれば、出力成分のうち、δ_2を無視し、（状態を簡約化して）１つのM_Tが求まる。したがつて、M_Tの代わりにＭを考えても一般性を失なわない。以下Ｍを考えていく。また、以下

$$\sigma_A(s_{A0}, u)=\sigma_A(u)$$

$$\sigma_B(s_{B0}, u)=\sigma_B(u)$$

$$\sigma(s_0, u)=\sigma(u)$$

$$\delta_1(s_0, u)=\delta_1(u)$$

とかき、A_A，A_Bの状態s_{Ai}，s_{Bi}への入力系列の集合を

$$U(s_{Ai})=\{u/\sigma_A(u)=s_{Ai}\}$$

$$V(s_{Bi})=\{u/\sigma_B(u)=s_{Bi}\}$$

で示し、Ｍの状態s_iへの入力系列、及び出力系列を

$$U(s_i)=\{u/\sigma(u)=s_i\}$$

$$V(s_i)=\{\delta_1(u)/\sigma(u)=s_i\}$$

で示す。

補題３.１：任意の$0\leqq i\leqq N$に対して

$$U(s_i)\subsetneqq U(s_{Aj}) \tag{1}$$

をみたす$0\leqq j \quad n$が存在する。

（証明）$u_1, u_2 \in U(s_i)$とする。任意の系列v（$\in A$）に対して

$$\delta_2(u_1 v)=\delta_2(u_1)\delta_2(S_i,v)$$
$$\delta_2(u_2 v)=\delta_2(u_2)\delta_2(S_i,v)$$
したがつて、$\delta_2(u_1 v)$ と $\delta_2(u_2 v)$ の最終状態は等しい。すなわち
$$u_1 v\in L_A \rightleftarrows u_2 v\in L_A$$
A_A が簡約化されているという仮定から、$\sigma_A(u_1)=\sigma_A(u_2)$ であり、これを S_{Aj} とおくと、$u_1, u_2\in U(S_{Aj})$。

$U(S_{A0}), U(S_{A1}), \cdots\cdot$ は互に共通な系列を含まないから、補題 3.1 より $i(0\leqq i\leqq N)$ に対して、(1)をみたす j が一意にきまる。そこで、
$$j=\alpha(i)$$
とおく。
$$U(S_i)\subseteqq U(S_{AD})$$
をみたすような S_i を Σ から除いた残りを $\widetilde{\Sigma}$ とおき、適当に番号をつけかえて
$$\widetilde{\Sigma}=\{S_0, S_1, \cdots S_i, \cdots, S_N\}$$
とおく。$\Sigma-\widetilde{\Sigma}$ の状態は L_A と L_B の翻訳には直接関係がない。たとえば、そのあとの入力如何にかゝわらず、入力が L_A でないとわかつたとき、それを指示する特別のシンボルを出すようなときを除いて、組み合わせ禁止 (don't care) の場合にあたる。したがつて、以下の議論で関係のあるのは $\widetilde{\Sigma}$ に含まれる状態である。

補題 3.2 : 任意の $i(0\leqq i\leqq N)$, $x_1, x_2\in X_i^{(A)}$ $x_1\neq x_2$ について
$$\delta_1(S_i, x_1)\neq\delta_1(S_i, x_2)$$
(証明) $u\in U(S_i)$ とする。$x_1, x_2\in X_i^{(A)}$ であるから、$u x_1 v_1\in L_A$, $u x_2 v_2\in L_A$ である $v_1, v_2\in X_A^*$ が存在する。いま $\delta_1(S_i, x_1)=\delta_1(S_i, x_2)=y$ とすると、

$$w_1 = \delta_1(u)\,y\,\delta_1(\sigma(S_1, x_1), v_1) \in L_B$$

$$w_2 = \delta_1(u)\,y\,\delta_1(\sigma(S_1, x_2), v_2) \in L_B$$

逆の写像において、w_1, w_2に対して、ux_1v_1, ux_2v_2が対応しなければならない。逆写像のcsmの出力関数をδ_Iとすると、csmは長さを保存するから

$$ux_1 = \delta_I(\delta_1(u)\,y)$$

$$ux_2 = \delta_I(\delta_1(u)\,y)$$

したがつて、$x_1 = x_2$となり仮定に反する。

補題 3.3：$\sigma(u_1), \sigma(u_2) \in \widetilde{\Sigma}, \sigma_1(u_1) = \sigma_1(u_2)$ ならば $u_1 = u_2$である。

補題 3.2 より明らかである。

補題 3.4：任意の $i\,(0 \leqq i \leqq N)$ に対して

$$V(S_i) \subseteqq V(S_{Bj}) \tag{2}$$

をみたす $0 \leqq j \leqq m$ が存在する。

（証明） $u \in L_A \rightleftarrows \delta_1(u) \in L_B$ である。

$u_1, u_2 \in U(S_i)$ とする。$\delta_1(u_1)w \in L_B$ である任意のwに対して写像が全射であることから、$u_3, v \in X_A^*$ が存在して、$\delta_1(u_3v) = \delta_1(u_1)w$, $\#(u_3) = \#(u_1)$, $u_3v \in L_A$となる。補題 3.3 より、$u_3 = u_1$。したがつて、$w = \delta_1(S_i, v)$。ゆえに、$\delta_1(u_2v) = \delta_1(u_2)\delta_1(S_i, v) = \delta_1(u_2)w$。補題 3.1 と$u_1v \in L_A$ より、$u_2v \in L_A$ したがつて、$\delta_1(u_2)w = L_B$。A_B が簡約化されているという仮定から、$\sigma_B(\delta_1(u_1)) = \sigma_B(\delta_1(u_2))$。これを$S_{Bj}$ とおくと、jは(2)をみたす。

補題 3.4 より $i\,(0 \leqq i \leqq n)$ のとき、(2)をみたす $0 \leqq j \leqq m$が一意にきま

るから、これを

$$j = \beta(i)$$

とおく。$u \in L_A$ なら、$\delta_1(u) \in L_B$ ゆえ、$s_{\alpha(i)} \in F_A$ ならば、$s_{\beta(i)} \in F_B$ であり、逆もなりたつ。

補題 3.5：$w_1 w_2 w_3 \subset L_B, w_2 \neq \Lambda, \sigma_B(w_1) = s_{Bj}$ とする。このとき、$j = \beta(i)$ であるような任意の s_i について、$\delta_1(s_i, u) = w_2, u \in X_A^*$ をみたす u が存在する。

（証明）$j = \beta(i)$ であるから、$\sigma_B(\delta_1(u_1)) = s_{Bj}, \sigma(u_1) = s_i$ であるような $u_1 \in X_A^*$ が存在する。仮定より、$\delta_1(u_1) w_2 w_3 \in L_B$ で、写像が全射であることと、補題 3.3 より

$$\delta_1(u_1 u_2) = \delta_1(u_1)\delta_1(s_i, u_2) = \delta_1(u_1) w_2 w_3$$

をみたす $u_2 \in X_A^*$ が存在する。csm の長さ保存性より、長さ $\#(w_2)$ の u_2 の部分系列を u_3 とすると、$\delta_1(s_i, u_3) = w_2$ である。

補題 3.6：任意の $i\ (0 \leqq i \leqq N)$ に対して、

$$y = \delta_1(s_i, x), \quad x \in X^{(A)}_{\alpha(i)}, \quad y \in X^{(B)}_{\beta(i)}$$

により、$X^{(A)}_{\alpha(i)}$ と $X^{(B)}_{\beta(i)}$ が 1 対 1、かつ写像は全射である。

（証明）任意の $y \in X^{(B)}_{\beta(i)}$ に対して、$w_1 y w_3 \in L_B$ であるような w_1, w_3 が存在する。$\alpha(i), \beta(i)$ の定義と補題 3.5 より、$y = \delta_1(s_i, x), x \in X^{(A)}_{\alpha(i)}$ であるような x が存在する。したがつて、補題 3.2 より、補題 3.6 の成立がわかる。

補題 3.6 より、δ_1 の逆関数が $X^{(B)}_{\beta(i)}$ の上で定義できる。それ以外では任意の X_A のシンボルと定義して、これを δ_1^{-1} とかく。

$$M^{-1} = (\Sigma, X_B, X_A, (\lambda_1^{-1}(s_i, x)), \delta_1^{-1}, s_0)$$

とおくと、M^{-1} は L_B から L_A への逆写像を与える。すなわち、任意の

$v \in L_B$ に対して、$u = \delta_1^{-1}(v) \in L_A$ ， $v = \delta_1(u) = M(u)$ である。したがつて、次の補題3.7が導かれる。

補題3.7： $v \in L_B$ のとき、$\delta_1^{-1}(v) = M^{-1}(v)$

つぎに、$\alpha(i_1)=\alpha(i_2)$， $\beta(i_1)=\beta(i_2)$， $i_1 \neq i_2$， $i_2 \neq 0$ のような i_1, i_2 が存在する場合を考える。

つぎのような M' による写像を定義する。

$$M' = (\Sigma - \{S_{i2}\}, X_A, X_B \times \{0, 1\}, \sigma', \delta', S_0)$$

ただし、 $\delta' = (\delta'_1, \delta'_2)$

また、$\sigma(S_i, x) = S_j, j \neq i_2$ のとき $\sigma'(S_i, x) = S_j$

$\sigma(S_i, x) = S_{i2}$ のとき、$\sigma'(S_i, x) = S_{i1}$

とする。さらに、$\delta'(S_i, x) = \delta_1(S_i, x)(i \neq i_2)$ と定義する。

補題3.8： M' は L_A ，L_B 間の条件(1)、(2)をみたす翻訳機である。

（証明） 先づ、写像が全射であることを証明する。任意の $w \in L_B$ に対して、$\delta_1(u_0) = w$ なる $u_0 \in L_A$ が存在する。u を入力系列とするとき、Mのたどる状態の系列に S_{i2} が含まれなければ、$\delta'_1(u_0) = w$。含まれるとき、u_0 の最初の部分系列※ u_1 のうち

$$\sigma(u_1) = S_{i2}$$

となる最小長のものを u_1 とする。$u_0 = u_1 v_1$ とおくと、あきらかに、$\delta_1(u_1) = \delta'_1(u_1)$。これを w_1 とおく。

$$w = \delta_1(u_1)\delta_1(S_{i2}, v_1) \tag{3}$$

これを $w_1 w'_1$ とおく。

$\beta(i_1) = \beta(i_2)$ であるから、補題3.5より

$$w'_1 = \delta_1(S_{i2}, v_1) = \delta_1(S_{i1}, v'_1) \quad , \quad v'_1 \in X_A^*$$

※ initial substring

であるようなv'_1が存在する。

ところが$u \in U(S_{i_1})$とすると、$\beta(i_1)=\beta(i_2)$より

$$\sigma_B(\delta_1(u))=\sigma_B(\delta_1(u_1))$$

となり、(3)より

$$\delta_1(u)\delta_1(S_{i_2},v_1)=\delta_1(u)\delta_1(S_{i_1},v'_1)=\delta_1(uv'_1)\in L_B$$

したがつて、$uv'_1 \in L_A$ である。

$\alpha(i_1)=\alpha(i_2)$ より、$\sigma_A(u)=\sigma_A(u_1)$である。したがつて、$u_1v'_1 \in L_A$。

状態S_{i_1}から、入力系列v'_1をMに入れるとき、Mの状態の系列にS_{i_2}が含まれなければ、$\delta'_1(u_1v'_1)=w$。もし含まれるとき、v'_1の最初の部分系列u_2のうち、

$$\sigma(S_{i_2},u_2)=S_{i_2}$$

となる最小長のものをu_2とする。$u_0=u_1u_2v_2$とおくと、

$$\delta'_1(u_1u_2)=w_1w_2$$

ただし、w_2はw'_1のうち空でない部分系列である。

以下同様にくりかえして、

$$\delta'_1(u_1u_2\cdots\cdots)=w_1w_2\cdots\quad=w$$

となることがわかる。

つぎに、1対1対応をなすことを証明する。

$\delta'_1(u_1)=\delta'_1(u_2)=w$, $w\in L_B$, $u_1 \neq u_2$, $u_1u_2 \in L_A$とする。

また、$u_1=u_0x_1u_3$, $u_2=u_0x_2u_4$, $x_1 \neq x_2$, $x_1, x_2 \in X_A$

とおく。すると容易に、$\sigma'_1(u_0)\in\tilde{\Sigma}$がわかる。したがつて、$\delta'_1$の定義と補題3.2より

$\delta'_1(\sigma(u_0),x_1)\neq\delta'_1(\sigma(u_0),x_2)$となる。これはあきらかに矛

盾である。

また、補題3.7より、逆写像を与えるcsmが存在することもわかる。

つぎに、M′についても同様に$\alpha(i)$，$\beta(i)$を定義すれば、任意の$i \neq i_2$についてMのα，βと等しいことが容易にわかる。したがつて、(α，β)が等しい状態が存在すれば、順次同じ変形を行なつて、つぎの補題3.9の成立がわかる。

補題3.9：条件(1)，(2)をみたす翻訳機が存在すれば、任意の$i_1 \neq i_2$ ($0 \leqq i_1, i_2 \leqq N$)に対して

$$(\alpha(i_1), \beta(i_1)) \neq (\alpha(i_2), \beta(i_2)) \qquad (4)$$

であるような条件(1)，(2)をみたす翻訳機が存在する。

したがつて、(4)が成立するような機械を考えれば充分であるから、以下Mがはじめから(4)をみたしているとする。Mの状態の数は高々$(n+1)(m+1)$である。$\widetilde{\Sigma}$の状態をs_iの代りに$(\alpha(i), \beta(i)) = (j, k)$で表わすことにする。すると、つぎの補題が導かれる。

補題3.10：任意の$(j, k) \in \widetilde{\Sigma}$に対して、

$$y = \delta((j, k), x) \quad , \quad x \in X_j^{(A)}$$

ならば

$$(j', k') = \sigma((j, k), x) \qquad (5)$$

ただし　$s_{Aj'} = \sigma_A(s_{Aj}, x)$

$$s_{Bk'} = \sigma_B(s_{Bk}, x)$$

(5)をみたすシンボルxの数を$P_{jk, j'k'}$とすれば、補題3.6および(4)より、つぎの補題が導かれる。

補題3.11：任意の$(j, k) \in \widetilde{\Sigma}$，$0 \leqq j' \leqq n$，$0 \leqq k' \leqq m$について

$$\sum_{\{\ell \mid (j', \ell) \in \Sigma\}} P_{jk, j'\ell} = P_{jj'}^{(A)} \tag{6}$$

$$\sum_{\{\ell \mid (\ell, k') \in \Sigma\}} P_{jk, \ell k'} = P_{kk'}^{(B)} \tag{7}$$

§ 3.4 翻訳可能の条件

以上よりつぎの定理が得られる。

定理 3.1 ： つぎの条件をみたす（n+1)(m+1)個の組（j，k）（o≦j≦n，o≦k≦m）の部分集合$\widetilde{\Sigma}$

(i) （o，o）$\in \widetilde{\Sigma}$

(ii) （j，k）$\in \widetilde{\Sigma}$ ならば

$$s_{Aj} \in F_A \rightleftarrows s_{Bk} \in F_B$$

(iii) 任意の o≦j≦n について、（j，k）$\in \widetilde{\Sigma}$であるkが少くとも1つ存在する。

(iv) 任意の o≦k≦m について、（j，k）$\in \widetilde{\Sigma}$である j が少くとも1つ存在する。

(v) (6)、(7)をみたす非負整数の組$P_{jk, j'k'}$（（j，k），（j′，k'）$\in \widetilde{\Sigma}$ が存在する。

が存在することが、条件(1)、(2)をみたすL_AとL_Bの翻訳機csmが存在するための必要充分条件である。

（証明） 必要なことはすでに示したので、充分なことを証明する。そのためには、実際に1つのMが存在することを示せばよい。

1） $\Sigma = \widetilde{\Sigma} \cup s_D$，$s_0 =$（0，0）とおく。

2） 各（j，k）$\in \widetilde{\Sigma}$について、$X_{jj'}^{(A)}$（$X_{kk'}^{(B)}$） を互いに共通な要素をも

たないで、$P_{jk,j'\ell}$ ($P_{jk,\ell k'}$) 個のシンボルよりなる部分集合 $X^{(A)}_{jk,j'\ell}$ ($\ell \mid (j',\ell) \in \widetilde{\Sigma}$) ($X^{(B)}_{jk,\ell k'}$ ($\ell \mid (\ell,k') \in \widetilde{\Sigma}$)) に任意に分割する。条件(V)よりこれは可能である。

あきらかに $X^{(A)}_{jk,j'k'}$ ($X_{jk,j'k'}$) ($(j,k),(j',k') \in \widetilde{\Sigma}$) は互いに共通な要素をもたず、かつ

$$X^{(A)}_j = \bigcup_{(j',k') \in \widetilde{\Sigma}} X^{(A)}_{jk,j'k'}$$

$$\left(X^{(B)}_k = \bigcup_{(j',k') \in \widetilde{\Sigma}} X^{(B)}_{jk,j'k'} \right)$$

3) 各 $(j,k) \in \widetilde{\Sigma}$ について、δ をつぎのように定義する。

$$\sigma((j,k),x) = (j',k'), \quad x \in X^{(A)}_{jk,j'k'}$$

$$\sigma((j,k),x) = S_D \qquad , x \in X^{(A)}_j$$

$$\sigma(S_D,x) = S_D \qquad , \text{(全ての } x \text{ について)}$$

こゝで、(0, 0) から到着できない状態は除くものとする。

4) 各 $(j,k) \in \widetilde{\Sigma}$ につき $X^{(A)}_{jk,j'k'}$ と $X^{(B)}_{jk,j'k'}$ ($(j',k') \in \widetilde{\Sigma}$) を任意に1対1に対応づける。両者の含むシンボルの数が同数ゆえ、これは可能である。これは全体として $X^{(A)}_j$ と $X^{(B)}_k$ の1対1対応となる。この対応を $\delta_1((j,k),x)$ とする。さらに x $X^{(A)}_j$ に対しては任意に X_B のシンボルを定義する。また、$\delta_1(S_D,x)$ を X_B の任意のシンボルで定義する。

このように定義されたMが実際に課せられた条件をみたすことはつぎのようにわかる。

長さ r のすべての X_A のシンボルより成る系列の集りを $\theta_r(X_A)$ とかく。各 $(j,k) \in \widetilde{\Sigma}$ について

$$U_{j,k} = \{ u \mid \delta(u) = (j,k) \}$$

$$V_{j,k} = \{ \delta_1(u) / \sigma(u) = (j,k) \}$$

とおく。rについての帰納法により容易に

$$\bigcup_{\{k \mid (j,k) \in \tilde{\Sigma}\}} U_{j,k} = U(S_{Aj})$$

$$\bigcup_{\{j \mid (j,k) \in \tilde{\Sigma}\}} V_{j,k} = V(S_{Bk})$$

§ 3.5 標　準　形

条件(1)、(2)、をみたす翻訳機によつて相互に翻訳できるL_A ，L_B を$L_A \approx L_B$ とかくと、関係"≈"によつて有限状態言語全体を類別することができる。このとき、同じ類に属する有限状態言語は、本質的に類似した性質をもつていると考えられる。各類の代表として、一定の標準言語を選ぶことは意味があろう。標準言語L_S として、状態数最小のものを選ぶことにする。与えられた言語をL_A 、その状態図$G(L_A)$とするとき、その標準形はつぎのように求めることができる。

(i)　$G(L_A)$の状態の集合$\{S_{Aj}\}$を最終状態か否かで二つに分類する。それを

$$J_0^{(0)} = \{ j / S_{Aj} \in F_A \}$$

$$J_1^{(0)} = \{ j / S_{Aj} \notin F_A \}$$

ここでF_A は最終状態の集合を表わす。このとき

$$\Delta_A^{(0)} = \{ J_0^{(0)}, J_1^{(0)} \}$$

とする。

(ii)　$\Delta_A^{(r)} = \{ J_0^{(r)}, \cdots\cdots, J_{q_r}^{(r)} \}$ が与えられたとし、

$$\rho_j^{(A)} = \Bigl(\sum_{j' \in J_0^{(r)}} P_{jj'}^{(A)}, \cdots\cdots, \sum_{j' \in J_{q_r}^{(r)}} P_{jj'}^{(A)} \Bigr)$$

