



Title	形式言語に関する研究
Author(s)	鳥居, 宏次
Citation	大阪大学, 1967, 博士論文
Version Type	VoR
URL	https://hdl.handle.net/11094/2400
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

形式言語に関する研究

1967年1月

鳥居宏次

形式言語に関する研究

目 次

緒 論	1
第 1 編 有限状態言語間の翻訳	4
第 1 章 緒 論	4
第 2 章 諸 定 義	6
第 3 章 csm による写像	9
第 4 章 gsm による写像	26
第 5 章 第 4 章の補足	46
第 6 章 結 論	55
第 2 編 CFL の認識と構文解析	56
第 1 章 緒 論	56
第 2 章 諸 定 義	58
第 3 章 CFL の認識	62
第 4 章 CFL の構文解析	71
第 5 章 CFL のある部分族の認識	80
第 6 章 結 論	88
結 論	89
謝 辞	90

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しており、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。一般的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、 $M(L_A) \subseteq L_B$ 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づ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って、 $T(n)$ 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。ここで 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、一般的 CFL 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づき、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。ここで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 (gem 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 gem 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 gem 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 g_∞ 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編ではコンテキスト・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)にもとづいて述べられる。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、適っ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 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。この概念にもとづき、嵩は一般的のコンテキスト・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章ではコンテキスト・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・センシティブ (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 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。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 を略記する) 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 $M(L_A) \subseteq L_B$ 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)} 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

$$u = u_1 u_2 \dots \dots u_k \quad (u_i \leq A; 1 \leq i \leq k)$$

$$v = v_1 v_2 \dots \dots v_h \quad (v_i \leq A; 1 \leq i \leq h)$$

とするとき

$$uv = u_1 u_2 \dots \dots u_k v_1 v_2 \dots \dots v_h$$

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 , s_0 , 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) s_0 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 $u, v \leq X^*$ 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。

^{††} このとき uv は u と v の concatenation といわれる。

$$L = \{ u \mid \delta(s_0, u) \leq F \}$$

あるような有限オートマトン $A = (\Sigma, X, \delta, s_0, F)$ 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 sequential 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 ($\Sigma, X, A, \sigma, \delta, s_0$) 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) A 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 A^* 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、 δ 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 A をも含めて有限長の入力系列に対して、出力で空系列 A 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 F S L 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

F S L L_A, L_B に対応するオートマトンを

$$A_A = (\Sigma_A, X_A, \sigma_A, S_{A0}, F_A)$$

$$A_B = (\Sigma_B, X_B, \sigma_B, S_{B0}, F_B)$$

とする。これらに対応する状態図は簡約化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、

$$\sigma_A(S_{A1}, u_A) \in F_A$$

$$(\sigma_B(S_{B1}, u_B) \in F_B)$$

である $S_{A1} \in \Sigma_A$ ($S_{B1} \in \Sigma_B$) 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

$$\tilde{\Sigma}_A = \{S_{A0}, \dots, S_{A1}, \dots, S_{An}\}$$

$$\tilde{\Sigma}_B = \{S_{B0}, \dots, S_{B1}, \dots, S_{Bm}\}$$

とおく。

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

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

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

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

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。ここで、 M は $M_T \times M_A$ の等価な csm で
ることは容易にわかる。ただし、

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

ここで

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

他のすべてのとき

$$\delta_A(s_{A1}, x) = 0$$

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_{A1}, s_{B1} への入力系列の集合を

$$U(s_{A1}) = \{u / \sigma_A(u) = s_{A1}\}$$

$$V(s_{B1}) = \{u / \sigma_B(u) = s_{B1}\}$$

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 $0 \leq i \leq N$ に對して

$$U(s_i) \subseteq U(s_{Aj}) \quad (1)$$

をみたす $0 \leq j \leq n$ が存在する。

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

$$\delta_2(u_1v) = \delta_2(u_1)\delta_2(s_i, v)$$

$$\delta_2(u_2v) = \delta_2(u_2)\delta_2(s_i, v)$$

したがつて、 $\delta_2(u_1v)$ と $\delta_2(u_2v)$ の最終状態は等しい。すなわち

$$u_1v \in L_A \iff u_2v \in L_A$$

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、 $u_1, u_2 \in U(s_{Aj})$ 。

$U(s_{A0}), U(s_{A1}), \dots$ は互に共通な系列を含まないから、補題 3.1 より $i (0 \leq i \leq N)$ 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。

$$U(s_i) \subseteq U(s_{AD})$$

をみたすような s_i を Σ から除いた残りを $\tilde{\Sigma}$ とおき、適当に番号をつかえて

$$\tilde{\Sigma} = \{s_0, s_1, \dots, s_i, \dots, s_N\}$$

とおく。 $\Sigma - \tilde{\Sigma}$ 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 $\tilde{\Sigma}$ に含まれる状態である。

補題 3.2 : 任意の $i (0 \leq i \leq N)$, $x_1, x_2 \in X_i^{(A)}$ $x_1 \neq x_2$ 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、 $ux_1v_1 \in L_A, ux_2v_2 \in L_A$ 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 に對して、 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 : $\sigma(u_1), \sigma(u_2) \in \tilde{\Sigma}, \sigma_1(u_1) = \sigma_1(u_2)$