とする。

(iii) 各 J_i（$0 \leqq i \leqq q_r$）の要素について、ことなる $\rho_j^{(A)}$ を大きさの順に並べる。[*] それを

$$\rho_{j_0}^{(A)} > \rho_{j_1}^{(A)} > \cdots > \rho_{j_r}^{(A)}$$

とする。$\rho_j^{(A)}$ が $\rho_{j\ell}^{(A)}$ に等しいものをまとめて

$$J_{i\ell}^{(r)} \qquad 0 \leqq \ell \leqq r_i$$

とおき

$$\Delta_A^{(r+1)} = \{ J_{00}^{(r)}, J_{01}^{(r)}, \cdots J_{10}^{(r)}, J_{11}^{(r)}, \cdots J_{q_r 0}^{(r)} \cdots J_{q_r r_{q_r}}^{(r)} \}$$

とおく。もし $\Delta_A^{(r+1)} \neq \Delta_A^{(r)}$ ならば、$r+1 \longrightarrow r$ として、(iii) をくり返す。$\Delta_A^{(r+1)} = \Delta_A^{(r)}$ ならば、$\Delta_A^{(r)}$ をかき直して

$$\Delta_A^{(r)} = \{ J_0^{(r)}, J_1^{(r)}, \cdots, J_{q(A)}^{(r)} \}$$

とおく。

このとき標準言語の状態図を状態図とするオートマトン $A_S = (\Sigma_S, X_{S'}, \sigma_{S'}, S_t, F_S)$ はつぎのように書ける。

$$\Sigma_S = \{ S_0, S_1, \cdots, S_{q_r}, S_D \}$$

$$F_S = \{ S_i \mid J_i^{(r)} \subseteq J_1^{(0)} \}$$

かつ、t は $0 \subseteq J_t^{(r)}$ をみたすものとし、X_{ih}, $X_{i'h'}$（$(i, h) \neq (i', h')$ のとき）は互いに共通の要素をもたないとする。

$$\delta_S(S_i, x) = S_h \quad , \qquad x \in X_{ih}$$

$$\delta_S(S_i, x) = S_D \quad , \qquad x \notin X_{ih}$$

また、すべての x について

---

[*] 普通行われているように実数の n-tuple を順序づける。$(a_1, a_2 \cdots a_i \cdots) > (b_1, b_2, \cdots b_i, \cdots)$ は $a_j = b_j$（$1 \leqq j < i$）で、且つ $a_i > b_i$ のとき、またそのときに限る。

$$\delta_S(S_D, x) = S_D$$

とすると、A_S はあきらかに簡約化された形である。A_S に対応する言語を L_S とすると、その作りかたからわかるように、L_S は S_D（組合せ禁止の状態）を除外すれば、L_A と"≈"の関係で結ばれる言語のうち、状態数最小である。また、前節からわかるように、$L_S \approx L_A$，L_S は、S_D と入力シンボルのえらび方を除いて一意的にきまる。

$\sum_{h=0}^{q_r} \mu_{ih}^{(A)}$ が、i に依存しないときのみ S_D を含まないものが存在し得る。また、簡約形に反しない範囲で、入力シンボルの数をへらすことができる。

以上の標準言語およびそれに対する状態図の定義よりつぎの補題3.12は明らかである。

補題3.12：任意の $G(L_A)$ に対して、その標準形 $S(G_A)$ は、一意的に定められる。

補題3.13：$J_0, J_1, \cdots J_q$ ； $K_0, K_1, \cdots K_q$ をそれぞれ整数の組 $\{j / 0 \leqq j \leqq n\}$，$\{k / 0 \leqq k \leqq m\}$ の分割とする。[❀]（すべての J_i，K_i が空でないとする）

$$\rho_j^{(A)} = \left(\sum_{j' \in J_0} P_{jj'}^{(A)}, \cdots, \sum_{j' \in J_q} P_{jj'}^{(A)} \right)$$

$$\rho_k^{(B)} = \left(\sum_{k' \in K_0} P_{kk'}^{(B)}, \cdots, \sum_{k' \in K_q} P_{kk'}^{(B)} \right)$$

とおく。もしMが存在して

$$\tilde{\Sigma} \quad \bigcup_{i=0}^{q} J_i \times K_i \text{[❀❀]}$$

ならば、任意の $(j, k) \in \tilde{\Sigma}$ について

❀ $J = \bigcup_i J_i$，$J_i \cap J_j = 0$，$(i \neq j)$、かつ全ての $J_i \neq 0$ のとき $\{J_i\}$ を J の分割とよぶ。

❀❀ $\tilde{\Sigma}$ に属する任意の組 (j, k) に対して、$j \quad J_i$，$k \quad K_\theta$ ならば $i = \ell$ ということ。

$$\rho_j^{(A)} = \rho_k^{(B)} \tag{8}$$

また、任意の $j \in J_j$ に対して、

$$\rho_j^{(A)} = \rho_k^{(B)} \quad , \qquad k \in K_i$$

である k が少くとも1つ存在する。任意の $k \in K_i$ についても同様なことが成立する。

（証明） (8)は(6)、(7)よりあきらかである。また後半は定理3.1の条件(iii)(iv)と(8)よりわかる。

定理3.2　$L_A \approx L_B$ の必要充分条件は $S(G_A) = S(G_B)$ である。

（証明） $L_A \approx L_B$ とする。したがつて定理3.1をみたすような翻訳機Mが存在する。任意の $(j, k) \in \tilde{\Sigma}$ について、$0 \leqq j \leqq n$，$0 \leqq k \leqq m$ とすると、(6)、(7)をみたすことから

$$P_j^{(A)} = \left(\sum_{j' \in J_0} P_{jj'}^{(A)}, \cdots \cdots, \sum_{j' \in J_{p_1}} P_{jj'}^{(A)} \right)$$

$$\rho_k^{(B)} = \left(\sum_{k' \in K_0} P_{kk'}^{(B)}, \cdots \cdots \sum_{k' \in K_{p_2}} P_{kk'}^{(B)} \right)$$

が作れる。ここで $J_0, J_1, \cdots, J_{p_1}$ は整数の組 $\{ j / 0 \leqq j \leqq n \}$ の分割であり、$K_0, K_1, \cdots, K_{p_2}$ は整数の組 $\{ k / 0 \leqq k \leqq m \}$ の分割である。

もし、$j, j' \in J_i$，$\rho_j^{(A)} \neq \rho_{j'}^{(A)}$ であるような j, j' が存在すれば、$\rho_i^{(A)}$ の等しい J_i の要素をまとめて

$$J_{i0}, J_{i1}, \cdots, J_{ir}$$

と J_i を細分割する。K_i についても同様に細分割すると、同数の

$$K_{i_0}, K_{i_1}, \cdots, K_{i_r}$$

が得られ、番号を適当につければ、任意の $j \in J_{i_\ell}$，$k \in K_{i_\ell}$ について補題3.12より

$$\rho_j^{(A)} = \rho_k^{(B)}$$

これと補題3.13より$S(G_A)=S(G_B)$ である。

充分条件は標準形の定義から明らかである。

§3.6 翻訳機を求めるアルゴリズム

任意に与えられた二つの有限状態言語L_A，L_B間で翻訳可能のとき、その翻訳機を求める能率的なアルゴリズムを示す。

前節の議論より、L_A，L_Bについてそれぞれ独立に、標準言語を求め、それらが一致しないならば、翻訳機は存在しないことがわかる。もし一致するならば、

$$\Delta_A^{(r)}=\Delta_A^{(r+1)}=\{J_{00}^{(r)},J_{01}^{(r)},\cdots\quad\cdots J_{10}^{(r)},J_{11}^{(r)},\cdots\\ J_{q_r 0}^{(r)}\cdots\quad\cdots J_{q_r\gamma_{q_r}}^{(r)}\}$$

$$\Delta_B^{(r)}=\Delta_B^{(r+1)}=\{K_{00},K_{01},\cdots\quad\cdots K_{10},K_{11},\cdots\\ K_{q_r 0}^{(r)}\cdots\quad\cdots K_{q_r\gamma_{q_r}}^{(r)}\}$$

であるとして、翻訳機はつぎのように求めることができる。

$\tilde{\Sigma}=\bigcup_{i=0}^{q_r} J_i^{(r)}\times K_i^{(r)}$ とおく。(6)、(7)は$(1+q_r)$個の独立な部分に分離され、各部分で

$$\sum_{\ell\in K_i^{(r)}} P_{jk,j'\ell}=P_{jj'}^{(A)}\quad,\quad j'\in J_i^{(r)}$$

$$\sum_{\ell\in J_i^{(r)}} P_{jk,\ell k'}=P_{kk'}^{(A)}\quad,\quad k'\in K_i^{(r)}$$

を別々にとけばよい。結局つぎの形の方程式をとくことになる。

$$\sum_{j=1}^{g} Z_{ij}=C_i\qquad(1\leqq i\leqq f)$$

$$\sum_{i=1}^{f} Z_{ij}=D_j\qquad(1\leqq j\leqq g)$$

ただし、Z_{ij}，C_i，D_jは非負整数でかつ

$$\Sigma C_i = \Sigma D_j$$

$Z_{ij} \geqq 0$ であるから、$C_i = 0$、または$D_i = 0$ならば、その行、或は列を除いて考えてよいから、

$$C_i, D_j > 0$$

とする。また、$f \leqq g$とする。

$$Z_{ii}^{(0)} = \min(C_i, D_i) \qquad 1 \leqq i \leqq f$$

$$Z_{ij}^{(0)} = 0 \qquad (i \neq j)$$

とすると、Z_{ij} の代りに$Z'_{ij} = Z_{ij} - Z_{ij}^{(0)}$について考えると容易に行と列の数がfだけ少い方程式に帰着される。したがつて同じことをくり返せば1つの解が求まる。もちろん解は一般に唯一なものでない。そこで、翻訳機のうちで、状態数の少いものを求める一つの方針としては、$\sum_{j,k} P_{jk,j'k'} = 0$である$(j', k')$の数をふやすことであろう。このような$(j', k')$は実質的には$\widetilde{\Sigma}$に入らない。

§3.7 結　　言

csmによる翻訳が可能であるという関係"≈"によりFSL全体を類別し、各類を代表する状態数最小の標準言語が得られた。これはcsmが長さを保存することから、能率が等しく状態数最小のVariable length codeを求める問題などに応用できる。

第 4 章　gsmによる写像

§ 4.1　序　　　言

前章のcsmより更によい翻訳プログラムのモデルとして、csmを一般化したgsmによる写像問題を文献[13]にもとずいて本章で取り扱う。以下条件(1)′ (2)′ のもとで考えるものとする。

gsmの写像問題の中でも、二つのFSL　L_A，L_B が与えられたとき、L_A を L_B の上へ写像するgsmが存在するか否かについては、文献(9)でも未解決な問題であつたが、定理 4.2 はこの問題に対する解答を与える。また $M(L_A)=L_B$ であるようなgsmによる写像Mが存在するための簡単な必要充分条件が示され、さらに単にSolvableであることが示されていた[9]にすぎないgsmによる中への写像問題について、L_A，L_B が無限の文章を含むとき、✿ $M(L_A)\subseteqq L_B$ で $M(L_A)$ が無限であるようなMがつねに存在することも示される。これらはいくつかのより基本的な問題に対する考察の結果として導びかれる。

§ 4.2　定義と準備

条件(1′)、(2′)をみたすMが存在するとき

$$L_A \sim L_B$$

と書くことにする。また1対1の仮定のないとき、すなわち、$M(L_A)=L_B$ をみたすgsmによるMが存在するとき

$$L_A \rightarrow L_B$$

✿　この判定法は知られている。[12]

とかく。

FSL　Lに対応する有限オートマトン$A=(\Sigma, X, \sigma, S_0, F)$の簡約化された状態図Gは、$S_0$とFに属する状態が指定され、状態間の遷移には対応する入力シンボル（以下枝の入力シンボルとよぶ）が書き込まれている。これは状態を節点とし、状態間の遷移には節点間の枝を対応させた向きをもつたグラフ[13]（線型接続 linear graph）であると考えられる。したがつて入力シンボルを枝の入力シンボルとよぶことにすれば、考えているオートマトンは、決定性（deterministic）[12]オートマトン故、一つの状態から出る枝の入力シンボルは互いに異ならねばならない。

さて、一般性を失なわずに特定の" end mark " をもつ言語について考える[1]。こゝでは文章ごとの対応について考え、いくつかの文章のつながつたものには、それぞれ対応する文章のつながりを対応させるにすぎない。したがつて1つづつの文章の対応を独立に考えればよいから、以下Lの状態図Gにおいて初期状態と最終状態（最終状態は複数個でもよい）が一致しておればそれを分離し、また最終状態から出る枝をとり除いたものを考える。また初期状態から遷移する道がないか、又は最終状態に属する状態への道がないような状態は以下の議論に直接関係ないから考察から省く。L_AからL_Bへの写像を与えるgsm　Mの状態図をG_M，L_A，L_Bの簡約状態図をG_A G_Bと書くことにする。必要ならG_Mにおいて状態の分割を行つてG_M上初期状態から各状態S_iへの任意の系列uをG_Aの上で入力系列として、初期状態からたどるとき、G_Aの同じ状態（S_iのみに依存する）に到達するようにできる。以下このようになつていると仮定する。またGについてのべたと同じように、G_Mの最終状態（G_Aの最終状態に対応する状態）から出る枝をとり除くことにする。G_Mにおいて枝の名前として入力シンボルのみに

着目したグラフを入力グラフG_Iとよぶ。G_IはL_Aの一つの状態図に他ならないが、簡約形とは限らないから一般にG_Aから状態の分割を適当に行ったものである。

状態S_iに関する" prefix set " Pをつぎの条件をみたす入力シンボルの有限個の系列の集合として定義する ： (1) 任意の$p_1, p_2 \in P$に対して、p_1，p_2は一方が他方の接頭語（prefix）になることはない。(2) $\sigma(S_i, u)$ Fであるような任意のuに対して、$u = pv$，$p \in P$，$v \in X^*$をみたすpがただ一つ存在する。また逆に任意の$p \in P$には$\sigma(S_i, pv) \in F$となるような$v \in X^*$が少くとも一つ存在する。

S_iに関する" prefix set " Pによる変換をつぎのように定義する ： Lの状態図G（gsmの状態図G_M）においてS_iから出る枝をとり除き、各$p_i \in P$についてS_iから$\sigma(S_i, p_j)$へ枝を結び、これにp_j（p_j／$\delta(S_i, p_j)$）をかき込む。

このような変換を有限図くり返してグラフG'が得られたとすると、あきらかにG'によつてもLが一意的に指定される。このようなG'を拡張された（prefix形の）状態図とよぶことにする。

簡単のためG_Mの枝bがラベルとしてもつ出力系列を$\delta(b)$、枝bの終端点を$\sigma(b)$とかくことにする。G_Mの枝のラベルとして出力系列のみに着目し、始点から出発し、順次その隣接節点についてつぎの操作をそれ以上変形できなくなるまで行う。着目している節点から出る枝$\{b_i\}$について

(i) $\delta(b_i) = \delta(b_j)$ならば、$\sigma(b_i)$と$\sigma(b_j)$を併合し、(ii) $\delta(b_{i_2}) = \delta(b_{i_1})v_2, \delta(b_{i_3}) = \delta(b_{i_2})v_3, \cdots \delta(b_{i_h}) = \delta(b_{i_{h-1}})v_h$，$(v_2, \cdots, v_h \neq \Lambda)$の関係があり各$b_{i_h}$についてはprefix関係がないとき、$b_{i_h}$を図4.1のように分割する。このような

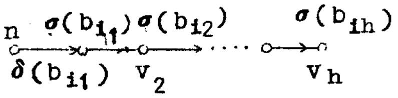


図 4.1　枝の分割

操作はもちろん有限回で終了する。得られたグラフを G_M の出力グラフ G_0 とよぶ。G_0 の初期状態、最終状態は G_M のそれらと等しいとする。G_0 はそのつくり方から、任意の点から出るどの２つの枝がラベルとしてもつ系列についても一方が他方の prefix になることはない。さらに G_M が L_A から L_B の上への写像を与える gsm の状態図であるとき、上述の G_0 の定義から G_0 は L_B の一つの拡張された（ prefix 形の）状態図である。したがつて G_B より状態の分割、prefix set による変換によつて得られる。

定義から L の系列全体と、G 上の初期状態から最終状態への（有限長の）道全体（ $P(G)$ とかく）が１対１に対応していることはあきらかである。L_A，L_B の状態図 G_A ，G_B が、枝の入力シンボルを度外視したとき等しい、すなわち、１－同形でかつ同形対応において初期状態は初期状態に、最終状態は最終状態に対応しているとする（以下単に等しいという）。G_A（G_B）において各枝にそれに対応する G_B（G_A）の枝がもつ入力シンボルに等しいシンボルを出力シンボルとして書き加えてできるグラフを状態図とする gsm を M_1（M_2）とすると、そのつくり方からあきらかに M_1 は L_A から L_B の上への１対１写像を、M_2 はその逆写像を与える。したがつて、$L_A \sim L_B$ である。以上から " ～ " の関係に着目する限り、状態図のかわりに枝のシンボルを度外視して、単に特別に初点といくつかの終点が指定された向きをもつたグラフを考えれば充分である。したがつて以下ではこのようなグラフ的な表現法を使うことにする❀（これは各枝の入力シンボルをその枝の固有の名

❀ G_M，G_I，G_0 についても同様。

前を示すシンボルそのものと考えることに相当する）

G_A ，G_B において$P(G_A)$から$P(G_B)$ の上への１対１対応（写像）が存在し、かつその対応および逆対応（写像）がgsmで実現できるとき、$G_A \sim G_B$（$G_A \rightarrow G_B$） と書く。

上述と、関係"～"が推移律をみたすことから、

補題 4.1： $L_A \sim L_B$（$L_A \rightarrow L_B$） の必要充分条件は、$G_A \sim G_B$（$G_A \rightarrow G_B$） である。

次節以下ではもっぱら$G_A \sim G_B$ の関係に着目する。

§ 4.3　関係"～"を保存する諸変形

この節では"～"の関係を保つようないくつかのGに関する基本変形を求め、それらを逐次行なつて、標準形に帰着させることを考える。

Gにおいて、節点nとそれから出る道で結ばれる節点全体およびそれらを結ぶ枝全体よりなる部分グラフをG(n)と書く。

まずいくつかのGに関する変形を定義する。

節点の分割：１つの節点nのかわりに$n_1, n_2, \cdots, n_k$ でおきかえ、つぎのように枝をつける。(i)　Gに枝$\overrightarrow{n'n}$（$n' \neq n$） があるとき、$\overrightarrow{n'n}$のかわりに１つのn_i（$1 \leq i \leq k$）をえらんで$\overrightarrow{n'n_i}$ をつける。 (ii) Gに枝$\overrightarrow{nn'}$（$n \neq n'$） があるとき、各n_i について$\overrightarrow{n_in'}$をつける。 (iii) nにh本のself-loop があるとき、各n_i についてh本の枝$n_in_{i_1}, n_in_{i_2}, \cdots$（$1 \leq i_1, \cdots \leq k$）（$n_i, \cdots$は重複してよい）をつける。（図 4.2）

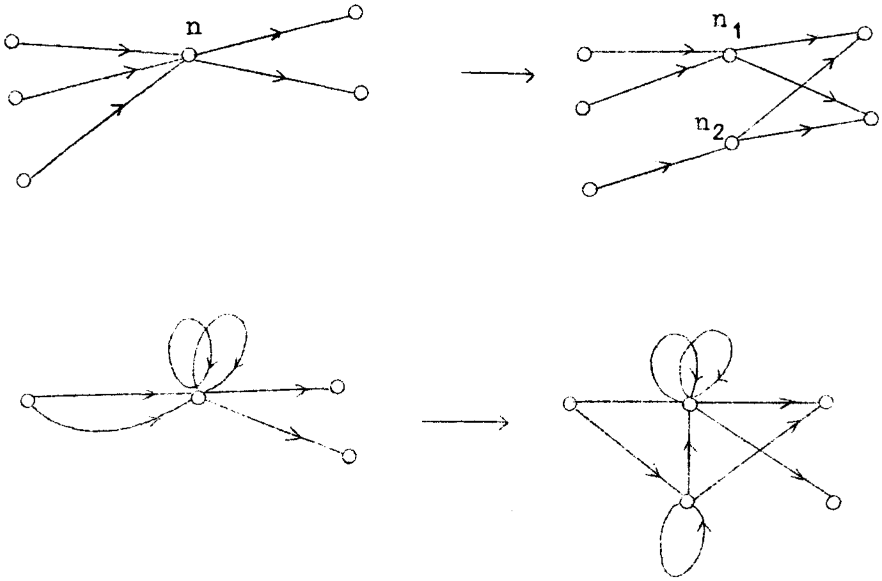


図 4.2　節 点 の 分 割

節点の消去：nをself-loopをもたず、始点、終点でないとする。nとそれに出入する枝をとり除き、かわりにnを通る長さ2の道で結ばれるすべての節点対n_1，n_2について、道の数だけの枝を結ぶ（図 4.3）

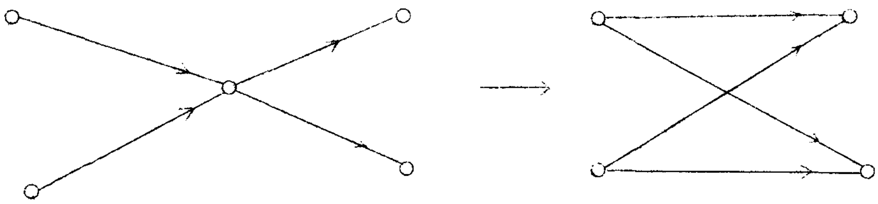


図 4.3　節 点 の 消 去

部分グラフの複製：nをGの節点とし、G(n)に属さない節点よりなる部分グラフG′からG(n)へ入射する枝$\overrightarrow{n_1 n}$，$\overrightarrow{n_2 n}$，・・・を任意にえらんだとする。G(n)と1－同形でGと共通点をもたないグラフ$\bar{g}$をつくり❀、nに対応す

❀　G(n)の終点に対応する点を終点とする。

る$\bar{g}$の節点を$\bar{n}$とする。$\overrightarrow{n_1 n}$，$\overrightarrow{n_2 n}$，・・・を取り除き、かわりに枝$\overrightarrow{n_1 \bar{n}}$，$\overrightarrow{n_2 \bar{n}}$，・・・を取り除き、かわりに枝$\overrightarrow{n_1 \bar{n}}$，$\overrightarrow{n_2 \bar{n}}$，・・・をつける。これは節点の分割のくり返しとみなせる。（図 4.4 ）

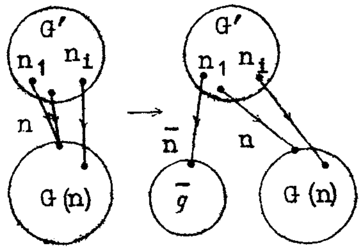


図 4.4　部分グラフの複製

部分グラフの併合：G の始点を含まない 1 一同形の部分グラフ g ，$\bar{g}$ よりG の残り G' の節点に入る枝が存在しないとする。このとき、g に含まれない $\bar{g}$ の部分をとり除き、とり除いた $\bar{g}$ の節点に G' から入る枝もとり除いて、かわりにそれぞれ g の対応する点へ入る枝をつける。

定義より直ちに、

補題 4.2：G に prefix set による変換、節点の分割、消去、部分グラフの複製、併合を行なつて得られたグラフを G' とすれば、G ～ G'

ここで以下で必要な補題を若干まとめておく。

補題 4.3　G の部分グラフ g がつぎの条件をみたすとする。(i) 節点 n_0 ($\in g$)以外の g の節点はそれから出る g の枝がないか、G においてそれに入る g に属さない枝がないかのいずれかである。それから出る g に属さない G の枝をもつ g の節点の集りを n_i ($a \leq i \leq b$) とする (n_0 がこれに含まれるとき、$a=0$ とし、それ以外 $a=1$ とする)。 (ii) 少なくとも $b+1$ 個の節点 $\bar{n}_i$ ($0 \leq i \leq b$)を含むグラフ $\bar{g}$ があり、n_0 ($\bar{n}_0$) を始点とし、n_i ($\bar{n}_i$) ($a \leq i \leq b$) を終点とする g ($\bar{g}$)の道全体を P_i ($\bar{P}_i$) とするとき、

$\bigcup_{i=a}^{b} P_i$ と $\bigcup_{i=a}^{b} \bar{P}_i$ 相互間にgsmによる1対1対応が存在し、かつこの対応において各 P_i と $\bar{P}_i$ の道が対応しているとする（以下このような関係の g , $\bar{g}$ を $g \overset{s}{\sim} \bar{g}$ $(n_0 ; n_a , \cdots , n_b)$ と書く。

このとき、G において g に属する枝と n_i（$0 \leq i \leq b$）以外の g の節点を取り除き、かわりに n_i と $\bar{n}_i$ を一致するように $\bar{g}$ をつなぎかえて得られるグラフを $\bar{G}$ とすると（図 4.5）、$G \sim \bar{G}$。

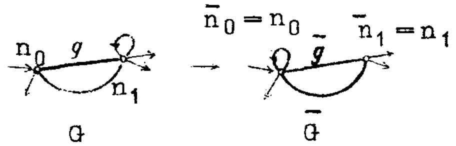


図 4.5　部分グラフの変換

（証明）　p（$\in P(G)$）の g に属さない部分道にはそれ自身を、g に属する部分道（それは(i)より n_0 から n_i への g の道である）には(ii)で与えられている対応によつて、$\bar{g}$ の道を対応させると、$G \sim \bar{G}$ は(ii)より明白である。

補題 4.4❀**:** 図 4.6で与えられる g , $\bar{g}$ について $g \sim \bar{g}$ $(n_0 ; n_0 , n_1)$

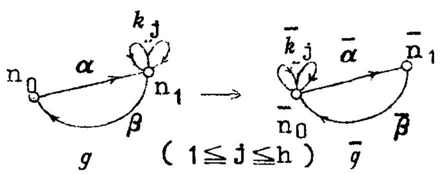


図 4.6　補題 4.4のグラフ g と $\bar{g}$

補題 4.5❀**:** g を節点 n ; $n_0 , \cdots , n_b$ と n_0 , n におけるそれぞれ ρ_0（≥ 0）, ρ（> 0）本のself-loop , n_0 から n への ν_0（≥ 0）本の枝、

❀　証明は5章参照。

n_0 ，nからn_i（$1 \leqq i \leqq b$）へのそれぞれμ_i（$\geqq 0$）本、ν_i（>0）本の枝よりなるグラフとする。また$\bar{g}$をgと同じ形でただ枝の本数がことなるグラフとし、節点、枝数を示す記号をgにおける対応する記号に"－"をつけて示すことにすると

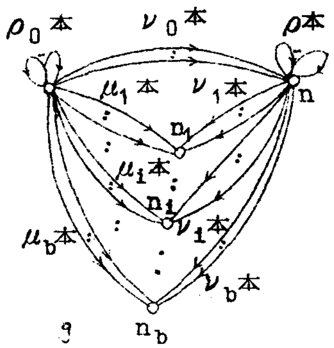


図 4.7　補題 4.5.のグラフ

(a)　ρ，$\bar{\rho} \geqq 2$ならば、$g \overset{s}{\sim} \bar{g}$（$n_0$；$n_0$，$n_1$，・・・，$n_b$）

(b)　$\rho_0 = \bar{\rho}_0$；$\rho = \bar{\rho} = 1$；$\nu_0 = \bar{\nu}_0$かつ

$$\nu_1 / \bar{\nu}_1 = \cdots = \nu_b / \bar{\nu}_b$$

ならば、$g \overset{s}{\sim} \bar{g}$（$n_0$；$n_0$，$n_1$，・・・，$n_b$）

（ただし、n_0（$\bar{n}_0$）からでる枝がないときは、すなわち$\rho_0 = \nu_0 = 0$（$\bar{\rho} = \bar{\nu}_0 = 0$）のとき、$n_0 = n$（$\bar{n}_0 = \bar{n}$）とする）。

以上の準備のもとに標準形への変形を説明しよう。まずGをleaf graph※に分解したとする。2つのleaf l_1，l_2について、l_1からl_2への道があるとき、l_1（l_2）はl_2（l_1）より上位（下位）にあるということにする。最上位のleaf（始点を含むleaf）から順次、つぎの操作を行なつたと考える。

※　互いに向きをもつた道で結ばれている節点全体の集合とそれらを結ぶ枝よりなる部分グラフをleaf graphとよぶ。[5] 以下、単にleafとよぶ。

問題のleaf l が2つ以上の節点をもつとする。l の2点以上に上位のleaf からの枝が入射しているときは、それらの点について部分グラフの複製を適当に行なつて、l ($=l^{(0)}$) およびそれと同形のleaf $l^{(1)}$ のそれぞれ1点 $n^{(0)}$, $n^{(1)}$, ・・・ のみに上位のleaf からの枝が入射するようにしたとする。$l^{(1)}$ の節点のうち $n^{(1)}$ 以外でself-loop をもたない節点があれば、節点の消去を行なう。またself-loop をもつ $n^{(1)}$ 以外の節点があるとき、その1つを n とする。n に入る枝 α (以下枝といえばself-loop を意味しない)に着目する。α の始点 n_0 は仮定より同じleaf $l^{(i)}$ に属するから、n から n_0 への道が存在する。そのうち長さ最小のものを β, n のself-loop の集りを $\{k_i\}$ とする。$\{k_i\}$, β を含む n についての最小のprefix set に関して変換を行なつても同じ類のグラフが得られるから、それをあらためて G とおき、前と同じ記号を使う。この変形により n に入る枝の数はふえることはない。この数が2以上のときは n を n_1, n' に分割して n_1 に入る枝数が1になるようにする。ただし n_1, n' には n と同数のself-loop をつける。節点 n_0, n_1, 枝 α, β と n_1 のself-loop $\{k_i\}$ よりなる G の部分グラフを g とする。この g は補題4.4の g と同じ形をしているから、補題4.3、4.4が使える。G においてself-loop $\{k_i\}$ を節点 n_0 に移動して得られるグラフ $\bar{G}$ は G と同じ類に属する。$\bar{G}$ において n_1 はself-loop をもたないから消去できる。この操作によつて n' に入る枝の数はふえない。n' について同じことをくりかえすと、結局 $l^{(1)}$ の節点の数が1つ減る。

これをくり返すと、$l^{(1)}$ は ρ 個のself-loop をもつた1点 $n^{(1)}$ に変形される。はじめに $l^{(1)}$ が1つの閉路❀のみをもつときはあきらかに $\rho=1$ である。また2つ以上の閉路をもつときは $\rho\geqq 2$ である。補題4.3と補題4.5の(a)よ

❀ 向きをもつた閉路。self loopも含める。

り、$\rho \geqq 3$ ならば $\rho=2$ におきかえても同類に属することがわかる。したがつて、ρ は1または2とできる。

変形前に $l^{(i)}$ から下位の leaf の1つの節点に μ 本（$\mu>0$）の枝が入射しているとき、上記の変形後 $n^{(i)}$ から同じ節点へ少なくとも μ 本の枝がある。$\rho=1$ のときは、単に1つの閉路上の節点の消去のみを行なうから、あきらかに本数はかわらない。$\rho=2$ のときは補題 4.3 と補題 4.5 の(a)により1本の枝でおきかえることができる。各 $l^{(i)}$ について上述の変形を行なうとき、変形後 $G(n^{(i)})$ は互いに1ー同形であることはあきらかである。したがつて部分グラフの併合により、$G(n^{(0)})$ のみを残し、$n^{(i)}$（$i \neq 0$）に上位の leaf から入射する枝をすべて $n^{(0)}$ につなぎかえてよいことがわかる。

以上の操作をまとめると（図 4.8）

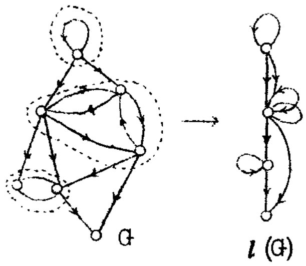


図 4.8　l(G) の説明

（i）　G を leaf に分解し、始点、終点以外で閉路をもたず1点のみよりなる leaf を節点の消去により消す。

（ii）　leaf l が1つのみ閉路をもつときは1つの self-loop をもつ1節点 n_l（以下このような節点の集りを D_1 と書く）でおきかえ、l が2つ以上の閉路をもつときは2重の self-loop をもつ1節点 n_l（以下このような節点の集りを D_2 と書く）でおきかえる。

（iii）　変形前 leaf l から leaf l' へ μ 本（$\mu>0$）の枝があつたとす

る。n $\in D_1$のときはμ本、$n_l \in D_2$のときは1本の枝をn_lからn_l'へ結ぶ。

上の操作でGから得られたグラフをl(G)とおくと、上述より、G ~ l(G)。つぎにl(G)について"~"の関係を保存する基本操作を説明する。

(i) 基本的でない枝の除去: 枝bの両端点を結ぶ長さ2以上の道がある(ない)ときbを基本的でない(な)枝という。基本的でない枝を取り除いても同じ類に属することは、補題4.3と補題4.5でμ_1, $\bar{\mu}_1$が任意であることに着目すれば、道の長さが2のとき成立することがわかる。長さが3以上のときは、同じことをくり返し利用すればよい。(図4.9)

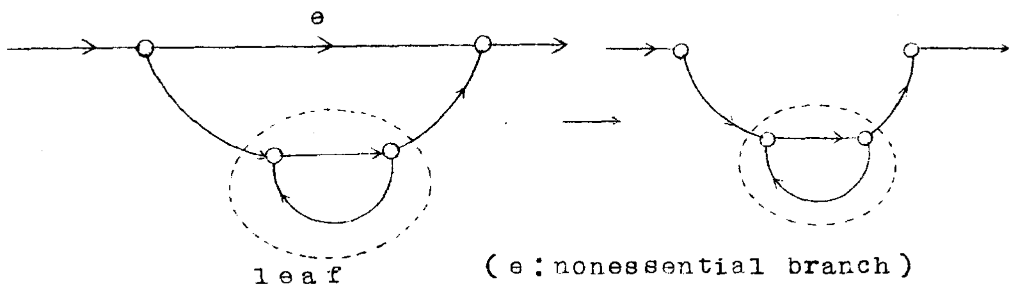


図4.9 基本的でない枝の除去

(ii) 2節点間を結ぶ枝数の減少:

(iia) 2節点n_1, n_2の少なくとも一方がD_2に属するとき、それらを結ぶ2本以上の枝を1本に減らす。(図4.10)

(iib) $n \in D_1$, nからの枝が入射しているD_1の節点を、n_1, n_2, ・・・, n_h, nからn_1への

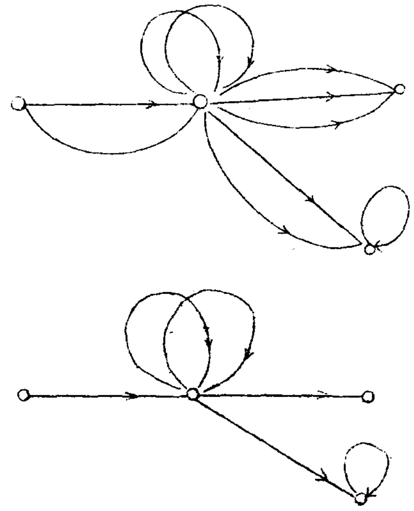
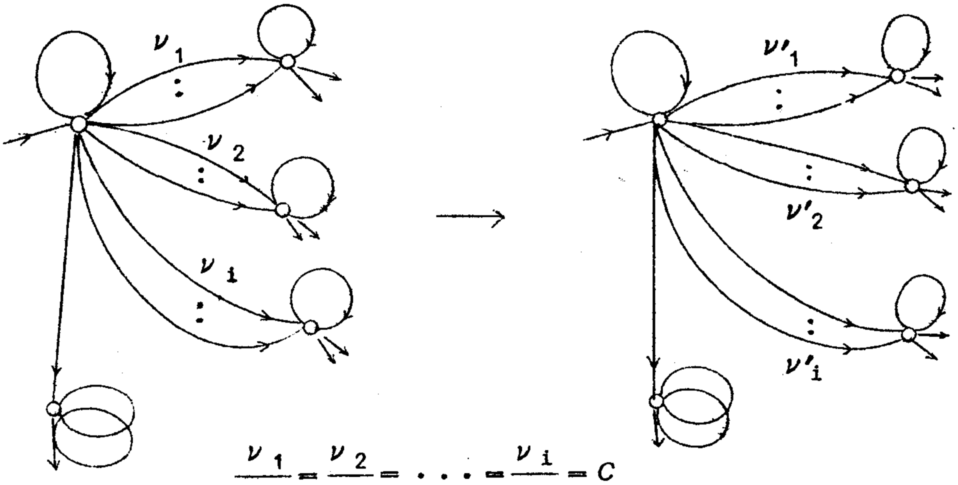


図4.10 枝数の減少(a)

枝数をν_i かつ$\nu_1, \nu_2 \cdots, \nu_h$の最大公約数を$\nu$とする。nから$n_i$ への枝数をν_i / ν 本に減らす。（図 4.1 1）



$$\frac{\nu_1}{\nu'_1} = \frac{\nu_2}{\nu'_2} = \cdots = \frac{\nu_i}{\nu'_i} = C$$

図 4.1 1　枝 数 の 減 少（ b ）

(iia) は補題 4.3 と補題 4.5 の(a)でν_0, ν_i（>0）が任意であることからあきらかであり、(iib) はまずnからD_2 の節点に枝が入射しているとき、これらをν本の枝でおきかえる（補題 4.5 の(a)では$\nu_0, \bar{\nu}_0$が任意である）。あとは補題 4.3 と補題 4.5 の(b)よりあきらかである。

（iii）　節点の併合

併合条件 1 ：n_1 ，n_2 ともにD_1 またはD_2 に属し、他の任意の節節点nに対し、n_1, n_2から出てnに入る基本的な枝の数（ないときは0として）が等しい。

併合条件 2 ：n_1がD_2かつn_2 から出る基本的な枝がすべてn_1 に入射している。

上の条件がみたされているとき、n_2 とそれから出る枝をとり除き、n_2

に入る枝をすべてn_1に入る枝としてつけかえる。

条件1の成り立つときは基本的でない枝をとり除くと、$G(n_1)$と$G(n_2)$が1－同形となり、部分グラフの併合になる。条件2については、補題4.3と補題4.5の(i)をρ_0, ν_0 $(\neq 0)$ を任意とし、$\bar{\rho}_0=\bar{\nu}_0=0$ として適用すればあきらかである。

§4.4 標　　準　　形

$l(G)$から標準形$S(G)$への変形の手続きを示す。

（1）　終点を1つに併合し、これをレベル0の点とする。この点に入る基本的でない枝を除去する。レベル0の点へのみでる枝をもつ節点をレベル1の点とする。

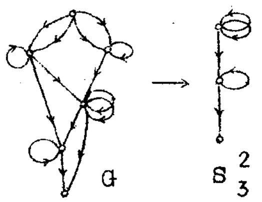


図4.12　標準化の説明

（2）　レベル$\nu-1$以下の節点について標準化が終わり、かつ、レベルνの点の集り、V_νが定まつたとする。

（2a）　V_νの各節点について、それから出る枝数の減少操作(ii)を行なう。

（2b）　V_νの2節点でともにD_1またはD_2に属する節点対について、併合条件1がみたされているかどうかをみて、みたせば(iii)により併合す

る。

（2c） V_ν のD_2 に属する各節点に対して、それに入射する各枝の始点について併合条件2がみたされているとき、（iii）により併合し、あらたに発生した基本的でない枝を除く。可能なかぎり（2c）をくり返す。