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 $i (0 \leq i \leq n)$ に對して

$$V(s_i) \cong V(s_{Bj}) \quad (2)$$

をみたす $0 \leq j \leq m$ が存在する。

(証明) $u \in L_A \iff \delta_1(u) \in L_B$ 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 に對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、 $\delta_1(u_2)v = L_B$ 。 L_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 $i (0 \leq i \leq n)$ のとき、(2)をみたす $0 \leq j \leq m$ 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 : $w_1 w_2 w_3 \in L_B$, $w_2 \neq A$, $\sigma_B(w_1) = s_{Bj}$ 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、 $\delta_1(u_1)w_2w_3 \in L_B$ 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 ($0 \leq i \leq N$) に対して、

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

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

(証明) 任意の $y \in X_{\beta(i)}^{(B)}$ 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、
 $y = \delta_1(s_i, x)$, $x \in X_{\alpha(i)}^{(A)}$ 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 $X_{\beta(i)}^{(B)}$ 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。

$$M' = (\Sigma - \{s_{i_2}\}, X_A, X_B \times \{0, 1\}, \sigma', \delta', s_0)$$

$$\text{ただし, } \delta' = (\delta'_1, \delta'_2)$$

$$\text{また, } \sigma(s_i, x) = s_j, j \neq i_2 \text{ のとき } \sigma'(s_i, x) = s_j$$

$$\sigma(s_i, x) = s_{i_2} \text{ のとき, } \sigma'(s_i, x) = s_{i_1}$$

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 s_{i_2} が含まれなければ、 $\delta'_1(u_0) = w_0$ 含まれるとき、 u_0 の最初の部分系列 u_1 のうち

$$\sigma(u_1) = s_{i_2}$$

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、

$$\delta_1(u_1) = \delta'_1(u_1) \text{ これを } w_1 \text{ とおく。}$$

$$w = \delta_1(u_1) \delta_1(s_{i_2}, v_1) \quad (3)$$

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

$$w'_1 = \delta_1(s_{i_2}, v_1) = \delta_1(s_{i_1}, v'_1), v'_1 \in X_A^*$$

* 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

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

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、
 $u_1 v'_1 \in L_A$ 。

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、 $\delta'_1(u_1 v'_1) = w$ 。もし含まれると、 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。 $u_0 = u_1 u_2 v_2$ とおくと、

$$\delta'_1(u_1 u_2) = w_1 w_2$$

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、

$$\delta'_1(u_1 u_2 \dots) = w_1 w_2 \dots = w$$

となることがわかる。

ついで、1対1対応をなすことを証明する。

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

また、 $u_1 = u_0 x_1 u_3$, $u_2 = u_0 x_2 u_4$, $x_1 \neq x_2$, $x_1, x_2 \in X_A$ 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、 δ'_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

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 $\alpha(i_1), \beta(i_1)$ を定義すれば、任意の $i_1 \neq i_2$ 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 $i_1 \neq i_2$
($0 \leq i_1, i_2 \leq N$)について

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

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 $(j, k) \in \widetilde{\Sigma}$ について、

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

ならば

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

$$\text{ただし } s_{Aj'} = \sigma_A(s_{Aj}, x)$$

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

(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 $(j, k) \in \widetilde{\Sigma}, 0 \leq j' \leq n, 0 \leq k' \leq m$ について

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

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

§ 3.4 翻訳可能な条件

以上よりつきの定理が得られる。

定理 3.1 : つきの条件をみたす $(n+1)(m+1)$ 個の組 (j, k) ($0 \leq j \leq n, 0 \leq k \leq m$) の部分集合 $\tilde{\Sigma}$

(i) $(0, 0) \in \tilde{\Sigma}$

(ii) $(j, k) \in \tilde{\Sigma}$ ならば

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

(iii) 任意の $0 \leq j \leq n$ について、 $(j, k) \in \tilde{\Sigma}$ である k が少くとも 1 つ存在する。

(iv) 任意の $0 \leq k \leq m$ について、 $(j, k) \in \tilde{\Sigma}$ 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 $P_{jk, j'k'} ((j, k), (j', k') \in \tilde{\Sigma})$ が存在する。

が存在することが、条件(i), (ii)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) $\Sigma \sim \tilde{\Sigma} U S_D, S_0 = (0, 0)$ とおく。

2) 各 $(j, k) \in \tilde{\Sigma}$ 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、 $P_{jk}, \mathcal{J}\ell$ ($P_{jk}, \mathcal{J}\ell$) 個のシンボルよりなる部分集合 $X_{jk, \mathcal{J}\ell}^{(A)}$
 $(\ell | (j', \ell) \in \tilde{\Sigma}) (X_{jk, \ell k'}^{(B)} (\ell | (\ell, k') \in \tilde{\Sigma}))$ 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 $X_{jk, \mathcal{J}k'}^{(A)} (X_{jk, \mathcal{J}k'}^{(B)}) ((j, k), (j', k') \in \tilde{\Sigma})$ は互いに共

通な要素をもたず、かつ

$$X_j^{(A)} = \bigcup_{(j', k') \in \tilde{\Sigma}} X_{jk, \mathcal{J}k'}^{(A)}$$

$$(X_k^{(B)} = \bigcup_{(j, k') \in \tilde{\Sigma}} X_{jk, \mathcal{J}k'}^{(B)})$$