（2d） V_ν のD_2 に属する節点について、それへ入射する枝の数を（iia）により減らす。

（2e） V_ν がD_1 に属する1節点nのみからなり、残りの節点が始点のみで、始点から出る枝が1本のみであるとき、始点とnとを併合する。

（2f） 残りの節点がなければ操作は完了し、得られたグラフがS(G)である。もしあればV_ν の節点に入る基本的でない枝を除去し、それから出る枝がV_j（$0\leqq j\leqq\nu$）の節点へのみ入射しているような節点をレベル$\nu+1$とし、(2)へもどる。

S(G)は一意にきまり、[*] 次式が成立する。

$$G\sim S(G),\ SS(G)=S(G)$$

以上まとめて、

定理 4.1： $L_A\sim L_B$ であるための必要充分条件は

$$S(G_A)=S(G_B)$$

充分条件の部分は$G_A\sim S(G_A)$, $G_B\sim S(G_B)$ より出、必要条件の証明は5章§5.3に示す。標準化の過程で節点、枝ともふえることはないから定理の系として、

系1：S(G)はその類のなかで節点、枝数の最小なグラフである。

なおS(G_A)=S(G_B)の判定法は、始点の含まれるレベルをレベル数N_L とよぶことにすると、

---

[*] 1ー同形の範囲で。

1) 両者のレベル数が等しく、各レベルのD_jに属する節点数が互いに等しくなければならぬ。

2) 各レベルの同じD_jに属する節点間につぎのような1対1対応がつかねばならぬ。

2a) レベル0の点を対応させる。レベルi－1以下についてすでに対応応がついているとする。

2b) 一方のレベルiのD_jの各節点について、他方のレベルiのD_jの節点のなかにそれらからレベルi－1以下の対応する節点への枝数が等しいものが1つだけ存在する。このような2点を対応させる。

この操作を順次上のレベルについて行う。

もつとも単純な言語はX^*そのものである。❀ アルフアベットXの含むシンボルの数$|X|$が3以上のとき、補題5から$|X|=2$の言語S_2^1と同類である。なおレベル1の$l(G)$の節点は操作（2a），（2b）により同じD_1 D_2に属する節点同志互いに併合されるから、$S(G)$のレベル1の節点はたかだか2点しかない。例として$N_L=1$および2の標準形を図4.13に示す。

系2；LがX^*（$|X|\geq 2$）と同類であるための必要充分条件はLの状

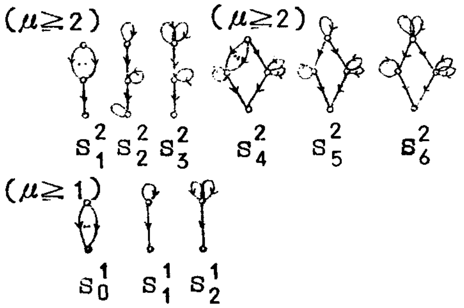


図4.13　$N_L=1$と$N_L=2$の標準形

態図Gにおいて、それから終点へのすべての道が閉路を含む他のleafを通らないようなどのleafも2つ以上の閉路を含むことである。

系2の条件は$l(G)$のレベル1の点がすべてD_2に属することをいいかえた

❀ ただし、ここでは"end mark"のつけたものを考える。

にすぎない。すべてD_2ならば併合条件1、2によりS_2^1に帰着する。

§ 4.5 gsmによる写像の存在条件

$L_A \to L_B$の条件について考える。一般にG'がGの部分グラフ❀ならばあきらかに$G \to G'$である。

Gを与えられたグラフとし、まず$S(G)$が少なくとも1つD_2の点を含むとすると、$S(G) \to S_2^1$、G'を任意に与えられたグラフとし、$S(G')$がD_1の点をもてばそれをすべてD_2の点におきかえて得られるグラフを$\bar{S}$とすると、併合条件1、2より$S_2^1 \sim \bar{S}$、また$\bar{S} \to S(G')$。したがつて、$G \sim S(G)$, $G' \sim S(G')$より、$G \to G'$が得られる。つぎに$S(G)$がD_2の点を含まないときを考える。D_2の点を含まない標準形は枝数の減少操作と併合条件1による併合を逐次行なうことができるから、図4.14に示した形以外にはない。D_1に属する節点数$Z(G)$と始点から最初のD_1の節点への枝数$\mu(G)$(≥ 2)(図4.14の右図のとき$\mu(G)=1$と規約する)により標準形は完全に定まる。$Z(G)$、$\mu(G)$はGから直接つぎのように定義できる。

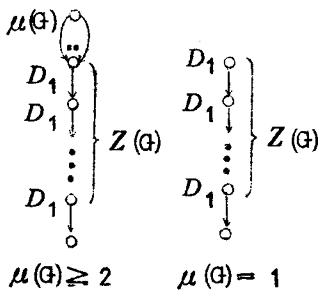


図4.14　D_1のみからなる標準形

❀ G'の始点、終点は必ずしもGのそれらに一致する必要はない。

Z(G)：始点から終点への道が含みうる閉路の最大数。μ(G)：Z(G)個のことなる閉路を通る始点から終点への道の最初の枝となるような枝の数、ただし、始点が閉路に含まれているときは1とする。

定理 4.2：L_A，L_B の状態図を G_A，G_B とすると、

（a）　G_A が2つ以上の閉路を含む leaf をもてば、任意の L_B に対して、$L_A \rightarrow L_B$。

（b）　G_A が2つ以上の閉路を含む leaf をもたないとき、$L_A \rightarrow L_B$ となるための必要充分条件は、G_B も2つ以上の閉路を含む leaf をもたず、かつ $Z(G_A)>Z(G_B)$ または、$Z(G_A)=Z(G_B)$，$\mu(G_A)\geqq\mu(G_B)$

（証明）(a)はすでに示した。(b)の場合、まず $L_A \rightarrow L_B$ と仮定し、その写像を与える gsm の状態図を G_M、その入力グラフ、出力グラフを G_I，G_0 とする。§4.2で述べたように、G_I は G_A から節点の分割を適当に行なつたものであるから G_I は2つ以上の閉路を含む leaf をもたず $Z(G_I)=Z(G_A)$，$\mu(G_I)=\mu(G_A)$ である。また G_0 は G_B より節点の分割、prefix set による変換によつて得られる。

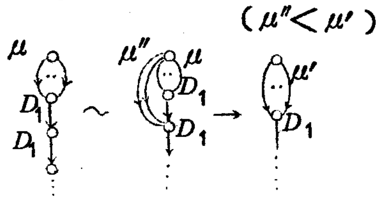
G_0 の任意の道 P_0（対応する G_M の出力系列を α）に対して、G_M 上出力系列が α を一部分として含むような（G_I の）道の集り $\{P_{I_i}\}$を対応させる。❀ ν を充分大きくとることにより、P_0 が閉路 $\bar{C}$ を ν 回まわつた部分を含むとき、どの P_{I_i} も少なくとも1つの閉路を含むようにできる。また G_0 のつくり方から P_0 がことなる閉路 $\bar{C}_1$，$\bar{C}_2$ を前後して ν 回ずつまわつた部分を含むとき、どの P_{I_i} も少なくとも異なる2つの閉路を含む。G_B したがつて G_0 がかりに2個以上の閉路をもつ leaf を含むとする。P_0 としてその leaf の閉路 $\bar{C}_1$，$\bar{C}_2$ を ν 回ずつ交互に充分なだけくりかえしてまわつた道

❀ 上への写像ゆえ空でない。

を考える。G_I では1度閉路の外にでれば再びもとにもどれないから、G_I が無限に異なる閉路を含むことになり不合理である。

同様にして、$Z(G_A) \geqq Z(G_B)$ が示される。また、$Z(G_A)=Z(G_B)$ のとき $\mu(G_A) \geqq \mu(G_B)$ も容易に示される。

逆に $S(G_B)$ が D_2 の点を含まず、$Z(G_A)=Z(G_B)$ ，$\mu(G_A) \geqq \mu(G_B)$ ならばあきらかに $G_A \to G_B$ である。つぎに $Z(G_A) > Z(G_B)$ とする。$\mu(G_A) \geqq \mu(G_B)$ ならあきらかゆえ $\mu(G_B) > \mu(G_A)$ とする。このとき図4.15よりわかる。



基本的でない枝を加える　一番上のloopを除く

定4.15　定理4.2(b)の説明

定理4.2から、

定理4.3：無限の文章を含む任意の有限状態言語 L_A ，L_B に対して、その写像像も無限の文章を含むような L_A から L_B の中へのgsmによる写像が存在する。

§4.6　結　　　言

翻訳機のよいモデルとしてのgsmにより、翻訳可能であるという関係"～"によりFSL全体を類別することができる。任意のFSLに対し、そ

れが含まれる類を一意的に代表し、状態数最小で構造の簡単な標準言語が得られた。

便宜上複数個の最終状態を許したが、ただ１つに限つても本質的に変わらない。このときＧの枝の向きを逆転したグラフをρ(G)（ρは系列の前後を逆転する操作）とすると、ρ(G)も１つのＦＳＬの状態図である。

Ｓ（ρ（Ｇ））とＳ(G)はかならずしも一致しない。この非対称性を除くには、写像を two passにして変換$\rho M_2 \rho M_1$(14)を考えればよい。この場合に前述の結結果を拡張することは容易である。

第5章　第4章の補足

§5.1　補題4.4の証明

$K=\{k_j\}$，$\bar{K}=\{\bar{k}_j\}$とおく。$n_0(\bar{n}_0)$から$n_1(\bar{n}_1)$への$g(\bar{g})$上の道が与えられたとき、つぎのようなm，γ，$H_i(\bar{H}_i)$が一意にきまる。❀

$P_1(0;\gamma)=(\alpha\beta)^{\gamma}\alpha$，$\gamma\geq 0$ $(\bar{P}_1(0;\gamma)=(\bar{\alpha}\bar{\beta})^{\gamma}\bar{\alpha}$，$\gamma\geq 0)$

$P_1(m;\gamma;H_1,\cdots,H_m)=(\alpha\beta)^{\gamma}\alpha H_1\beta\alpha H_2\cdots\beta\alpha H_m$

$(\bar{P}_1(m;\gamma;\bar{H}_1,\cdots,\bar{H}_m)=(\bar{\alpha}\bar{\beta})^{\gamma}\bar{H}_1\bar{\alpha}\bar{\beta}\bar{H}_2\cdots\bar{\alpha}\bar{\beta}\bar{H}_m\bar{\alpha})$

$m\geq 1$，$\gamma\geq 0$，$H_i\in K^*$　$(\bar{H}_i\in\bar{K}^*)$，$H_1\neq\Lambda$　$(\bar{H}_1\neq\Lambda)$。

同様に$n_0(\bar{n}_0)$から$n_0(\bar{n}_0)$への$g(\bar{g})$上の道は

$P_0(0;\gamma)=(\alpha\beta)^{\gamma}$，$\gamma\geq 1$ $(\bar{P}_0(0;\gamma)=(\bar{\alpha}\bar{\beta})^{\gamma}$，$\gamma\geq 1)$

$P_0(m;\gamma;H_1,\cdots,H_m)=(\alpha\beta)^{\gamma}\alpha H_1\beta\alpha H_2\cdots\beta\alpha H_m\beta$ $(\bar{P}_0$ $(m;\gamma;\bar{H}_1,\cdots,\bar{H}_m)=(\bar{\alpha}\bar{\beta})^{\gamma}\bar{H}_1\bar{\alpha}\bar{\beta}\bar{H}_2\cdots\bar{\alpha}\bar{\beta}\bar{H}_m)$

$m\geq 1$，$\gamma\geq 0$，$H_i\in K^*$　$(\bar{H}_i\in\bar{K}^*)$，$H_1\neq\Lambda$　$(\bar{H}_1\neq\Lambda)$

ここで$P_d(0;\gamma)$と$\bar{P}_d(0;\gamma)$ $(d=1,\gamma\geq 0;d=0,\gamma>0)$，$P_d(m;\gamma;H_1,\cdots,H_m)$と$\bar{P}_d(m;\gamma;\bar{H}_1,\cdots,\bar{H}_m)$ $(m\geq 1,\gamma\geq 0,$ $H_i=k_{i1}k_{i2}\cdots k_{iq_i}$，$\bar{H}_i=\bar{k}_{i1}\bar{k}_{i2}\cdots\bar{k}_{iq_i}$，$H_1\neq\Lambda)$を対応させると、1対1でもれなく対応している。$(\alpha\beta)^{\gamma}$，$\bar{H}_i$がそのまま対応しているから、有限な記憶で充分であることが容易にたしかめられる。

❀　$(\alpha\beta)^{\gamma}=\underbrace{\alpha\beta\alpha\beta\cdots\cdots\alpha\beta}_{\gamma}$

§ 5.2　補題 4.5 の証明

(a)を四つの場合

(i) $\bar{\rho} \geqq 3$ が $\rho = 2$ にできること

(ii) ν_i が任意であること

(iii) μ_i が任意であること

(iv) ν_0 が任意であること

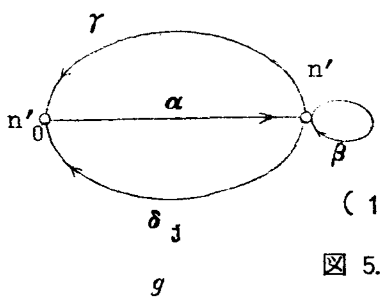
について証明する。

(i) $\rho=\tau+2$，$\bar{\rho}=\tau+3$ ($\tau \geqq 0$) で、それ以外両グラフが等しいとき❀

(図 5. 1 (a))

図 5. 1 (a)

図 5. 1 (b)のような g'，$\bar{g}'$ を考える。n'_0，$\bar{n}'_0$ から途中 n'_0，$\bar{n}'_0$ を通らな



($1 \leqq j \leqq \tau$)

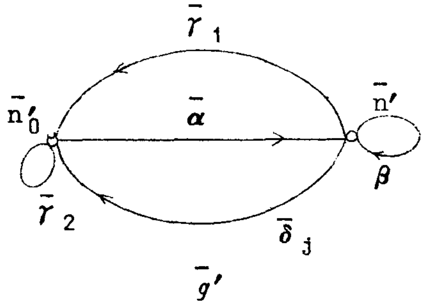


図 5. 1 (b)

いでそれぞれ n'_0，$\bar{n}'_0$ にかえる g'，$\bar{g}'$ 上の道をたがいにつぎのように対応させる。

❀ 以下 g，$\bar{g}$ において枝数の等しいものはかかない。

$\alpha\beta^{n}\delta_{j} \longleftrightarrow \bar{\alpha}\bar{\beta}^{n}\bar{\delta}_{j}$ $(n \geqq 0)$；$\alpha\gamma \leftrightarrow \bar{\gamma}_{2}$；$\alpha\beta\beta^{n}\gamma \leftrightarrow \bar{\alpha}\bar{\beta}^{n}\bar{\gamma}_{1}$ $(n \geqq 0)$
補題 4.3 の (ii) の条件があきらかにみたされている。ゆえに $g' \overset{s}{\sim} \bar{g}'(n'_0 : n'_0)$。$g'$，$\bar{g}'$ に補題 4.4 を適用し、n'，$\bar{n}'$ を消去すれば、ρ，$\bar{\rho}$ 本の self-loop をもつ一点よりなるグラフが " $\overset{s}{\sim}$ " の関係にあることがわかる。あとは補題 4.3 より明らか。

(ii) $\rho = 2$，$\bar{\rho} = 3$，$\nu_1 = 1 + \tau$，$\bar{\nu}_1 = 2 + \tau$ $(\tau \geqq 0)$ のとき（図 5.2 (a)）図 5.2 (b) の g' の節点 n' において prefix set $\{\beta^2, \gamma, \delta, \beta\gamma, \beta\delta\}$

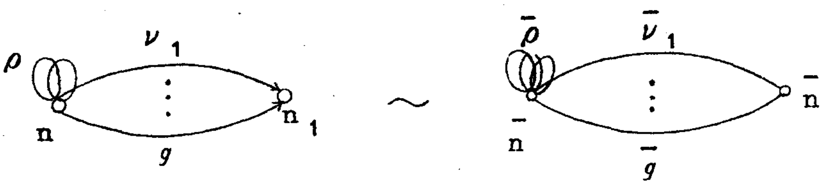


図 5. 2 (a)

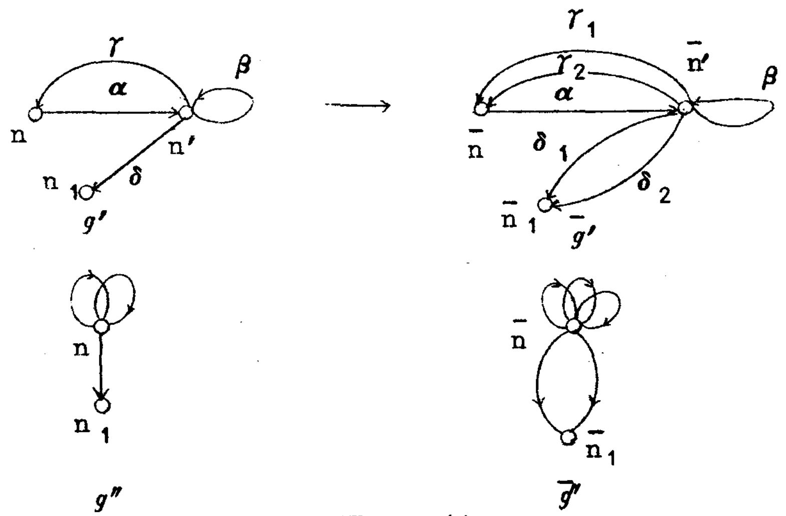


図 5. 2 (b)

による変換を行なえば $\bar{g}'$ が得られる。g'，$\bar{g}'$ に補題 4.4 を適用し、n'，$\bar{n}'$ を消去すれば、g''，$\bar{g}''$ が得られ、$g'' \overset{s}{\sim} \bar{g}''(n ; n, n_1)$。補題 4.3 により結果を得る。

(iii) $\nu_0=\bar{\nu}_0=1$, $\mu_1=0$, $\bar{\mu}_1=1$, （図 5.3(a)）、便宜上先ず $\rho=\bar{\rho}=1$

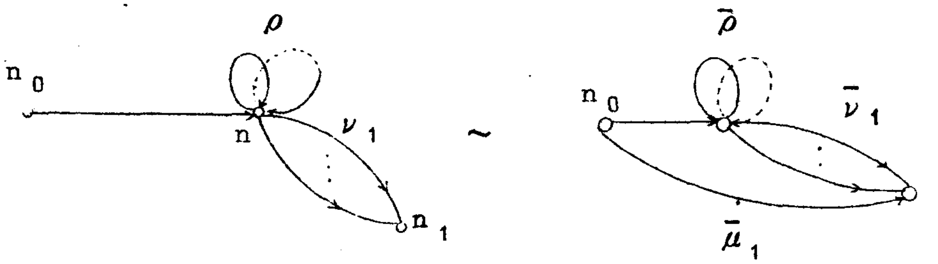


図 5.3(a)

のときを考える。P_i（n_0 から n_i への道の集り）と $\bar{P}_i$（$\bar{n}_0$ から $\bar{n}_i$ への道の集り）との間に、次のような補題 4.3 の (ii) をみたす対応をつけうる。

$\alpha_k\varepsilon_{11}\leftrightarrow\bar{\gamma}$; $\alpha_k\beta\beta^n\varepsilon_{11}\leftrightarrow\bar{\alpha}_k\bar{\beta}^n\bar{\varepsilon}_{11}$; $\alpha_k\beta^n\varepsilon_{ij}\leftrightarrow\bar{\alpha}_k\bar{\beta}^n\bar{\varepsilon}_{ij}$

$[(i, j)\neq(1, 1)]$ 。ただし $n\geqq 0$（図 5.3(b)）

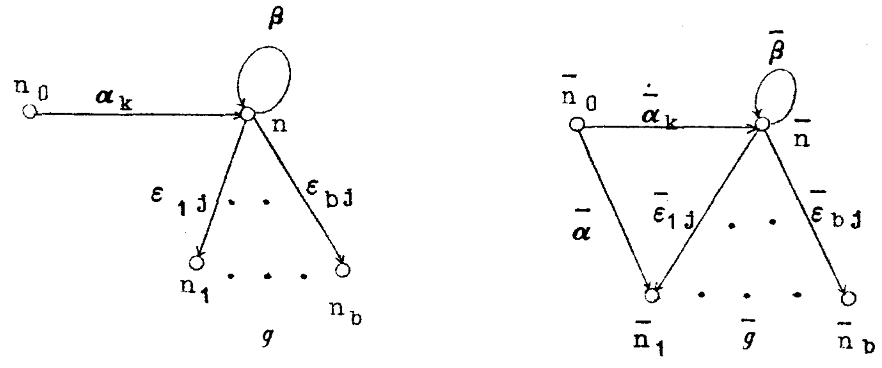


図 5.3(b)

(i) よりあと $\rho=\bar{\rho}=2$ としてよい。g において n を n′, n″ に分割し、n′ から n″ へ1本の枝、n′, n″ にそれぞれ 1、2本の self-loop をつける（図 5.3(c)）。n_0, n′, n'_0 からなる部分グラフに $\rho=\bar{\rho}=1$ の結果を適用して、n_0 から n_1 に枝をつけ、ふたたび n′, n″ を併合すれば $\bar{g}$ が得られる。

(iv) $\rho_0=\nu_0=\mu_1=0$, $\rho=\bar{\rho}=\bar{\rho}_0+\bar{\nu}_0$, $\nu_1=\bar{\nu}_1=\bar{\mu}_1$ のとき。
（図 5. 4 (a)）

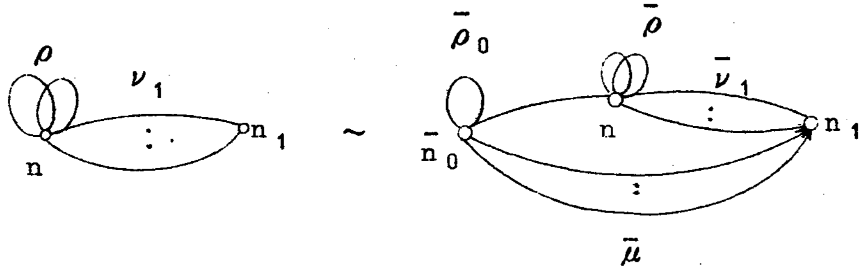


図 5. 4 (a)

(i)より $\rho=\bar{\rho}$ が2本以上任意であることと、（iii）の後半より明らか。

以上から“$\overset{s}{\sim}$”が推移律をみたすことと、補題 4. 3 より(a)が示される。

（ b ） $\rho=\bar{\rho}=1$ で $\nu_1/\bar{\nu}_1=\nu_2/\bar{\nu}_2=\cdots=\nu_b/\bar{\nu}_b=C$ とする❀。このときの節点 n における self-loop の名前を ρ，ν_1 本の枝の名前を ν_{11}，ν_{12}，$\cdots\nu_{1\nu_1}$，このとき、節点 n において pretix set $\{\rho^c, \nu_{11}, \nu_{12}, \cdots\nu_{1\nu_1}, \rho\nu_{11}, \rho\nu_{12}, \cdots, \rho\nu_{1\nu_1}, \rho^2\nu_{1\nu_1}, \cdots, \rho^{c-1}\nu_{11}, \cdots, \rho^{c-1}\nu_{1\nu_1}\cdots\}$ による変換を行えばよい。さらに前述の(a)の証明（iii）より $\mu_1=\bar{\mu}_1$ とおけること、“$\overset{s}{\sim}$”が推移律をみたすこと、および補題 4. 3 より(b)が証明される。

§ 5. 3　定理 4. 1 の必要条件の証明

目標は $L_A\sim L_B$（$G_A\sim G_B$；G_A，G_B は L_A，L_B の簡約状態図）ならば、$S(G_A)=S(G_B)$ を示すことである。$L_A\sim L_B$ とし、条件 1、2 をみたす L_A から L_B への写像を与える gsm の状態図を G_M とし、その入力グ

❀ 以下、g，$\bar{g}$ において枚数の等しいものは、かかない。

ラフ、出力グラフをG_I，G_0とする。

補題 4.6：Gにprefix set による変換、節点の分割、消去、部分グラフの複製、併合を行つて得られたグラフをG′とすれば、$S(G)=S(G')$

証明は定義より容易である。この補題とG_I，G_0の性質から（§4.2）

$$S(G_A)=S(G_I),\ S(G_B)=S(G_0) \tag{1}$$

こゝでまずつぎの補題に注意する。

補題 4.7：つぎの性質をもつG_M固有の正整数L_{max}が存在する：u_1，u_2を最初のシンボルがことなり、長さがL_{max}より小さくない任意の2系列とする。またS_iをG_Mの任意の状態とし、u_1，u_2をS_iを出発点としたときの入力系列として共に意味をもつとする。❀ このとき、u_1，u_2に対する出力系列v_1，v_2は一方が他方をprefixとしてもつ（等しい場合を含めて）ことはない。

証明は逆写像がまた有限状態のgsmで実現できなければならないことからあきらかである。

$S(G_A)$のレベル数N_Lについての数学的帰納法により証明することにし、$N_L \leqq Z$のG_Aについて定理が成立すると仮定する。

補題6よりG_Aのかわりにその標準形が$S(G_A)$に等しいような任意のグラフを考えてよい。G_Aをレベル数Zの標準形とし、G_Aの最高レベル節点（すなわち始点）n_0のself-loopを$\{a_i\}$（self-loopをもたないときには空集合とする）、n_0から下のレベルの節点への枝を$\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_r$とする。各$G_A(\varepsilon_i)$を$G_{Ai}$とする。$S(G_{Ai})$のレベル数は$Z$より小である。$G_I$は$G_A$より節点の分割によつて得られることは§4.2に述べた

❀ $\sigma(S_i, u_1) \in F$，$\sigma(S_i, u_2) \in F$

が、$\{n_0^{(j)}\}$をG_Aのn_0から分割されてえられたG_Iの節点の集りとし、$\{n_0^{(j)}\}$とそれらの節点を結ぶ枝よりなるG_Iの部分グラフをG_{I0}とする。

こゝでG_Mの各節点において"end mark"でおわるprefixを除いて、prefixの長さが補題4.7のL_{max}以上であるようなprefix setをえぶ。とくに各$n_0^{(j)}$においてはつぎのようなprefix setをえらぶことにする。(1) $\{a_i\}$のシンボルよりなる長さL_{max}のすべての系列、および (2) $\{a_i\}$のシンボルよりなる長さ$L_{max}-1$以下のすべての系列に番号をつけ$\{\alpha_\tau\}$とする。各α_τ，ε_jについてつぎの形の系列$\alpha_\tau \varepsilon_i u$（ただし、uはend mark で終わらないときは長さが$L_{max}-1$であるような系列）。

$n_0^{(j)}$のprefix setのうち$\alpha_\tau \varepsilon_i u$の形のprefixの集りを$Q_{\tau i}^{(j)}$とする。各点において上のprefix setによる変換を行い、あらためてG_Mとする。（補題4.6よりG_Iは変形後も同じ標準形をもつ）。G_Iにおいて$Q_{\tau i}^{(j)}$に属するprefixに対応する枝$\{b_k\}$に対して、G_Iの部分グラフ$\bigcup_k G_I(b_k)$を$G_{\tau i}^{(j)}$とする。変形前のG_Mにおいて入力シンボルε_iをもつ枝を含む閉路が存在しなかつたら、適当に部分グラフの分割を行つて、$n_0^{(j)}$を除いて、$G_{\tau i}^{(j)}$が互いに共通節点をもたないようにできる。こうしたものをあらためてG_Mとおく。そのつくり方から$G_{\tau i}^{(j)}$はG_{Ai}に有限回節点の分割、prefix setによる変換を行つて得られたものであるから補題 4.6より

$$G_{Ai} \sim G_{\tau i}^{(j)} \qquad (2)$$

一般にG_M（G_I）のいくつかの節点が併合されてG_0の1つの節点となる。G_Mの節点nに対応するG_0の節点をf(n)とかく。節点集合$\{f(n) \mid n \in G_{\tau i}^{(j)}\}$（$\{f(n_0^{(j)})\}$）とそれに含まれる2節点を結ぶ道の上にある節点、枝よりなるG_0の部分グラフを$G_{\tau i0}^{(j)}$（G_{00}）とおくと（図4.16）

補題 4.8：$(j,\tau,i) \neq (j',\tau',i')$なら$f(n_0^{(j)})$，$f(n_0^{(j')})$を除いて、

$G_{\tau i_0}^{(j)}$ と $G_{\tau' i'_0}^{(j')}$ は共通な節点をもたない。また$G_{\tau i_0}^{(j)}$ と G_{00} も $f(n_0^{(j)})$ を除いて共通節点をもたない。

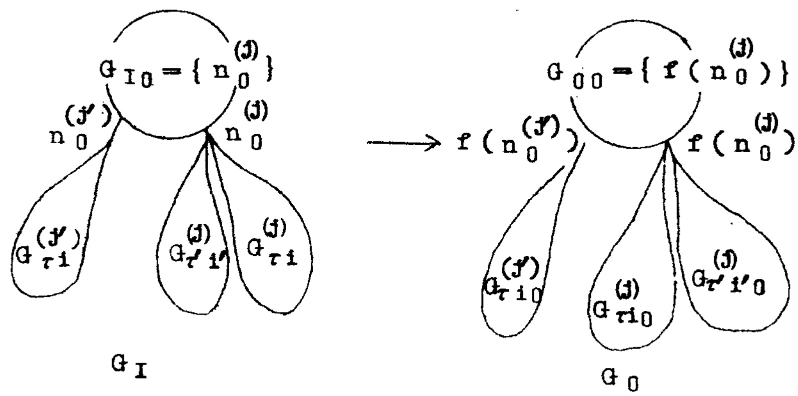


図 4.1 6　補題 4.8 の説明

証明は、$G_{\tau i}^{(j)}$，$G_{\tau i_0}^{(j)}$の定義、前にえらんだ prefix set の性質と補題4.7から簡単にわかる。$G_{\tau i}^{(j)}$ が$n_0^{(j)}$を除いて互に共通節点をもたないこと。G_M がG_I からG_0 への1対1対応を与えていることから容易に、$G_{\tau i}^{(j)} \sim G_{\tau i_0}^{(j)}$。ゆえに(2)より$G_{Ai} \sim G_{\tau i_0}^{(j)}$。帰納法の仮定より

$$S(G_{Ai}) = S(G_{\tau i_0}^{(j)}) \qquad (3)$$

G_0 は上述より、G_{00} の各節点nにG_{00} との共通節点nを除いて互いに共通節点をもたない$G_{\tau i_0}^{(j)}$、（τはすべて、jは$n_0^{(j)} = f(n)$をみたす全体、$1 \leq i \leq \gamma$）がぶら下つた形をしている。可能な（j，τ）の総数を$\nu(n)$とする。またG'_A をG_A の始点が self-loop をもつとき、それを除いたグラフと1−同形とする。G'_A の始点をn_t とかく。G_{00} の各節点nから$\nu(n)$本の枝をG'_A のn_t に結んで得られるグラフをG'_0 とする。(3)と上述のG_0の形からG'_0 とG_0 は同じ標準形をもつ。✿（標準化の過程はどの操作を一部

✿ G'_Aと$G_{\tau i_0}^{(j)}$ との標準形は等しいから、変形し、併合すれば同じこと。

さきに実行しても結果は変わらない）。一方、G'_0 において n_t より上の部分グラフを G'_{00} とすると、

補題 4.9：G_A の始点が D_2（D_1）に属するとき、$S(G'_{00})=S_2^1$（S_1^1）

（証明）　G_A の始点が D_2 のとき、G_{I0} のすべての最下位の leaf (graph) が２つ以上の閉路を含む。補題 4.7からこのような leaf に対応する G_{00} の部分グラフが G_{00} の単一閉路よりなる leaf に含まれえないから $S(G'_{00})=S_2^1$，G_A の始点が D_1 のとき、G_{I0} はたゞ１つの閉路を含む。えらんだ prefix set の性質と補題 4.7 から G_{00} もただ１つの閉路よりなることが示され、$S(G'_{00})=S_1^1$。

n_t は消去できるから上の補題❀より、$S(G_A)=S(G_0)$ が成立する。したがつて(3)より $S(G_A)=S(G_B)$。

レベル数が１のときは標準形は S_0^1，S_1^1，S_2^1 のみである。この場合、G_{Ai}，$G_{\tau i}^{(j)}$，$G_{\tau i0}^{(j)}$，G'_A がただ１本の枝よりなる場合であり、G'_{00} は G_0 そのものである。したがつて補題 4.9よりあきらかである。

❀ G_A の始点が self-loop をもたないときはあきらか。

第６章　結　　論

csmについて、条件(1)、(2)をみたす写像が存在するための必要十分条件を導いた。またこの意味で二つのＦＳＬが翻訳可能かどうかにより、ＦＳＬの全てを類別できるが、各類の代表として状態数最小の標準言語が求まることを示した。逆に各ＦＳＬについて標準言語を求めておけば、標準言語の比較のみにより翻訳可能か否かが簡単にわかる。なお翻訳可能なとき、一つの翻訳機を与える能率的なアルゴリズムが求められた。

gsmについて、条件(1)（(1)′）、(2)をみたす写像が存在するための必要十分条件が得られた。とくに中への写像問題について、L_A，L_Bが無限の文章を含むとき、$M(L_A) \subseteqq L_B$で$M(L_A)$が無限であるようなgsmによる写像Ｍがつねに存在することが示された。さらに、gsmにより翻訳可能かどうかによりＦＳＬ全体を類別できるが、Ｌが与えられたとき、それが含まれる類を一意的に代表し、状態数が最小である標準言語を求める簡単な手続きを示した。一般に複雑なＬも構造のきわめて単純な標準言語に帰着できる。これに基づき$L_A \sim L_B$かどうか判定できる。これらに要する手数はこの種の問題にありがちな“パラメータとともに指数的に増大する”ことなく判定法として能率的である。

第２編　ＣＦＬの認識と構文解析

第１章　緒　　論

与えられた文章が文法に適つた文章か否かを判定し、適つている場合にはその文章構造の解析を行うことは、言語の研究のうちでも基本的な問題の一つである。これらは問題の性質上、アルゴリズムの能率および必要記憶の大きさがとくに重要な問題である。このような立場から、記憶のみからではなく計算時間についての議論が問題として取上げられている。たとえばHartmanisとStearns[23]は標準のチューリング機械を考え、入力系列の長さをnとしたとき、その機械による言語の認識について、$T(n)$以下の操作数で、ある言語が認識されるとき、その言語は$T(n)$-recognizableであるという概念を導入した。

Floyd[27]により、ある特別な制限をつけたContext-free言語について、入力系列の長さに比例する手数で終了する能率的な認識、および構文解析の方法が提案されているが、その制限が余りにも厳しいために実用には種々の困難がある。Hennie[28]はある系列の集まりに対してオンラインのチューリング機械で認識される際の計算時間についてよい下限を示した。またContext-free言語、とくにプログラミング言語を対象とした構文解析の方法などについてもいくつか発表されているが、[31]~[35]大部分の方法では必要とする手数が入力系列の長さnとともに指数的に増大する欠点をもつている。そのうち、Kasami[36][37]は一般のＣＦＬがn^3に、リニア言語がn^2に比例する手数以内で認識されることを示している。

本編第３章では文献(38)(39)にもとづき一般のＣＦＬについて、第５章では（メター）リニア言語をその一部として含むＣＦＬのある部分族について能率

よく認識されることを示し、第4章では文献(38)(39)にもとづき能率的な構文解析の方法を計算時間と記憶の大きさとの関係から述べる。※また文献(40)にもとづき、コンテクスト・センシティブ（context-sensitive）言語の認識について述べる。

※ Younger(41)が全く独立にチューリング機械でワーキングテープ4本を用いて、n^3−recognizable であるなど類似した結果を発表している。

第２章　諸　定　義

第２編に直接関係ある定義をまとめておく。[❀]

句構造文法（ phrase structure grammar ）は、つぎのような順序づけられた４組の文字 $G=(V_N, V_T, P, S)$ である。

(i)　V_N は空でない有限個の non-terminal シンボル（ nt －シンボルと略記する）の集合で、 nonterminal vocabulary といわれる。

(ii)　V_T は空でない有限個の terminal シンボル（ t －シンボルと略記する）の集合で、 terminal vocabulary といわれる。

V_N と V_T は共通な要素を持たない。

(iii)　P は空でない有限個の書き替え規則（ルール）$\varphi\rightarrow\psi$ の集合である。たゞし　$\varphi\in V_N^*-\{\Lambda\}$，$\psi\in(V_N\cup V_T)^*-\{\Lambda\}$ である。

(iv)　$S\in V_N$ は initial シンボル（ i －シンボルと略記する）を表わす。

なお $V=V_N\cup V_T$ と書くこともある。

$G=(V_N, V_T, P, S)$ は句構造文法とする。$\varphi_1, \varphi_2\in V^*$ に対して、$\varphi_1=w_1uw_2$，$\varphi_2=w_1vw_2$ でかつ $u\rightarrow v$ がルール名 g を持つ P の要素であるような $w_1, w_2, u, v\in V^*$ が存在するとき

$$\varphi_1 \xrightarrow{g} \varphi_2$$

と書く。$\varphi, \psi\in V^*$ に対し、$\varphi=\psi$ または、$\varphi=\varphi_0$，$\varphi_r=\psi$ で $\varphi_i\xrightarrow{g_{i+1}}\varphi_{i+1}$ であるような系列 $\varphi_0, \varphi_1, \cdots, \varphi_r$ が存在するとき

$$\varphi \overset{G}{\Rightarrow} \psi \qquad (\text{または}\quad \varphi\Rightarrow\psi\ )$$ [❀❀]

と書く。とくに $\varphi_0=S$ で $\varphi_r\in V_T^*$ のときに限り系列 $\varphi_0, \varphi_1, \cdots, \varphi_r$ は

❀　第１編第２章で述べたものは一部を除き省略する。

❀❀　G を特に注意する必要のないとき、このようにかく。

derivation sequence（d-seq と略記する）という。

$G=(V_N, V_T, P, S)$ が句構造文法であるとき、

$$L(G)=\{x\in V^* \mid S\overset{G}{\rightarrow}x\}$$

は句構造言語（phrase structure language）と言いL(G)はGにより生成されると言う。あるいは、GはL(G)を生成すると言う。

句構造文法 $G=(V_N, V_T, P, S)$ において、P の各要素が、$w_1Xw_2\rightarrow w_1xw_2$ の形のみで、$X\in V_N$, w_1, $w_2\in V_N^*$, $x\in V^*-\{\Lambda\}$ のとき、Gは context-sensitive 文法❀（CSGと略記する）と言う。CSGにより生成される言語は、context-sensitive 言語（CSLと略記する）と言われる。

句構造文法 $G=(V_N, V_T, P, S)$ において、P の各要素が $X\rightarrow x$ の形のみで $X\in V_N$, $x\in V^*-\{\Lambda\}$ のとき、Gは context-free 文法（CFGと略記する）という。CSGにより生成される言語は context-free 言語（CFLと略記する）といわれる。

句構造文法 $G=(V_N, V_T, P, S)$ において P の各要素が

$$X\rightarrow uZv \quad , \quad X\rightarrow x$$

$$(X, Z\in V_N, u, v, x\in V_T^*)$$

の形のみのとき、Gはリニア（linear）文法という。また、リニア文法の形のルールの他に、Sを除く任意のシンボルの系列 ψ が存在して

$$S\rightarrow\psi \qquad (\psi\in(V-\{S\})^*-\{\Lambda\})$$

の形のルールが含まれるとき、Gはメタ・リニア（meta-linear）文法という。（メター）リニア文法により生成される言語は（メター）リニア言語といわれる。

❀ context−sensitive の代りに、context−dependent という言葉が使われることがある。[(1)]