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

$$\sigma((j, k), x) = (j', k'), x \in X_{jk, \mathcal{J}k'}^{(A)}$$

$$\sigma((j, k), x) = s_D, x \in X_j^{(A)}$$

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

ここで、 $(0, 0)$ から到着できない状態は除くものとする。

4) 各 $(j, k) \in \tilde{\Sigma}$ につき $X_{jk, \mathcal{J}k'}^{(A)}$ と $X_{jk, \mathcal{J}k'}^{(B)}$ ($(j', k') \in \tilde{\Sigma}$)

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 $X_j^{(A)}$ と $X_k^{(B)}$ 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 $x \in X_j^{(A)}$ 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 $(j, k) \in \tilde{\Sigma}$ について

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

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

とおく。 Γ についての帰納法により容易に

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

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

§ 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“ \approx ”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

$$J_0^{(0)} = \{ j \mid s_{Aj} \in F_A \}$$

$$J_1^{(0)} = \{ j \mid s_{Aj} \notin F_A \}$$

ここで F_A は最終状態の集合を表わす。このとき

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

とする。

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

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

とする。

(iii) 各 $J_{i\ell}$ ($0 \leq i \leq q_r$) 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

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

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

$$J_{i\ell}^{(r)} \quad 0 \leq \ell \leq r_i$$

とおき

$$A_A^{(r+1)} = \{ J_{00}^{(r)}, J_{01}^{(r)}, \dots, J_{10}^{(r)}, J_{11}^{(r)}, \dots, J_{q_r 0}^{(r)}, \dots, J_{q_r r}^{(r)} \}$$

とおく。もし $A_A^{(r+1)} \neq A_A^{(r)}$ ならば、 $r+1 \rightarrow r$ として、(iii) をくり返す。 $A_A^{(r+1)} = A_A^{(r)}$ ならば、 $A_A^{(r)}$ を書き直して

$$A_A^{(r)} = \{ J_0^{(r)}, J_1^{(r)}, \dots, J_{q(A)}^{(r)} \}$$

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 $A_s = (\Sigma_s, X_s, \sigma_s, S_t, F_s)$ はつきのように書ける。

$$\Sigma_s = \{ s_0, s_1, \dots, s_{q_r}, s_D \}$$

$$F_s = \{ s_i \mid J_i^{(r)} \subseteq J_1^{(0)} \}$$

かつ、 t は $0 \leq J_t^{(r)}$ 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。

$$\delta_s(s_i, x) = s_h \quad , \quad x \in X_{ih}$$

$$\delta_s(s_i, x) = s_D \quad , \quad x \notin X_{ih}$$

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。 $(a_1, a_2, \dots, a_i, \dots) > (b_1, b_2, \dots, b_i, \dots)$

$(b_1, b_2, \dots, b_i, \dots)$ は $a_j = b_j$ ($1 \leq j < i$) 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。

$\sum_{h=0}^{q_f} \mu_{ih}^{(A)}$ 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 3.1.2 は明らかである。

補題 3.1.2：任意の $\theta(L_A)$ に對して、その標準形 $S(\theta_A)$ は、一意的に定められる。

補題 3.1.3： J_0, J_1, \dots, J_q : K_0, K_1, \dots, K_q をそれぞれ整数の組 $\{j / 0 \leq j \leq n\}$, $\{k / 0 \leq k \leq m\}$ の分割とする。 (すべての J_i, K_i が空でないとする)

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

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

とおく。もし M が存在して

$$\tilde{\Sigma} = \bigcup_{i=0}^q J_i \times K_i$$

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

* $J = \bigcup_i J_i$, $J_i \cap J_j = \emptyset$ ($i \neq j$)、かつ全ての $J_i \neq \emptyset$ 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) に對して、 $j \in J_i$, $k \in K_0$ ならば $i = \emptyset$ というこ

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

また、任意の $j \in J_j$ に対して、

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

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、 $0 \leq j \leq n, 0 \leq k \leq m$ とすると、(6)、(7)をみたすことから

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

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

が作れる。ここで J_0, J_1, \dots, J_{p_1} は整数の組 $\{j / 0 \leq j \leq n\}$ の分割であり、 K_0, K_1, \dots, K_{p_2} は整数の組 $\{k / 0 \leq k \leq m\}$ 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、 $\rho_1^{(A)}$ の等しい J_i の要素をまとめて

$$J_{j_0}, J_{j_1}, \dots, J_{j_r}$$

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

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

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、

$$\begin{aligned} A_A^{(r)} &= A_A^{(r+1)} = \{ J_{00}^{(r)}, J_{01}^{(r)}, \dots, \dots, J_{10}^{(r)}, J_{11}^{(r)}, \dots \\ &\quad J_{q_r 0}^{(r)} \dots \dots J_{q_r r_{q_r}}^{(r)} \} \\ A_B^{(r)} &= A_B^{(r+1)} = \{ K_{00}, K_{01}, \dots, \dots, K_{10}, K_{11}, \dots \\ &\quad K_{q_r 0}^{(r)} \dots \dots K_{q_r r_{q_r}}^{(r)} \} \end{aligned}$$

であるとして、翻訳機はつきのように求めることができる。

$\Sigma = \bigcup_{i=0}^{q_r} J_i^{(r)} \times K_i^{(r)}$ 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