句構造文法$G=(V_N, V_T, P, S)$において、Pの各要素が

$$X \longrightarrow aY \quad , \quad X \longrightarrow a$$

$$(X, Y \in V_N, a \in V_T)$$

の形のみの場合、あるいは、

$$X \longrightarrow Ya \quad , \quad X \rightarrow a$$

$$(X, Y \in V_N, a \in V_T)$$

の形のみの場合、Gは有限状態文法（ＦＳＧと略記する）といわれる。ＦＳＧにより生成される言語は有限状態言語（ＦＳＬと略記する）といわれる。❀

句構造文法のルール$g : \varphi \rightarrow \psi$において、$\ell(g)$，$r(g)$によりそれぞれ$\varphi$，$\psi$を表わすものとする。とくに$\psi = AB\ (A, B \in V)$のとき、$r_1(g)$，$r_2(g)$により、それぞれシンボル$A$，$B$を表わす。

オンラインのチューリング機械は有限個の状態をとりうる制御部と、１方向にのみ動く入力テープと出力テープ、および２方向に動くワーキングテープ（一般には複数本）をもつ。入力テープの各コマ（square）は入力アルファベットＡの任意のシンボルを含み、出力テープの各コマは出力シンボル“１”または“０”である。またワーキングテープの各コマは有限なアルファベットＫ上の任意のシンボルを含む。機械は最初、制御部で或る状態（初期状態）にあり、ワーキングテープの各コマはブランクであり、入力テープ上にはA^*のある系列があり、その読取りヘッドはその系列の左端のシンボルを読むものとする。この系列は入力系列といわれる。機械の動作は、制御部の状態と入力テープ上およびワーキングテープ上で読みとられるシンボルとから決定され、それらはワーキング上に書き加えること、入力テープを１コマ左に動かすこと、

❀ 第１編、第２章の定義と矛盾しないことは、ｎｔ－シンボルを状態に、ｔ－シンボルを状態間の入力シンボルに対応させれば状態図が得られることから簡単にわかる。

ワーキングテープを動かすこと（方向およびコマ数は任意）、状態をかえることあるいは、出力テープ上に出力を書くことからなる。

オンラインのチューリング機械MがA^*のある部分集合Lに対して、A上の任意の入力系列について最初のn個の入力シンボルの系列がLに含まれるとき、出力テープ上に第n番目の出力として"1"を書き、含まれないとき"0"を書くとき、またそのときに限り、チューリング機械MはLを認識するという。さらに、Lを認識するオンラインチューリング機械が存在し、かつ任意の入力系列に対して、T(n)以下の操作でn番目の出力シンボルを出すとき、かつそのときに限り，LはHartmanis-Stearns の意味において、T(n)- recognizableであるという。[23]

第 3 章　CFLの認識

§ 3.1 主　定　理

本章ではKasami[36]において、スタンダード形の文法[42]を仲介として証明された定理をCocke の構造解析のアルゴリズムと同じ方法[43]によるノーマル形の文法[1] を仲介としてより簡単に証明する。

定理 3.1： 一般のCFLは2本のワーキングテープをもつオンラインのチューリング機械により、Hartmanis-Stearns の意味において、n^3-recognizable である。

（証明の概略）　CFG　$G=(V_N, V_T, P, S)$ において、ルールが、

$$X \rightarrow YZ$$

$$X \rightarrow a$$

$$(X, Y, Z \in V_N \quad , \quad a \in V_T)$$

なる形のみからなるとき、Gはノーマル形であるという。一般に任意のCFGはノーマル形に表わし得ることが知られている[1]。以下では一般性を失なうことなく、CFGはノーマル形であるものとする。

入力系列 $\alpha = a_1, a_2, \cdots a_n$ において、k 番目の入力シンボルを a_k　V_T とするとき、

$$N(i, k) = \{ Y \mid Y \rightarrow a_i a_{i+1} \cdots a_k , Y \in V_N \}$$

$$(1 \leq i \leq k)$$

と書くことにする。このとき文法Gにより生成される言語Lの定義より $S \in N(1, k)$ のとき、かつそのときに限り入力系列 α がLに含まれる。

また、

$$Y \rightarrow a_i a_{i+1} \cdots a_k \qquad (i+1 \leq k)$$

ならば、

$$X_1 \to a_i a_{i+1} \cdot\cdot\cdot\cdot a_j$$

$$X_2 \to a_{j+1} \cdot\cdot\cdot\cdot a_k$$

$$(i \leqq j < k)$$

でかつ

$$Y \to X_1 X_2 \quad P$$

なるX_1，X_2が存在する。したがつて、

$$N(i, k) = \bigcup_{j=i}^{k-1} N_j(i, k)$$

$$N_j(i, k) = \{Y\}\, Y \to X_1 X_2 \quad P, X_1 \quad N(i, j), X_2 \quad N(j+1, k)\}$$

と書くことができる。このとき、任意のkについて、$N(1, k)$を求めるにはすべての$N(i, k')$（$1 \leqq i \leqq k' < k$）と$N(k, k)$とが必要であり、逆にそれらがわかれば各j（$i \leqq j \leqq k$）について$N_j(k, k)$，$N_j(k-1, k)$，・・・，$N_j(1, k)$が順次求められ、その結果$N(k, k)$，$N(k-1, k)$，・・・，$N(1, k)$が求められる。

チューリング機械で上の操作を実行するには$N(i, k')$をテープ上一つの区画❀に貯えるものとすると、k^2に比例する長さのテープ$T^{(1)}$を用意する必要がある。このテープの内容と$N(k, k)$とから$j = k-1$から始めて$i = 1$まで順次$N_j(k, k)$，$N_j(k-1, k)$，・・・，$N_j(1, k)$を求める手数を小さくするためには、このテープの各区画への参照の回数を小さくすればよい。したがつてjに関する中間結果を貯えるテープ$T^{(2)}$を準備し、テープ$T^{(1)}$の各区画への参照は一度とし、その区画の内容を必要とする全ての操作をその参照中に行う様にする。したがつて中間結果用のテープ$T^{(2)}$上

❀ §3.2におけるチューリング機械の構成では区画に$N(i, k')$および$i = k'$のときに限り、$a_{k'}$をも貯える。

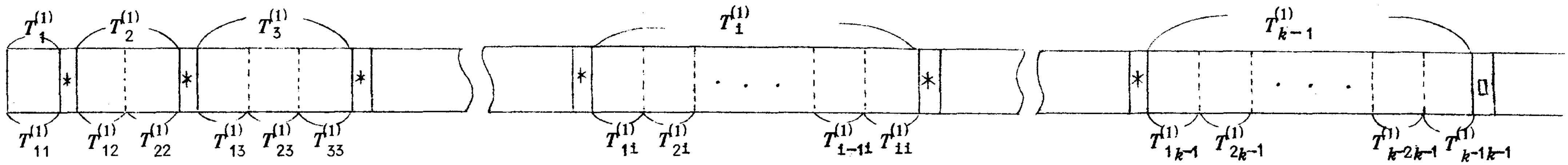
には j が $k-1$, $k-2$, ・・・, 1 の夫々の場合が順次書き加えられていく。たとえば区画 $T_i^{(2)}$ には $N_{k-1}(k-i+1, k)$, $N_{k-2}(k-i+1, k)$, ・・・, $N_{k-i+1}(k-i+1, k)$ が順次書き加えられる。その結果

$$N(k-i+1, k) = \bigcup_{j=k-i+1}^{k-1} N_j(k-i+1, k)$$

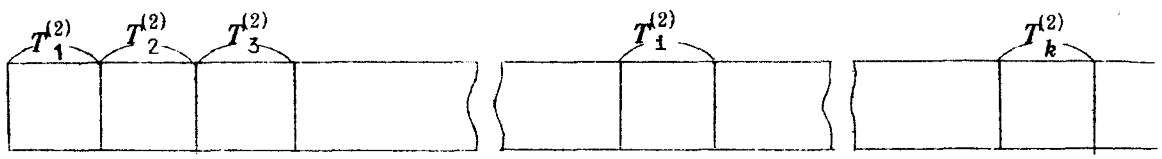
が $T_i^{(2)}$ 上に得られる。この操作により得られた各 $T_i^{(2)}$ の内容を $T^{(1)}$ 上に移すことにより $T^{(1)}$ 上に望む結果 $N(k, k)$, $N(k-1, k)$, ・・・, $N(1, k)$ が得られる。以上の操作では、k 番目の入力シンボル a_k を読み $N(k, k)$ を求めてから $N(1, k)$ を求める手数の上限は、k^2 に比例する長さのテープ $T^{(1)}$ 上の各区画を1回参照するだけであることから $C_1 k^2$ である。たゞし C_1 は k に無関係な定数である。この間のテープ $T^{(2)}$ 上の各区画は $i(<k)$ 回づつ参照されているが明らかに $T^{(2)}$ は k に比例する長さであるから問題はない。したがつて入力系列の長さが n のとき、最初のシンボル a_1 が読みこまれてから、n 番目の出力を出すのに必要な手数の上限は $C_2 n^3$ である。ただし C_2 は n に無関係な定数である。

§ 3.2　定理 3.1の証明

ワーキングテープ $T^{(1)}$ と $T^{(2)}$ をもち、それぞれは右半無限とする。$T^{(1)}$ は $T_1^{(1)}$, $T_2^{(1)}$, ・・・, $T_k^{(1)}$, ・・・と順次区画に分割されていくものとし、各区画の間には区切りの印＊をつけていくものとする。さらに各 $T_k^{(1)}$ は等長の $T_{1k}^{(1)}$, $T_{2k}^{(1)}$, ・・・, $T_{ik}^{(1)}$, ・・・, $T_{kk}^{(1)}$ と順次細分され、各 $T_{ik}^{(1)}$ には t－シンボルおよび $N(i, R)$ なる n t．シンボルの集合 V_N のある部分集合、および特別の印□、△、○が同時に書き込めるものとする。$T^{(2)}$ は $T_1^{(2)}$, $T_2^{(2)}$, ・・・, $T_k^{(2)}$, ・・・と分割していくものとする。制御部には V_N の任意



ワーキングテープ　$T^{(1)}$



ワーキングテープ　$T^{(2)}$

図3.1　定理3.1の証明のチューリング機械のワーキングテープ

の部分集合を貯え得る有限な記憶Mがあるものとする。〔$T^{(1)}_{ik}$〕，〔$T^{(2)}_{k}$〕、および〔M〕でそれぞれ$T^{(1)}_{ik}$，$T^{(2)}_{k}$およびMに貯えられたシンボルの集合を表わすものとすると、一般に〔$T^{(1)}_{ik}$〕はN(i，k)とt－シンボルであり、〔$T^{(2)}_{k}$〕および〔M〕は中間結果となる。なお記号□は２区画間につけられる＊を書くためのもので、k番目の入力シンボルが読まれてk番目の出力を出すための操作が行われているとき、区画$T^{(1)}_{k-1}$の右側に書き込まれており、その出力を出すとき＊におきかえられる。記号△は$T^{(2)}$上でN_j(i，k')を求めたあと、N_{j+1}(i，k')を求めるとき、$T^{(2)}$上のどの区画から書き加え始めるかを示すためのもので、j＝iなる区画につけられる。記号○はk番目の出力を出すための操作が行われているとき、テープ$T^{(2)}$の区画$T^{(2)}_{k}$内に書かれるもので$T^{(2)}$の右端の区画を示すためのものである。

このチューリング機械の動作は次のように進行する。

（１）　$T^{(1)}$のヘッドH$_1$および$T^{(2)}$のヘッドH$_2$は左端のコマにあるものとする。$T^{(1)}$および$T^{(2)}$のコマはすべてブランク(blank)であるとする。最初の入力シンボルa$_1$を入力テープから読み込み、a$_1$と、文法GでY→a$_1$であるようなすべてのルールのnt－シンボルYを$T^{(1)}_{1}=T^{(1)}_{11}$に書く。したがつて、

$$〔T^{(1)}_{11}〕=N(1,1)\quad \{a_1\}$$

になる。$T^{(1)}_{11}$にi－シンボルSが表われゝば、出力テープに出力の最初として"１"を、なければ"０"を書く。$T^{(1)}_{11}$の右隣りのコマに区切り印のために　を書く。テープ$T^{(1)}$のヘッドH$_1$を$T^{(1)}_{11}$上に、テープ$T^{(2)}$のヘッドH$_2$は$T^{(2)}_{1}$に動かす。

（２）　入力テープより最初のk－１個のシンボルa$_1$a$_2$・・・a$_{k-1}$が既に読み込まれ、

$$[T^{(1)}_{ik'}] = N(i, k) \qquad (1 \leqq i \leqq k' < k)$$

とし、$T^{(1)}_{k'-1}$ と $T^{(1)}_{k'}$ ($2 \leqq k' \leqq k$) の間のコマに区切り印＊が書かれていると仮定する。さらに $T^{(1)}_{k-1}$ の右隣りに特別の記号□が書かれているものとする。このとき $T^{(1)}$ のヘッド H_1 は $T^{(1)}_{k-1\,k-1}$ 上に、$T^{(2)}$ のヘッド H_2 は $T^{(2)}_1$ 上にあるとする。

(2.1) k 番目の入力シンボル a_k を入力テープから読みとり、a_k と、文法Gで $Y \rightarrow a_k$ であるようなすべてのルールの nt－シンボルYと $T^{(2)}_1$ をMに書き込む。したがつて

$$[T^{(2)}_1] = [M] = N(k, k) \quad \{a_k\}$$

となる。H_2 を $T^{(2)}_2$ へ動かし、記号△を書く。H_1 で $T^{(1)}_{k-1\,k-1}$ を読み、$T^{(1)}_{k-1\,k-1}$ 内の X_1 とM内の X_2 について、Gで $Y \rightarrow X_1 X_2$ であるようなすべてのルールの nt－シンボルYを $T^{(2)}_2$ に書き込む。したがつて、

$$[T^{(2)}_2] = N(k-1, k) \qquad (\text{△は除く})$$

となる。H_1 を $T^{(1)}_{k-2\,k-1}$ に H_2 を $T^{(2)}_3$ に動かす。

(2.2) $T^{(2)}$ のヘッド H_2 が $T^{(2)}_i$ ($2 \leqq i < k$) にあり、

$$[M] = N(k, k) \cup \{a_k\}$$

であるとし、$T^{(1)}$ のヘッド H_1 は $T^{(1)}_{k+1-i\,k-1}$ にあるとする。このとき $T^{(1)}_{k+1-i-k}$ 内の X_1 とM内の X_2 について、Gで $Y \rightarrow X_1 X_2$ であるようなすべてのルールの nt－シンボルYを $T^{(2)}_i$ に書き込む。したがつて

$$[T^{(2)}_i] = N(k+1-i, k)$$

H_1 を1区画左へ、H_2 を1区画右へ動かす。※

(2.3) $T^{(2)}$ のヘッド H_2 が $T^{(2)}_{i-1}$ ($2 \leqq i \leqq k$) にあり、

$$[M] = N(k', k) \qquad (1 \leqq k' < k)$$

※ $T^{(2)}$ 上でその右側に区画が作られてない場合も右に動かす。

$[M] = N(k', k) \cup \{a_k\} \qquad (k' = k)$

であるとする。

(2.3.1) H_1 が$T^{(1)}_{k-ik'}$ にあるとき、$T^{(1)}_{k-ik'}$ 内のX_1 とM内のX_2 について、Gで$Y \rightarrow X_1 X_2$ であるようなすべてのルールの nt－シンボルYを$T^{(2)}_i$ に書き加える。すなわち、$T^{(2)}_i$ にあらたに$N_{k'}(k-i+1, k)$ が書き加えられる。H_1 を1区画左へ、H_2 を1区画右へ動かす。*

(2.3.2) H_1 が*を読むと、H_2 を左側の区画$T^{(2)}_k$ に戻し、○を附す。(既に○があれば必要はない)。H_2 を△のある区画まで動かし、Mの内容をその区画の内容(△は除く)でおきかえ、△を消す。H_2 を1区画右へ動かし、そこに△を書き加える。H_1 を1区画左へ動かす。

(2.3.3) H_1 が$T^{(1)}$ の左端になると、H_1 を□のある位置に動かし、□を*に書きかえる。さらにH_1 を1コマ右に動かす。H_2 を○のある区画に動かし、その区画の内容を読みとり、H_1 で$T^{(1)}$ 上に順次書き込む($T^{(2)}$ 上の○のある区画に等しいコマ数が必要であり、それだけを1区画として$T^{(1)}_{1k}$ が求められたことになる)。すなわち

$[T^{(1)}_{1k}] = [T^{(2)}_k]$

である。○のある区画すなわち$T^{(2)}_k$ の内容を消去する。H_2 を1区画左へ、H_1 を右へ動かし、$T^{(2)}$ でH_2 が読みとる$T^{(2)}_{k-1}$ の内容を$T^{(1)}$ 上に書き込む。$T^{(2)}_{k-1}$ の内容を消去し、H_2 を1区画左へ、H_1 をさらに右へ動かす。こうして$T^{(2)}$ 上の内容をすべて$T^{(1)}$ 上に書き、$T^{(2)}$ 上の各コマがブランクになるまで続ける。その結果$T^{(1)}$ には$T^{(1)}_k$ として、$T^{(1)}_{1k}, T^{(1)}_{2k}, \cdots, T^{(1)}_{kk}$ が書かれたことになる。$T^{(1)}_{kk}$ の右隣りのコマに□を書く、H_1 は$T^{(1)}_{kk}$ に、H_2 は$T^{(2)}_1$ におく、$T^{(1)}_{1k}$ に i－シンボルSがあれば出力テープに"1"を、な

* $T^{(2)}$上でその右側に区画が作られてない場合も右に動かす。

ければ"0"と書く。

入力系列$\alpha = a_1 a_2 \cdots a_n$ に対しては、上の操作を$k = 1$から$k = n$まで行えば終了する。

第 4 章　ＣＦＬの構文解析

§ 4.1　序　　言

前章、定理 3.1の証明途上、仲介として使つたノーマル文法を利用した能率的な構文解析の方法を導く。❀一般のＣＦＬの認識問題をチューリング機械で実行する場合については前章で述べた通りであるが、この問題を計算機と密接な関係にあるものとして、計算に必要な手数をランダム、アクセス記憶を用いて必要な記憶の大きさとともに求める。

§ 4.2　ランダム、アクセス記憶を用いる認識アルゴリズム

一般にd-seq. はその構造をも表現する目的で、グラフ的に表わすことがある。たとえばderivation tree (d-tree) とよばれるものは、最終端の節点はt－シンボルを、最終端以外の節点はnt－シンボルを表わし、最終端の列は入力系列になつている。枝は適用されたルールgに対し、l(g)からr(g)へ、その向きに描かれたものである。こゝでは一つのd-seq. ある

S　→　A B
A　→　B A
A　→　a
B　→　b

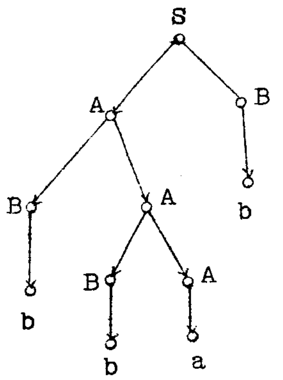
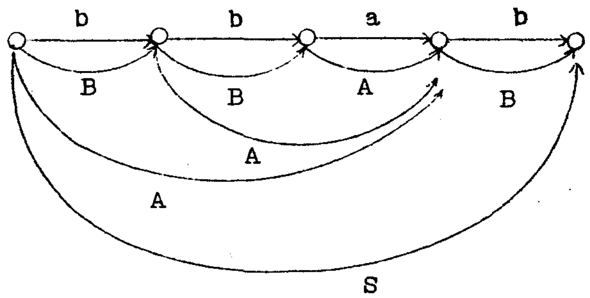


図 4.1　d－treeの例

❀ 構文解析をノーマル文法を使つて、行うことについては、既に知られている。たとえば文献 (43) など。

いは一つのd-treeに対応して、方向性グラフ(directed linear graph)Dを考える。これは入力系列$\alpha = a_1 a_2 \cdots a_n$ のとき、(n+1)個の節点$v_0, v_1 \cdots v_n$と、節点v_iからv_j ($1 \leqq i, j \leqq n$)に矢印が向く枝とから成り、各枝はtーシンボルあるいはntーシンボルを名前としても



$\alpha =$ b b a b

図 4. 2　グラフDの例

つ。そのうち入力系列を作るシンボルa_i ($1 \leqq i \leqq n$) は枝$\overrightarrow{v_{i-1}\ v_i}$の一つ[※]に相当し、他の枝はすべてntーシンボルである。さらにgが適用されたルールのとき、γ(g)を枝の名前の系列としてもつ道の両端の節点が名前ℓ(g)をもつ枝で結ばれたものである。

グラフDはd-treeと1対1に対応するが、同時にd-seq.とも1対1に対応し、ある入力系列αに対してd個の異なるd-seq.があるとき、入力系列αに対して、Dに相当するd個の異なるグラフが得られる。したがつて認識を行うことは、任意の入力系列に対して、あらゆる可能なグラフDを求め、その内少くとも一つ、始点v_0から終点v_nに至る枝が存在し、その枝の名前が文法Gのiーシンボルであるとき、かつそのときに限り入力系列がL(G)に含まれるものとすることである。

※　節点v_iから節点v_jに至る矢印を持つ枝を$\overrightarrow{v_i\ v_j}$とかく。また混乱の恐れのないとき、このような枝の集合をも$\overrightarrow{v_i\ v_j}$で表わすことがある。

以上のことより、文法Gから生成される言語L(G)についてつぎのような認識のアルゴリズムが得られる。

入力系列を$\alpha = a_1, a_2 \cdots, a_n$として、t－シンボル$a_i$はi番目の入力シンボルとする。

(1) 節点v_0のみからなるグラフR_0を作る。

(2) グラフR_{i-1}まで作られたとする。R_{i-1}に節点v_iを加え、節点v_{i-1}からv_iに向きを持つ枝で結ぶ。ただしその名前をa_iとする。得られたグラフをR_{i0}とする。

(3) グラフR_{ij-1}（$1 \leq j \leq i$）まで得られたとする。各ルールgについてR_{ij-1}において節点v_{i-j}からv_iに至る道があり、その名前の系列が$\gamma(g)$であるとき、かつそのときに限り、節点v_{i-j}からv_iに名前$\ell(g)$をもつ向きをもつた枝を書き入れる。（このようなルールが2つ以上あれば重複して書く）。得られたグラフをR_{ij}とする。

(4) グラフ$R_i = R_{ii}$を作る。

(5) グラフR_nにおいて始点v_0から終点v_nに至るi－シンボルSを名前としてもつ枝が一つでも存在すれば、入力系列αはL(G)に含まれることになる。そのような枝が存在しなければ、αはL(G)に含まれない。

以上の操作における計算時間の上限は、ランダム・アクセス記憶を使えば、R_{ij}において、各iについてi^2に比例する単位操作を必要とすることから、R_nを求めるには$C_1 n^3$である。こゝでc_1はnに無関係な定数となる。

§ 4.3 構文解析のアルゴリズム

前節に述べた認識のアルゴリズムにより認識されたものとして、全ゆるd-seq.を求めることを考える。したがつてグラフR_nにおいて、v_0からv_nに名前Sをもつ枝が少くとも一つ存在すると仮定する。ただし、認識アルゴリズムではルールgが適用されたときℓ(g)を枝の名前としたが、本節では便宜上ℓ(g)の代りにgを枝の名前として取扱う。この仮定はアルゴリズムに何ら本質的な影響を与えるものではない。

ルールgの右辺γ(g)はさらに、γ_1(g), γ_2(g)で右辺の最初のシンボルおよび2番目のシンボルをそれぞれ表わすものとする。$\gamma_2(g)=\Lambda$をも含める。各枝の名前について、R_nにおけるv_{i-j}からv_iへの枝の名前の集合を、E^i_jで表わし、E^i_jを貯える記憶部分には、$g \in E^i_j$および記号△、○をも貯え得るものとする。PDでpush down storeを表わし、任意のルールgについて、γ_1(g)とγ_2(g)とがそれぞれ枝の名前の系列g′, g″のℓ(g), ℓ(g″)に一致するとき、g″およびg″に関する情報を貯えるものとする。このPDの最初に貯えられた内容を〔PD〕で表わし、σはPDに何も入つていないときの〔PD〕を表わすものとする。さらにα, βはそれぞれ一時記憶のレジスタを表わす。

(1) $n \to i$, $n \to j$とする。

(2) E^i_jの内、△がなく、ℓ(g)（$g \in E^i_j$）がSである枝に△を附し、$d.s.=g$とする。$\gamma_1(g) \to \alpha$, $\gamma_2(g) \to \beta$とする。このような枝が存在しないならば操作は終る。

(3.1) $1 \to k$として、$\alpha = l(g')$, ($g' \in E^k_k$)なるg′を求め、○を附す。ただし、既にg′に△があれば○はつけずにE^k_k内の他の枝の名前についてこの操作を行う。E^k_k内にそのような枝がなければ$k+1 \to k$として、

(3.1)をくり返す。

(3.2)　$\beta = \ell(g'')$ (g'' 　E^{i}_{i-k}) なる g'' を求め△を附し、E^{k}_{k} 内の○を△に書きかえる。ただし g'' に既に△があれば、E^{i}_{i-k} 内の他の枝についてこの操作を行う。E^{i}_{i-k} 内にそのような枝がなければ $k+1 \to k$ とし E^{k}_{k} 内の○を消し (3.1) を行う。γ_2 (g') がなければ(5)を行う。

(4)　$d.s.=g'$，$\gamma_1(g) \to \alpha$，$\gamma_2(g'') \to \beta$ とし、g'' および i，i-k を PD に貯える。便宜上このような i，i－k をそれぞれＩ，Ｊとする。つぎに (3.1) を行う。

(5)　PD を１段上げる。〔PD〕$=\sigma$ ならば求められた $d.s.$ の系列が一つの d-seq. になる。したがつてつぎに異なる d-seq. を求めるために(1)を行う。〔PD〕$\neq\sigma$ ならば、〔PD〕のルール名 g_p を $d.s.$ とし、$\gamma_2(g_p)$ が存在しなければ(5)を行う。存在すれば $\gamma_1(g_p) \to \alpha$，$\gamma_2(g_p) \to \beta$ とし、〔PD〕のルール名 g_p とともに貯えられた数Ｉ，Ｊについて $\mathrm{I} \to i$，$\mathrm{J} \to j$ として (3.1) を行う。

以上の操作は、つぎのことを行つていることであり、あらゆる d-seq. が求められることは簡単にわかる。すなわち入力系列が左から右に並べられた d-tree についていえば、その頂点に当たる i－シンボル S を導くルール g_1 (2つ以上あるときはその内の一つ) から出発して、$\gamma_1(g_1)$ を導くルール g_2 を求めて $\gamma_2(g_1)$ を導くルールを PD に貯えておく。さらに $\gamma_1(g_1)$ を導くルール g_3 を求めて $\gamma_2(g_2)$ を導くルールを PD に貯える。こうして t－シンボル a_1 を右辺とするルールを求めたあと、PD の内容と、グラフ R_n とから t－シンボル a_2 を右辺とするまでのルールが a_1 の場合と同様に求められ、入力系列の n 番目の t－シンボル a_n を右辺とするルールが求められれば、それまでに求められたルール列は一つの d-seq. になつてい

る。一つのd-seq. が求められたあと、i－シンボルSの位置に戻り、△の附されていない枝から再び上のことをくり返すことにより、次に異なるd-seq. が求められる。全てのi－シンボルSの位置（v_0，v_n間の枝）に△が附されれば操作は終り全ての異なるd-seq. が求められたことになる。

上述のアルゴリズムに必要な手数の上限は、d-seq. の一つのルールを求めるのに$C_2 n$ だから一つのd-seq. を求めるには$c_3 n^2$ となり、N個の異なるd-seq. を求めるには、認識アルゴリズムに必要な手数$c_4 n^3$ と加えて、$c_4 n^3 + c_5 n^2 N$である。ここで各c_i はnに無関係な定数である。

上のアルゴリズムにおいては、ランダム・アクセスの記憶を用いて、等しい数からなるある大きさの記憶を各E_j^i に割りあてるものとしている。したがつて各E_j^i 内にはたかだか nt－シンボルの数が貯えられ、$0 \leqq i \leqq n-1$，$1 \leqq j \leqq n$で、且つ各単位記憶の大きさは定められているから、必要な記憶容量はたかだかcn^2 である。

§ 4.4　CSLのある部分族の認識への拡張

ルールが

$$X \rightarrow a \qquad (X \in V_N \ , \ a \in V_T)$$

$$S \rightarrow XY \qquad (S : \text{i－シンボル} ; X , Y \in V_N)$$

の他に

$$YZ \rightarrow UVX \qquad (Y, Z, U, V, X \in V_N)$$

なる形のみのCSG　$G_S = (V_N, V_T, P, S)$ から生成されるCSLの集まりを$\mathcal{L}_S$ とする。このとき、$L \in \mathcal{L}_S$ の認識のアルゴリズムは、つぎのようにグラフ的に書くことができる。

a_i　V_T（$1 \leqq i \leqq n$）はi番目の入力シンボルとする。

(1)　節点$v_{00}^{(1)}$ のみからなるグラフk_0 を作る。

(2)　グラフk_{i-1}まで作られたとする。k_{i-1}に節点$v_{i0}^{(k_0)}$（$k_0=1$）を加え、節点$v_{i-10}^{(k_1)}$ から$v_{i0}^{(k_0)}$ に向きを持つ枝を結ぶ。その枝の名前をA_iとする。たゞし、$A_i \to a_i$ はPの一つの要素とする。得られたグラフをk_{i0} とする。

(3)　グラフ　$_{ij-1}$（$1 \leqq j < i$，$3 \leqq i$）まで作られたとする。各ルールgについて$v_{ij-1}^{k''_{j-1}}$（$k''_{j-1} \leqq k_{j-1} \leqq 1, 2, \cdots$）を終端点とする長さ3の道があり、その道の名前の系列が$\gamma(g)$であるとき、かつそのときに限り、その道の初端点から終端点$v_{ij-1}^{k_{j-1}}$ に同じ向きをもち、$\ell(g)$を名前としてもつ枝（ルールの形から明らかに枝は2本になる）と、一つの節点を書き込む。こうして作られた節点を任意に、順次$v_{ij}^{k_j}$（$k_j=1, 2, \cdots$）と名付ける。

このような操作を、可能な限り続ける。（上の操作を行つたために新たに作られた節点についても可能な限り続けるものとする）。ただし同じ道についてはくり返すことはしない。この結果得られたグラフをk_{ij} とする。

(4)　$k_i = k_{i0}$　　　ただし　$i = 1, 2$

　　　$= k_{i1}$　　　〃　　$i = 3$

　　　$= k_{ii-3}$　　〃　　$i \geqq 4$

とする。

以上の操作の結果、グラフk_n において、始点$v_{00}^{(1)}$ から終点$v_{0n}^{(1)}$ に至るiーシンボルSを名前としてもつ枝が一つでも存在すれば、入力系列αは$\mathcal{L}_S$ に含まれることになり、そのような枝がなければ、αは$\mathcal{L}_S$ に含まれない。このことはCFLの認識の際と同じように、d-treeと1対1に対応するような、全てのグラフを同時に求めていることであることから明らかであ

ろう。

以上の認識アルゴリズムは$L \in \mathcal{L}_S$にのみ適用できるが、さらに

$$X \rightarrow YZ \qquad (X, Y, Z \in V_N)$$

の形のルールが含まれるとき、第3章の方法を並行して行えば認識可能なことは簡単にわかる。なおKuroda[45]によれば、任意のorder n(≧3)❀のCSGに強等価なorder (n−1)のCSGが存在することが示されている。したがつてorder 3の任意のCSGから、両辺の系列の長さがともに3のルール、および2のルールを除いたCSGにより生成される言語の集まりが$\mathcal{L}_S$になつている。

§4.5　結　　言

一般のCFLの認識および構文解析は入力の長さnとしたときc_1n^2の大きさのランダム・アクセス記憶を用いて計算時間の上限は$c_2n^3+c_3n^2N$で終了することが示された。こゝでc_iはnに無関係な定数とする。

4.3節のアルゴリズムで記憶の方式を、実際問題ではE_j^iの各i，jについて現われる枝の数が$c_4n(n-1)/2$より極めて少い場合も稀ではないことを考えて、単位記憶に各i，j，k，をそのまゝ貯えるようにすれば、明らかに記憶はc_5n^3必要で計算のための手数の上限は$c_6n^3+c_7n\log n\cdot N$❀❀に減少されるであろう。これらの問題については、個々の問題の特徴を考慮すれば、記憶容量と、手数の減少が期待される。

❀　CSGにおいて、ルールに表われる系列の長さが、どのルールをとつてもn以下のとき、この文法はorder nであるという。[45]

❀❀　単位記憶内に$1 \leqq i, j, k \leqq n$が貯えられるために$\log n$ nの項が表われる。

4.4節では、ノーマル文法の概念を拡張することにより、ＣＳＬのある部分族についての認識アルゴリズムを示した。しかしながら一般にＣＳＬの認識問題は、指数的増大はまぬがれ得ないであろう。❀ したがつて、指数関数の形が今後問題にされるべきであると考えられる。

❀ ＣＳＬはnondeterministic linear boundedオートマトンで受理されることが知られている。(45) したがつて、その状態数A、入力系列の長さnのとき、少くともA^nのconfigurationを覚悟する必要がある。

第5章　CFLのある部分族の認識

§5.1　主　定　理

本章ではCFLのある部分族$\mathcal{L}$に含まれる言語の認識について考える。Kasami[37]はリニア言語がHartmanis-Stearnsの意味においてn^2-recognizableであることを示しているが$\mathcal{L}$はリニア言語、さらにメタ・リニア言語を部分族として含む。

アルファベット$V_{T0}=\{A_1, A_2, \cdots, A_{t_0}\}$をt－シンボルの集合とし、Sをi－シンボル、$V_{N0}$をnt－シンボルの集合とするFSG　G_0が与えられたとき、i－シンボルをA_p（$1\leqq p\leqq t_0$）とするt_0個のリニア文法$G_p=\{V_{N_p}, V_{T_p}, P_p, A_p\}$（$1\leqq p\leqq t_0$）を考える。こゝで議論を簡単にするため文法$G_p$の$A_p$以外のnt－シンボルは$V_{N0}$および$G_{p'}$（$p\neq p'$, $1\leqq p'\leqq t_0$）のnt－シンボルとも異なるものとする。このとき

$$L=\{\psi \mid S\overset{G_0}{\Rightarrow}\varphi,\ \varphi\in V_{T_0}^*,\ \varphi\overset{G_p}{\Rightarrow}\psi,\ \psi\in V_{T_p}^*\}$$

なるLの集合を$\mathcal{L}$と定義する。$\mathcal{L}$はリニア言語、さらにはメタ・リニア言語を部分族として含むことが定義から明らかである。この$L\in\mathcal{L}$についてつぎの定理が成立する。

定理5.1：任意の$L\in\mathcal{L}$は1本のワーキングテープをもつオンラインのチューリング機械により、Hartmanis-Stearnsの意味においてn^2-recognizableである。

（証明の概略）まず一般性を失うことなく、FSG　G_0の各ルールはすべて

$$Y \rightarrow XA$$

$Y \rightarrow a$

$(Y, X \in V_{N0}, a \in V_{T0})$

なる形からなるものとでき、同様に各リニア文法 G_p について、各ルールは

$Y \rightarrow X a$

$Y \rightarrow a X$

$Y \rightarrow a$

$(Y, X \in V_{Np}, a \in V_{Tp}; 1 \leqq P \leqq t_0)$

のいずれかであるものとできる。❀ なお簡単のため

$V_{Tp} = V_{Tp'} = V_T \qquad (p \neq p', 1 \leqq p \leqq t_0, 1 \leqq p' \leqq t_0)$

であるとする。

定理3.1の証明と同様に、k 番目の入力シンボルを $a_k \in V_T$ として、

$N(i, k) = \{Y \mid Y \overset{G_p}{\Rightarrow} a_i a_{i+1} \cdots a_k, Y \in V_{Np}, 1 \leqq p \leqq t_0\}$

$(1 \leqq i \leqq k)$

と書くことにする。$Y \overset{G_p}{\Rightarrow} a_i a_{i+1} \cdots a_k$ ならば、

$X \overset{G_p}{\Rightarrow} a_i \cdots a_{k-1}, \quad Y \rightarrow X a_k \subset P$

または

$X \overset{G_p}{\Rightarrow} a_{i+1} \cdots a_k, \quad Y \rightarrow a_i X \in P$

したがつて、$1 \leqq i < k$ について

$N(i, k) = N_1(i, k) \cup N_2(i, k)$

$N_1(i, k) = \{Y \mid Y \rightarrow X a_k \in P, X \in N(i, k-1)\}$

$N_2(i, k) = \{Y \mid Y \rightarrow a_i X \in P, X \in N(i+1, k)\}$

と書くことができ、すべての $N(i, k-1)$ $(1 \leqq i \leqq k-1)$ と $N(k, k)$ とがわかれば、$N(k-1, k), N(k-2, k), \cdots, N(1, k)$ が求められ

❀ 一般のCFLのノーマル形の文法の概念から明らかである。

る。つぎに、

$$M(k)=\{Y \mid Y \overset{G_{P}'}{\Rightarrow} a_1 a_2 \cdots a_k,\ Y \in V_{N0}\}$$

こゝで$Y \overset{G_{P}'}{\Rightarrow} \varphi$は$Y \overset{G_0}{\Rightarrow} \varphi_1$，$\varphi_1 \overset{G_P}{\Rightarrow} \varphi$とする。このとき言語$L \in \mathcal{L}$の定義より$S \in M(k)$のとき、かつそのときに限り入力系列が$L$に含まれる。$Y \overset{G_{P}'}{\Rightarrow} a_1 a_2 \cdots a_k$（$Y \in V_{N0}$）ならば、ある$i$について

$$X \overset{G_0}{\Rightarrow} a_1 a_2 \cdots a_i \qquad (X \in V_{N0})$$

$$Y \rightarrow XA_P \in P_0$$

$$A_P \overset{G_P}{\Rightarrow} a_{i+1} \cdots a_k$$

である。したがつて$M(i)$と$N(i+1,k)$（$1 \leqq i \leqq k-2$）とがわかれば$M(k)$が求められる。この手続きをチューリング機械で実行するには、定理3.1の一般のCFLの場合と異り（$k-1$）番目の入力シンボルまで読みこんだとき、すべてのiについて$N(i,k')$（$1 \leqq i \leqq k' < k$）を貯えておく必要はなく、a_iと$N(i,k-1)$でよい。しかしその他に今の場合には定理3.1で必要なかつた$M(i)$（$1 \leqq i \leqq k-1$）を貯える必要がある。これらを貯えるテープTは明らかにkに比例する長さを持てばよい。計算の手数を小さくするために、各i（$1 \leqq i \leqq k-1$）についての処理をiに関する情報、すなわちa_iと$N(i,k-1)$および$M(i)$の参照中に行う様にすればよい。したがつて、k番目の入力シンボルが読みこまれたとき、各i（$1 \leqq i \leqq k-1$）について、$i=k-1$からはじめて$i=1$まで$N(i,k-1)$とa_kとから$N(i,k)$を求め、さらにその$N(i,k)$と$M(i-1)$とから$M_i(k)$を求める。こゝで

$$M_i(k)=\{Y \mid Y \rightarrow XA,\ X \in M(i),\ A \in N(i+1,k)\}$$

とする。したがつて$M_i(k)$の意味から、$M(k)$が（$k-1$）回に分けて順次求められることを示す。たゞし$M(k)$の要素の数はたかだかG_0のntーシン

ボルの数であるから、$M_i(k)$ と$M_{i'}(k)$（$i \neq i'$）に同じシンボルを含めばそのシンボルをあらたに考える必要はない。以上の操作で、k番目の入力シンボルa_k を読み$M(k)$を求める手数の上限はk に比例する長さのテープT上を一回参照するだけであることからC_3k である。たゞしC_3 はkに無関係な定数である。したがつて入力系列の長さがnのとき、最初のシンボルa_1を読み込んでからn番目の出力を出すのに必要な手数の上限はC_4n^2 である。ただしC_4 はnに無関係な定数である。