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

$$\sum_{\ell \in J_i^{(r)}} P_{j k, \ell k'} = P_{k k'}^{(B)} \quad , \quad k' \in K_i^{(r)}$$

を別々にとけばよい。結局つきの形の方程式をとくことになる。

$$\sum_{j=1}^g Z_{i j} = C_i \quad (1 \leq i \leq f)$$

$$\sum_{i=1}^f Z_{i j} = D_j \quad (1 \leq j \leq g)$$

ただし、 Z_{ij} , C_i , D_j は非負整数でかつ

$$\sum C_i = \sum D_j$$

$Z_{ij} \geq 0$ 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、 $f \leq g$ とする。

$$Z_{ij}^{(0)} = \min (C_i, D_i) \quad 1 \leq i \leq f$$

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

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 Σ 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 " \approx " により F S L 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 より更により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(8)にもとづ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、 $M(L_A) \subseteq L_B$ 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。

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（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、考へ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へる。⁽¹¹⁾ こゝでは文章ごとの対応について考へ、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へればよいから、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取り除いたものを考へる。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 $g : M$ 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）に到達するようになる。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）から出る枝を取り除くことにする。 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つたものである。

状態 s_i に関する "prefix set" P をつき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 $p_1, p_2 \in P$ に對して、 p_1, p_2 は一方が他方の接頭語 (prefix) になることはない。 (2) $\sigma(s_i, u) \in F$ であるような任意の u に對して、 $u = p v$, $p \in P$, $v \in X^*$ 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 $\sigma(s_i, p v) \in F$ 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 による変換をつきのように定義する : 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回くり返してグラフ 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き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) $\delta(b_{i_2}) = \delta(b_{i_1})v_2, \delta(b_{i_3}) = \delta(b_{i_2})v_3, \dots, \delta(b_{i_h}) = \delta(b_{i_{h-1}})v_h$, ($v_2, \dots, v_h \neq A$) 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

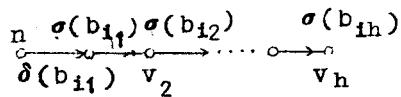


図 4.1 枝の分割

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 g_{SM} 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 Σ による変換によつて得られる。

定義から L の系列全体と、 L 上の初期状態から最終状態への（有限長の）道全体 ($P(L)$ 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 g_{SM} 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“ \sim ”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“ \sim ”を保存する諸変形

この節では“ \sim ”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, \dots, 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、 $n'n$ 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 $\overrightarrow{n_i n'}$ 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 $n_i n_{i1}, n_i n_{i2}, \dots$ ($1 \leq i_1, \dots, \leq k$) (n_1, \dots は重複してよい) をつける。(図 4.2)

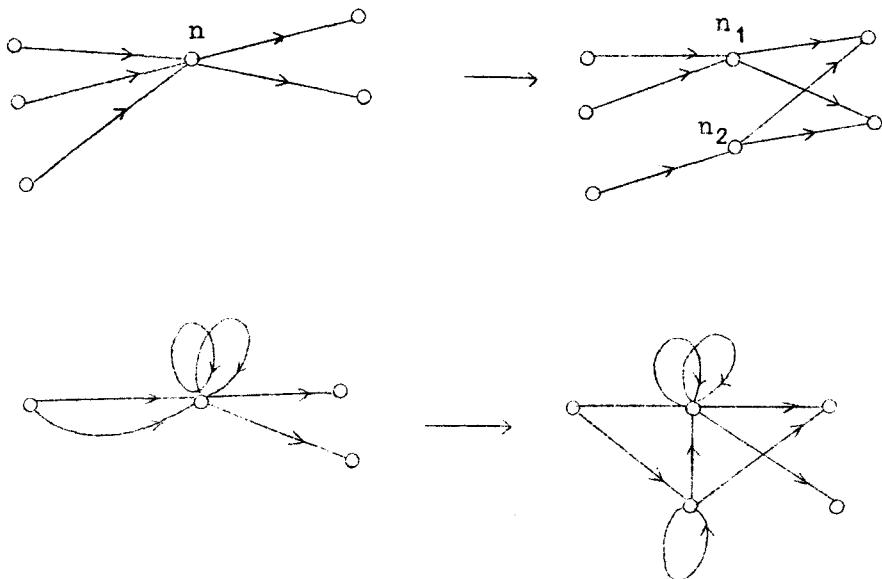


図 4.2 節点の分割

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）

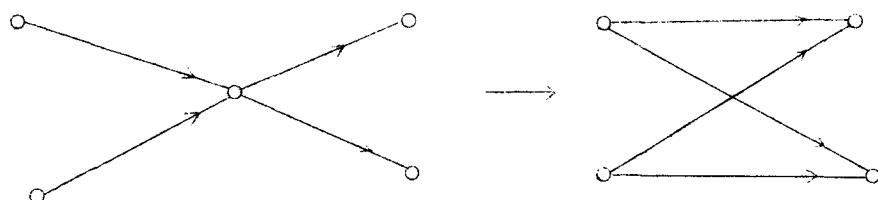


図 4.3 節点の消去

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 $\overrightarrow{n_1n}, \overrightarrow{n_2n}, \dots$ 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。 $\vec{n}_1 \bar{n}, \vec{n}_2 \bar{n}, \dots$ を取り除き、かわりに枝 $\vec{n}_1 \bar{\bar{n}}, \vec{n}_2 \bar{\bar{n}}, \dots$ を取り除き、かわりに枝 $\vec{n}_1 \bar{n}, \vec{n}_2 \bar{n}, \dots$ 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)

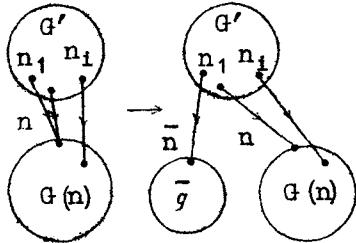


図 4.4 部分グラフの複製

部分グラフの併合： G の始点を含まない 1-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} の部分を取り除き、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、 $G \sim G'$
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 がつき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 $P_1 (\bar{P}_1)$ 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 $g \sim \bar{g}$ ($n_0; n_a, \dots, n_b$) 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）、 $G \sim \bar{G}$ 。

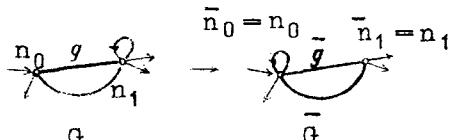


図 4.5 部分グラフの変換

（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 $g \sim \bar{g}$ ($n_0; n_0, n_1$)

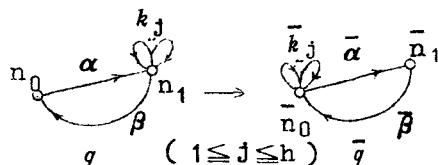


図 4.6 補題 4.4 のグラフ g と \bar{g}

補題 4.5*： g を節点 $n; n_0, \dots, n_b$ と n_0, n におけるそれぞれ $\rho_0 (\geq 0)$ ， $\rho (> 0)$ 本の self-loop， n_0 から n への $\nu_0 (\geq 0)$ 本の枝、

* 証明は 5 章参照。

n_0, n から n_i ($1 \leq i \leq b$) へのそれぞれ μ_i (≥ 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 “ \sim ” をつけて示すことになると

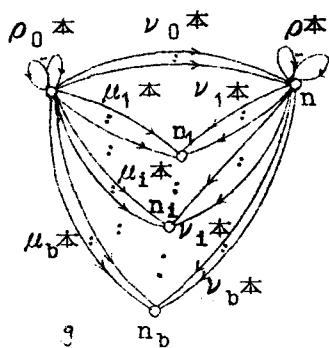


図 4.7 補題 4.5 のグラフ

(a) $\rho, \bar{\rho} \geq 2$ ならば、 $g \stackrel{s}{\sim} \bar{g} (n_0; n_0, n_1, \dots, n_b)$

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

$$\nu_1 / \nu_0 = \dots = \nu_b / \nu_0$$

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

(ただし、 $n_0 (\bar{n}_0)$ からでる枝がないときは、すなわち $\rho_0 = \nu_0 = 0$ ($\bar{\rho}_0 = \bar{\nu}_0 = 0$) 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 graph⁸ 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、つき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 $n^{(0)}, n^{(1)}, \dots$ 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 $l^{(1)}$ 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 $\{k_1\}$ とする。 $\{k_1\}, \beta$ 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 になるようとする。ただし n_1, n' には n と同数の self-loop をつける。節点 $n_0, n_1, \text{ 枝 } \alpha, \beta$ と n_1 の self-loop $\{k_1\}$ 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 $\{k_1\}$ 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 $\rho \geq 2$ 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、 $\rho \geq 3$ 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 $l^{(1)}$ 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 $n^{(1)}$ 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 $l^{(1)}$ について上述の変形を行なうとき、変形後 $G(n^{(1)})$ は互いに 1-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、 $n^{(1)} (i \neq 0)$ 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)

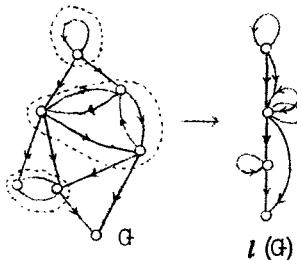


図 4.8 $l(G)$ の説明

(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 $I(G)$ とおくと、上述より、 $G \sim I(G)$ 。つぎに $I(G)$ 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： 棱 e の両端点を結ぶ長さ 2 以上の道がある（ない）とき e 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 $\mu_1, \bar{\mu}_1$ 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）