$\mathcal{L}$の定義より、（メタ－）リニア言語について文献 (36)で示された定理が、定理2から系として導びかれることは明らかである。

系：（メタ－）リニア言語は1本のワーキングテープをもつオンラインのチューリング機械により、Hartmanis － Stearns の意味においてn^2－recognizable である。

たとえばt－シンボルを0，1，sとして

$$s b_1 s b_2 s \cdots s b_m s$$

$$(b_i \in \{0, 1\}^* \; ; \; 1 \leq i \leq m)$$

でかつ、$1 \leq i_1 < i_2 < \cdots < i_k < j_1 < j_2 < \cdots < j_\ell \leq m$

について$b_{i_1} b_{i_2} \cdots b_{i_k}$（$k \geq 1$）が$b_{j_1} b_{j_2} \cdots b_{j_\ell}$（$\ell \geq 1$） の順序を逆にしたものであるような系列の集合$L_Y$※は、Younger[41]が$n^3$ に比例する手数以下で終了する認識アルゴリズムが知られていない例として示したものであるが、つぎのようにリニア文法G_Y で生成されることより、n^2－recognizable であることがわかる。※※

※ $L_Y = \{ S\varphi_1 S\varphi_2 S\varphi_3 SR(\varphi_2) S\varphi_4 S \mid \varphi_1, \varphi_2, \varphi_3, \varphi_4 \in \{0, 1, S\}^* \}$とかける。ただし$R(\varphi_2)$は$\varphi_2$を逆にした系列を示す。

※※ 脚注のL_Yの記号を使えばXはφ_1，φ_4に、Yはφ_2，$R(\varphi_2)$に、そしてZはφ_3に直接関連するシンボルである。

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

$$V_N = \{ S, X, Y, Z \}$$

$$V_T = \{ 0, 1, s \}$$

$$
\begin{array}{llll}
P \; ; & S \rightarrow sXs, & X \rightarrow 0X, & X \rightarrow 1X \\
& X \rightarrow sX, & X \rightarrow X0, & X \rightarrow X1 \\
& X \rightarrow Xs, & X \rightarrow sYs, & \\
& Y \rightarrow 0Y0, & Y \rightarrow 1Y1, & Y \rightarrow sYs \\
& Y \rightarrow sZs, & & \\
& Z \rightarrow 0Z, & Z \rightarrow 1Z, & Z \rightarrow sZ \\
& Z \rightarrow Z0, & Z \rightarrow Z1, & Z \rightarrow Zs \\
& Z \rightarrow 0, & Z \rightarrow 1, & Z \rightarrow s
\end{array}
$$

§ 5.2　定理 5.1 の証明

ワーキング・テープ T をもち、T は右半無限で等長の区画 T_1, T_2, $\cdots$, T_i, $\cdots$ と分割されていくものとする。各区画 T_i はさらに 2 つに細分 T_{0i} および T_{1i} とする。T_{0i} には $a_i \in V_T$ と FSG G_0 の nt－シンボルの集合が貯えられ、T_{1i} にはリニア文法 G_p ($1 \leq p \leq t_0$) の nt－シンボルが貯えられるものとする。テープ T の各コマは最初ブランクとする。制御部には有限の容量の記憶 M_0, M_1, M_2 および M_3 があるものとする。M_0 は最後に読まれた入力シンボルを、M_1, M_2, M_3 には中間結果が入るものとし、M_1, M_2 に各 V_{Np} ($1 \leq p \leq t_0$) の部分集合、M_3 には V_{N0} の部分集合が入れられる。$[T_{hi}]$ ($h=0, 1$, $i=1, 2 \cdots$) および $[M_h]$ ($h=0, 1, 2, 3$) はそれぞれ T_{hi} および M_h, に貯えられた

シンボルの集合を表わすものとする。なおシンボル k ，i により機械の動作を記述するが、k ははじめから k 個の入力シンボルが読みこまれ、i はヘッドが T_i にあるとき、N（i，k）が求められることを示す。さらに N（i，k）が求められた後、M(k)を求める。この機械の動作はつぎのように進行する。

(1) 入力テープから最初の入力シンボル a_1 を読み、T_{01} に $*a_1$ と FSG G_0 で $Y \to a_1$ であるようなすべてのルールの nt－シンボル Y を書く。さらに各リニア文法 G_p（$1 \leq p \leq t_0$）$Y \to a_1$ であるようなルールについて、すべての nt－シンボル Y を T_{11} に書く。したがつて

$$[T_{11}] = N(1, 1)$$

である。T_{11} が G_0 の i－シンボル S を含めば出力テープに“1”を、含まれなければ“0”を書く。ヘッドを T_{02} に動かす。

(2)
$$[T_{01}] = *a_1$$
$$[T_{0k'}] = a_{k'} \cup M(k'-1) \quad (1 < k' < k)$$
$$[T_{0k}] = M(k-1)$$
$$[T_{1k'}] = N(k', k-1) \quad (1 \leq k' < k)$$

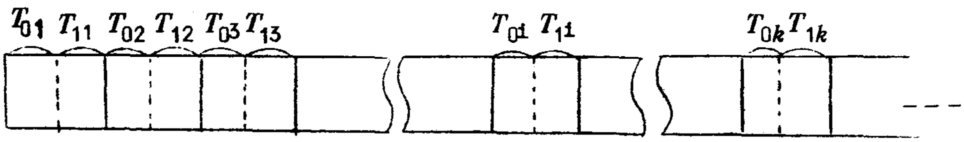


図 5.1　定理 5.1 の証明のチューリング機械のワーキングテープ T

であり、ヘッドがT_{0k}にあるとする。そのときk番目の入力シンボルa_kを入力テープから読み、a_kをM_0とT_{0k}に貯える。文法G_pで$Y \to a_k$であるようなすべてのルールの nt－シンボルYをT_{1k}とM_2に貯える。したがつて

$$[T_{1k}]=[M_2]=N(k, k)$$

である。ヘッドを再びT_{0k}に戻し、$M(k-1)$のX_1とM_2のA_p（A_pがなければ何もしない）について、G_0で$X \to X_1 A_p$であるようなすべてのルールの nt－シンボルXをM_3に貯える。このときの$[M_3]$は$M_{k-1}(k)$である。ヘッドをT_{1k-1}に動かす。

(2.1)　ヘッドがT_{hi}（$h=0, 1, 1 \leqq i < k$）上にあり

$$[T_{1k'}]=N(k', k-1) \qquad (1 \leqq k' \leqq i)$$

$$[T_{1k'}]=N(k', k) \qquad (i < k' \leqq k)$$

$$[M_0]=a_k$$

$$[M_2]=[T_{1\,i+1}]=N(i+1, k)$$

$$[M_3]=\bigcup_{k'=i+1}^{k-1} M_{k'}(k)$$

であるとする。こゝで$[M_3]$の$\bigcup_{k'=i+1}^{k-1} M_{k'}(k)$の意味は各$k'$について$M(k')$と$N(k'+1, k)$とから$M(k)$の一部として求められたシンボルの集合を表わす。

T_{1i}内のY_1について文法G_pで$Y \to Y_1 a_k$であるようなすべてのルールの nt－シンボルYをM_1に貯える。したがつて

$$[M_1]=N_1(i, k)$$

ヘッドをT_{0i}に動かし、T_{0i}のa_iとM_2内のY_1について、文法G_pで$Y \to a_i Y_1$であるようなすべてのルールの nt－シンボルYをM_1にさらに加える。すなわち

$$[M_1] = N_1(i,k) \cup N_2(i,k)$$
$$= N(i,k)$$

である。ヘッドをT_{0i}に動かし、T_{0i}内のM(i−1)のX_1とM_1内のA_p（A_pがなければ何もしない）について、G_0で$X \rightarrow X_1 A_p$であるようなすべてのルールのnt−シンボルXをM_3に加える。T_{0i}に"＊"がなければM_1の内容をM_2とT_{1i}に写し、ヘッドをT_{1i-1}に動かして、（2.1）の操作をi−1について行う。T_{0i}に"＊"があればM_1の内容を$T_{1i}=T_{11}$に写し、ヘッドをT_{1k}の右側の区画に動かし、M_3の内容をその区画に写す。M_3に"S"があれば、出力テープ上にk番目の出力として"1"を、なければ"0"を書く。

入力系列$\alpha = a_1 a_2 \cdots a_n$に対しては、上の操作を$k=1$から$k=n$まで行えば終了する。

第６章　結　　論

一般のＣＦＬが２本のワーキングテープをもつオンラインのチューリング機械で、Hartmanis-Stearns の意味においてn^3-recognizable であること、（メター）リニア言語をその一部に含むＣＦＬのある部分族について、１本のワーキングテープをもつオンラインのチューリング機械、同じくn^3-recognizable であることを示した。

ＣＦＬの構文解析はランダム・アクセス記憶の必要量がc_3n^2 で$c_1n^3+c_2n^2N$ を上限とする手数で終了することを示した。こゝでｎは文章の長さ、Nは異なるｄｅｒｉｖａｔｉｏｎの数、そしてc_i はｎに無関係な定数である。

結　　　論

第１編、第２編ともに形式言語の範囲での議論であるが、自然言語のきわめて簡単な一つのモデルの議論とも考えられる。

各編の新しい諸結果はそれぞれの結論ですでにまとめられているからこゝではくり返すことをやめ、今後に残された問題について若干のべる。

有限状態言語間の翻訳を写像問題と考えることは、言語学的な観点に立てば構文論的問題の取扱いである。したがつて当然意味論的要素をも含めた取扱いが望まれる。その一つの方法として意味論の構文論的取扱いがあるが、今後の研究にまつべきものが多いようだ。有限状態言語より広いクラスの言語への拡張[46]~[48]（たゞし、unsolvable になれば意味がない）、写像によつて保存されるべき性質をより具体的に指定したときの問題などがある。

第２編の構文解析においても、明らかに意味論は入りこんでいないが、うまく情報を取入れることにより更にアルゴリズムの能率が上げられよう。また、それは自然言語の問題解決への糸口になるかも知れぬ。この種の問題について、プログラミング言語に潜在的に存在すると考えられるルールの適応に関する条件など付加した場合などについてさらに研究を続ける予定であり、一部の結果を得ている。[50]

ＣＦＬが n^3－recognizable でありあるＣＦＬの部分族が n^2－recognizable であることがわかつたが、$\overline{\lim}\, T(n)/(n/\log n)^2=0$ ならば、いかなるオンラインのチューリング機械でも $T(n)$－recognizable でないリニア言語が存在することが示されている。[37] したがつて $(n/\log n)^2$ と n^3 との間隔について、n^2 のみならずその間隔を狭め得る言語を求めることや、一般のＣＳＬの認識問題については今後に残された問題である。

謝　　　　辞

本研究の全過程を通じて、直接理解ある御指導を賜わり、つねに励ましていただいた尾崎弘教授、ならびに基礎工学部嵩忠雄教授に心から深謝する。また樹下行三講師には適切な御指導、御討論をいただいた。心から感謝の意を表する。

大学院修士、博士両過程において、御指導御教示を賜わつた電子工学教室の菅田栄治教授、喜田村善一教授、宮脇一男教授、中井順吉教授、寺田正純教授、裏克己教授、松尾幸人教授、中村勝吾教授、通信工学教室の熊谷三郎教授、青柳健次教授、笠原芳郎教授、板倉清保教授、産業科学研究所の加藤金正教授、基礎工学部の牧本利夫教授、藤沢和男教授、藤沢俊男教授に対して厚く御礼申し上げる。

本研究に関し、基礎工学部保田豊講師、電々公社電気通信研究所橋本昭洋氏には本大学在学中に有益な御助言をいただいた。心から感謝する。

筆者の所属している尾崎研究室の谷口慶治助手、大学院学生、白川功氏、都倉信樹氏、岡本光曜氏、谷口健一氏、村上伸一氏、高橋浩光氏、大川正尋氏、河田亨氏、また同研究室の大谷慧江、二宮和子、東尾洋子諸氏には種々の面で御協力いただいた。こゝに記して感謝する次第である。

以　上

文　　　　献

(1) N.Chomsky,Formal Properties of Grammars,in D.Luce, R.Bush,and E.Galanter (eds.),"Handbook of Mathematical Psychology",John Wiley & Sons,Inc.,New York, 1963

(2) N.Chomsky and M.P.Schutzenberger,The Algebraic Theory of Context-free Languages,In P.Braffort and D.Hirschberg (eds.),"Computer Programming and Formal Systems,"pp. 118-161,North Holland Publishing Company,Amsterdam, 1963

(3) Y.Bar-Hillel,M.Perles,and E.Shawir,"On formal properties of simple phrase structure grammars," In "Language and Information" by Y.Bar-Hillel,Addison-Wesley Publishing Co.Inc.,Reading,Mass., 1964

(4) 池野、"有限オートマトンの理論"、電気通信学会雑誌第４６巻、昭和３８年１１月、PP．1495－1502

(5) M.Davis,"Computability and Unsolvability," McGraw-Hill Book Co.,New York, 1958

(6) 赤，"ComputabilityとUnsolvability," 電気通信学会雑誌第４６巻，昭和３８年１１月，PP．1518－1524

(7) S.Ginsburg and G.F.Rose."Some recursively unsolvable problems in ALGOL-like languages,J.ACM,vol.10, 1963 PP．29－47

(8) S.Ginsburg and G.F.Rose,"Operations which preserve definability in languages," J.ACM,vol.10.1963, PP.175-

195

(9) S.Ginsburg and T.N.Hibbard,"Solvability of machine mappings of regular sets to regular sets," J.ACM,vol, 11, 1964,PP.302-312

(10) S.Ginsburg,"The Mathewatical Theory of Context-free Languages," McGraw-Hill Book Co., New York, 1966

(11) V.H.Yngve,"The depth hypothesis," In "Structure of Language and Its Mathematical Aspects," Am.Math.Soc., Rhode Island, 1961, PP.130-138

(12) M.O.Rabin and D. Scott,"Finite automata and their decision problems," IBMJ.of Research and Development, vol.3, 1959, PP.114-125

(13) O,Ore,"Theory of Graphs," American mathematical Society, 1962

(14) M.P.Schutzenberger,"A remark on finite transducers," Inf.and Contr.vol.4. 1961, PP.185-196

(15) 鳥居、嵩、尾崎、"有限状態言語間の翻訳についての一考察"、電気通信学会オートマトン研究会資料　昭和39年1月

(16) 鳥居、嵩、尾崎、"形式言語についての一考察"、電気四学会連合大会、昭和39年4月

(17) T.Kasami,K.Torii,and H.Ozaki,"Translation of finite, state languages by a sequential machine," In " Summaries of Papers , Part 3 Information Theory," ICMCI,1964

PP.153−154, The Inst. of Elect. Comm. Engrs. of Japan.

(18) 嵩、鳥居、尾崎、"一般化された順序回路による正規集合間の写像について"、電気通信学会雑誌 第48号、昭和40年7月

(19) 樹下、鳥居、尾崎、"一般化された順序回路による linear 言語間の写像についての一考察"、電気通信学会全国大会、昭和40年11月

(20) T.Kasami, K.Torii and H.Ozaki, "Generalized sequential machine mappings of finite state languages to finite state languages," Tech.Rep.of the Osaka Univ.vol.16, No. 688 (1966) PP.71−88

(21) T.Kasami, K.Torii and H.Ozaki, "A supplement to a Generalized Sequential Machine Mappings of Finite State Languages to Finite State Languages," To appear in Tech. Rep.of the Osaka Univ.

(22) R.D.Elbourn and W.H.Ware, "The evolutiou of concepts and languages of computing," Proceedings of the IRE, 1962, PP.1059−1066

(23) J.Hartmanis and R.E.Stearns, "On the computational complexity of algorithms," Transactions of the Am. Math. Soc. vol.117, 1965, PP.285−306

(24) J.Hartmanis and R.E.Stearns, "Computational Complexity of Recursive Sequences," Proc.Fifth Annual Sympos. on Switching Theory and Logical Design, Princeton, N.J. 2964, PP.82−90

(25) D.M.Lewis Ⅱ,R.E.Stearns,and J.Hartmanis,"Memory bounds for recognition of context-free and context-sensitive languages,"IEEE Conf.Record on Switching Circuit Theory and Logical Design,Sixth Annual Symp. 1965, PP. 191-202

(26) J.Hartmanis,P.M.Lewis Ⅱ,and R.E.Stearns," Classifications of computations by time and memory requirements," In Information Processing, 1965, PP. 31-35. Macmillan and Co.,Ltd,London, 1965

(27) R.W.Floyd,"Syntactic analysis and operator precedence," J.ACM vol.10,1963,PP.316-333.

(28) F.C.Hennie,"On-line turing machine computations," IEEE Trans.Electronic Computers EC-15,PP.35-44

(29) F.C.Hennie,"One-tape,off-line Turing machine computations," Inf. and Contr. vol.8, 1965,PP,553-578

(30) H.Yamada,"Real-time computation and recursive functions not real time computable," IRE Trans.Electronic Computers EC-11,1962,PP.753-760

(31) P.Gilbert,"On the syntax of algorithmic langnages," J.ACM,vol.13.No.1, 1966,PP.90-107

(32) S.Kuno,"The predictive analyzer and a path elimination technique," Comm.ACM.vol.8,No.7, 1965,PP.453-462

(33) S.Kuno and A.G.Oettinge,"Multiple-path syntactic analyzer," In Inform.Processing 1962,PP.306-312,

North Holland, Amsterdam, 1963

(34) T.V.Griffiths and S.R.Petriek, "On the relative efficiencies of context-free grammar recognizers," Comm.of ACM, vol.8, No.5, 1965, PP. 289-300

(35) A.G.Oettinger, "Automatic syntactic analysis and the pushdown store," In "Structue of Language and Its Mathematical Aspects," Am.Math.Soc., RHode Island, 1961, PP. 104-129

(36) T. Kasami, "An efficient recognition and syntax-analysis algorithm for context-free languages," Rep.of Coordinated Science Lab., Univ.of Ill., (1966).

(37) T. Kasami : "A note on conputing time for recognition of languages generated by linear grammars," Rep.of Coordinated Science Lab.Univ.of Illinois, 1966. To appere in Inf. and Contr.

(38) 鳥居、嵩、尾崎、" Context-free 言語の Recognition と Parsing の一方法" 電気通信学会オートマトン研究会資料、昭和４１年６月

(39) 鳥居、嵩、尾崎、" Context-free 言語の認識に関する一考察" 電気通信学会雑誌発表予定

(40) 鳥居、尾崎、" Context-sensitive 言語の認識に関する一考察" 電気通信学会全国大会、昭和４１年１１月

(41) D. H. Younger : "Recognition and parsing of context-free languages in time n^3," to appear.

(42) S. A. Greibach, "A new-normal-form theorem for context-

free phrase structure grammars," J. ACM, vol.12. 1965, PP. 42-52

(43) M. Kay : "A Parsing procedure," In Inf.Process. 1962, C. M. Popplewell(Ed.), North Holland.

(44) 戸田、"形式言語、特に前後無関係型言語について" 東北大学電気通信研究所シンポジウム資料、1966年3月

(45) S. Y. Kuroda, "Classes of languages and linear-bounded automata," Inf.and Contr.vol.7, 1964, PP. 207-223

(46) J. Myhill, "Linear bounded automata," Wright Air Development Division. Tech. Note 60-165, 1960

(47) P. C. Fischer, "On computability by certain classes of restricted Turing machine," Proc.the Forth Annual Symposium of Switching Theory and Logical Design, 1963, PP. 23-32

(48) R. M. Graham, "Bounded context translation," Proc, of Spring Joint Comp.Conf. 1964, PP. 17-29

(49) A. G. Oettinger, "Automatic processing of natural and formal languages," In Information processing, 1965, PP. 9-16, North Holland, 1966

(50) 鳥居、樹下、尾崎、"形式言語間の翻訳に関する一考察" 電気通信学会全国大会、昭和40年11月