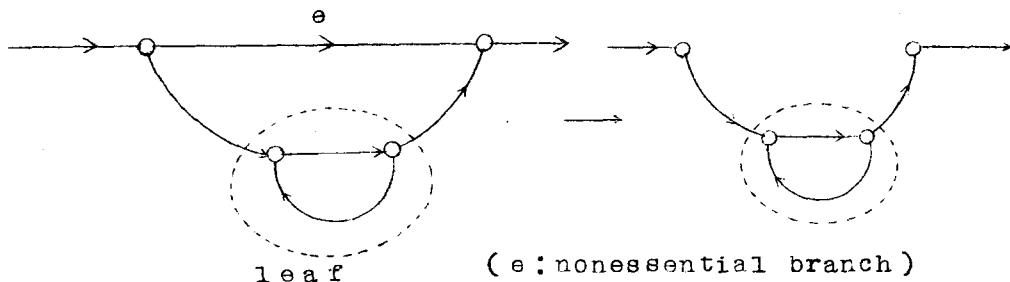


図 4.9 基本的でない枝の除去

(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, \dots, n_h , n から n_1 への

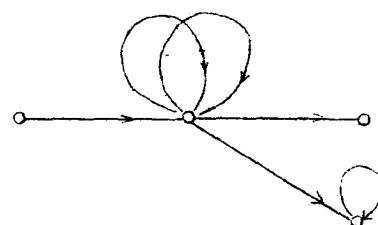
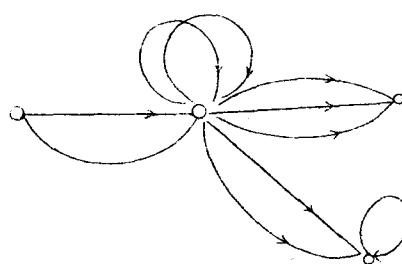


図 4.10 枝数の減少 (a)

枝数を ν_i かつ $\nu_1, \nu_2, \dots, \nu_h$ の最大公約数を ν とする。 n から n_i へ
の枝数を ν_i/ν 本に減らす。(図 4.11)

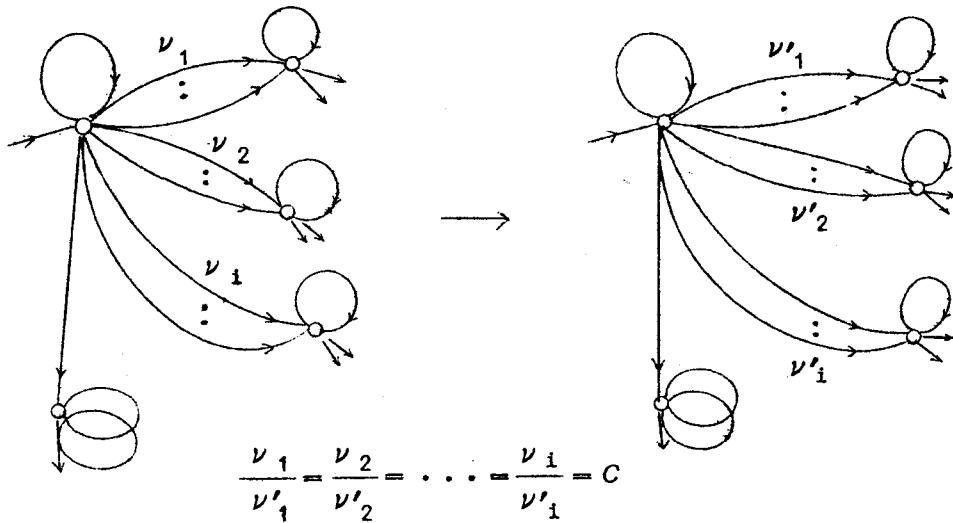


図 4.11 枝 数 の 減 少(b)

(iia) は補題 4.3 と補題 4.5 の(a)で $\nu_0, \nu_i (>0)$ 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
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取り除くと、 $G(n_1)$ 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 $\rho_0, \nu_0 (\neq 0)$ 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

$I(G)$ 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。

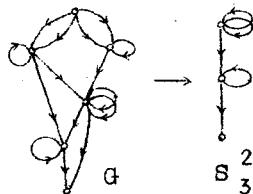


図 4.1.2 標 準 化 の 説 明

(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、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。可能な限り (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 ν 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 ν 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 ($0 \leq j \leq \nu$) 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。

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

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 とよぶことになると、

* 1- 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きのような 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がついているとする。
 - 2b) 一方のレベル 1 の D_j の各節点について、他方のレベル 1 の D_j の節点のなかにそれからレベル $i - 1$ 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。^{*} アルファベット 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 $I(G)$ 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 4.1.3 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

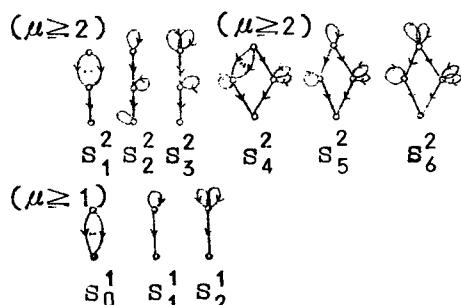


図 4.1.3 $N_L = 1$ と $N_L = 2$ の標準形

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 $I(G)$ 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 $g \otimes m$ による写像の存在条件

$L_A \rightarrow L_B$ 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 $G \rightarrow G'$ 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、 $S(G) \rightarrow S_2^1$ 、 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。併合条件 1、2 より $S_2^1 \sim \bar{S}$ 、また $\bar{S} \rightarrow S(G')$ 。したがつて、 $G \sim S(G)$ 、 $G' \sim S(G')$ より、 $G \rightarrow G'$ 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 4.1.4 に示した形以外にはない。 D_1 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 4.1.4 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 から直接つきのようにならなければならない。

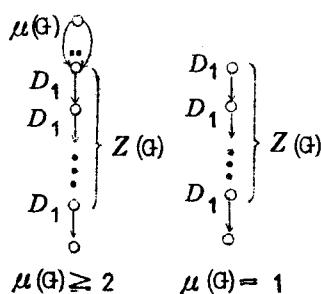


図 4.1.4 D_1 のみからなる標準形

* 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。 $\mu(G)$: $Z(G)$ 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 $L_A \rightarrow L_B$ 。

(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、 $Z(G_A) = Z(G_B), \mu(G_A) \geq \mu(G_B)$

(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 g_{em} の状態図を G_M 、その入力グラフ、出力グラフを G_I, G_O 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_O 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_O の任意の道 P_O (対応する G_M の出力系列を α) に對して、 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 $\{P_{I_1}\}$ を対応させる。これを充分大きくとることにより、 P_O が閉路 \bar{C} を ν 回まわつた部分を含むとき、どの P_{I_1} も少なくとも 1 つの閉路を含むようにできる。また G_O のつくり方から P_O 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_1} 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_O がかりに 2 個以上の閉路をもつ leaf を含むとする。 P_O 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

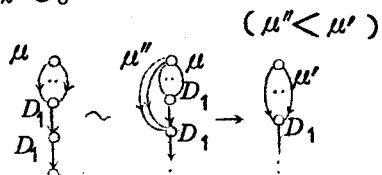
* 上への写像ゆえ空でない。

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、 $Z(G_A) \geq Z(G_B)$ が示される。また、 $Z(G_A) = Z(G_B)$ のとき $\mu(G_A) \geq \mu(G_B)$ も容易に示される。

逆に $S(G_B)$ が D_2 の点を含まず、 $Z(G_A) = Z(G_B)$, $\mu(G_A) \geq \mu(G_B)$ ならばあきらかに $G_A \rightarrow G_B$ である。つぎに $Z(G_A) > Z(G_B)$ とする。

$\mu(G_A) \geq \mu(G_B)$ 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。



基本的で一番上の
ない枝を loop を除く
加える

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 に對して、その写像像も無限の文章を含む。うな L_A から L_B の中への gsmC 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 gsmC 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 に對し、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 $\rho(G)$ (ρ は系列の前後を逆転する操作)とすると、 $\rho(G)$ も1つのFSLの状態図である。

$S(\rho(G))$ 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 $\rho M_2 \rho M_1^{(1)}$ を考えればよい。この場合に前述の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、つきのような $m, r, H_1 (\bar{H}_1)$ が一意にきまる。^{*}

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

$$P_1(m; r; H_1, \dots, H_m) = (\alpha\beta)^r \alpha H_1 \beta \alpha H_2 \dots \beta \alpha H_m$$

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

$$m \geq 1, r \geq 0, H_1 \in K^* \quad (\bar{H}_1 \in \bar{K}^*), H_1 \neq A \quad (\bar{H}_1 \neq A).$$

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

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

$$P_0(m; r; H_1, \dots, H_m) = (\alpha\beta)^r \alpha H_1 \beta \alpha H_2 \dots \beta \alpha H_m \beta (\bar{P}_0$$

$$(m; r; \bar{H}_1, \dots, \bar{H}_m) = (\bar{\alpha}\bar{\beta})^r \bar{H}_1 \bar{\alpha} \bar{\beta} \bar{H}_2 \dots \bar{\alpha} \bar{\beta} \bar{H}_m)$$

$$m \geq 1, r \geq 0, H_1 \in K^* \quad (\bar{H}_1 \in \bar{K}^*), H_1 \neq A \quad (\bar{H}_1 \neq A)$$

ここで $P_d(0; r)$ と $\bar{P}_d(0; r)$ ($d=1, r \geq 0$; $d=0, r > 0$), P_d

$(m; r; H_1, \dots, H_m)$ と $\bar{P}_d(m; r; \bar{H}_1, \dots, \bar{H}_m)$ ($m \geq 1, r \geq 0$,

$H_1 = k_{i_1} k_{i_2} \dots k_{i_{q_1}}, \bar{H}_1 = \bar{k}_{i_1} \bar{k}_{i_2} \dots \bar{k}_{i_{q_1}}, H_1 \neq A$) を対応させる

と、1対1でもれなく対応している。 $(\alpha\beta)^r, \bar{H}_1$ 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。

* $(\alpha\beta)^r = \underbrace{\alpha\beta\alpha\beta\dots\dots\alpha\beta}_r$

§ 5.2 補題 4.5 の証明

(a) を四つの場合

(i) $\bar{\rho} \geq 3$ 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) $\rho = \tau + 2$, $\bar{\rho} = \tau + 3$ ($\tau \geq 0$) 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'_j , \bar{n}'_j を通らな

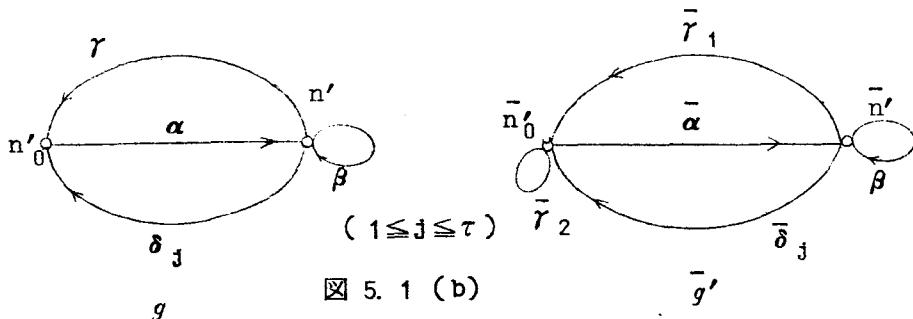


図 5.1 (b)

いでそれぞれ n'_0 , \bar{n}'_0 にかかる 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。

$$\alpha\beta^n\delta_j \leftrightarrow \bar{\alpha}\bar{\beta}^n\bar{\delta}_j \quad (n \geq 0); \alpha\tau \leftrightarrow \bar{\tau}_2; \alpha\beta\beta^n\tau \leftrightarrow \bar{\alpha}\bar{\beta}^n\bar{\tau}_1 \quad (n \geq 0)$$

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 $g' \xrightarrow{S} \bar{g}'$ ($n'_0 : n'_0$)。 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 “ \sim^S ” 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) $\rho = 2$, $\bar{\rho} = 3$, $\nu_1 = 1 + \tau$, $\bar{\nu}_1 = 2 + \tau$ ($\tau \geq 0$) のとき (図 5.2 (a))

図 5.2 (b) の g' の節点 n' において prefix set $\{\beta^2, \tau, \delta, \beta\tau, \beta\delta\}$

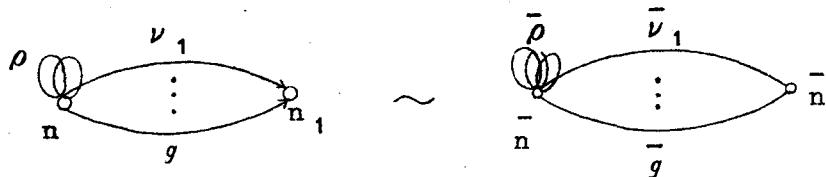


図 5.2 (a)

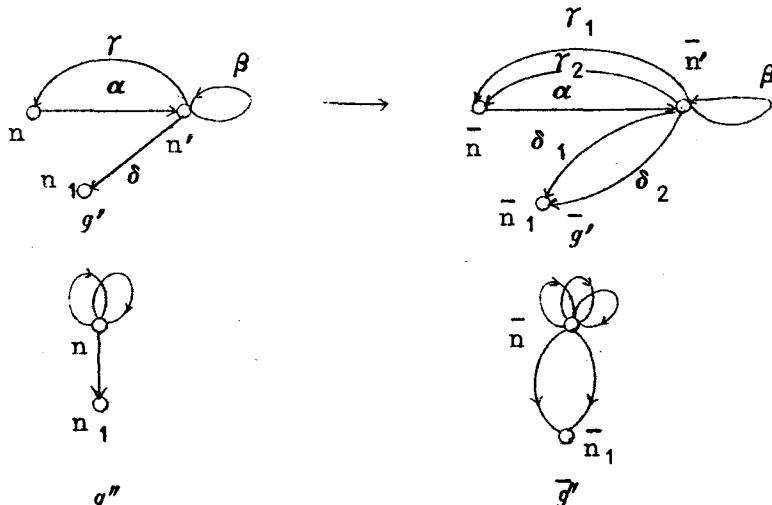


図 5.2 (b)

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、 $g'' \xrightarrow{S} \bar{g}''$ ($n : n, n_1$)。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 $\rho = \bar{\rho} = 1$

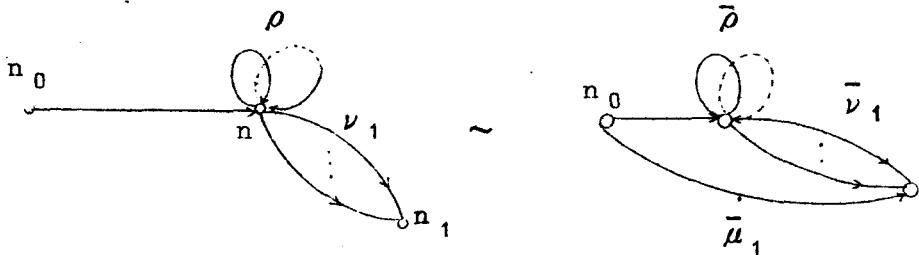


図 5.3 (a)

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よう。

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

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

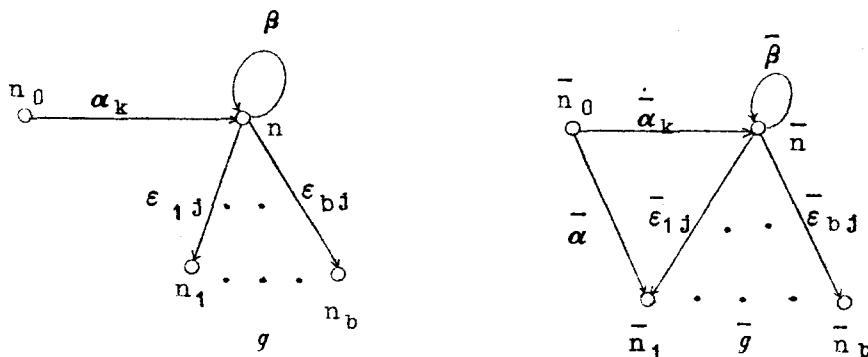


図 5.3 (b)

(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))

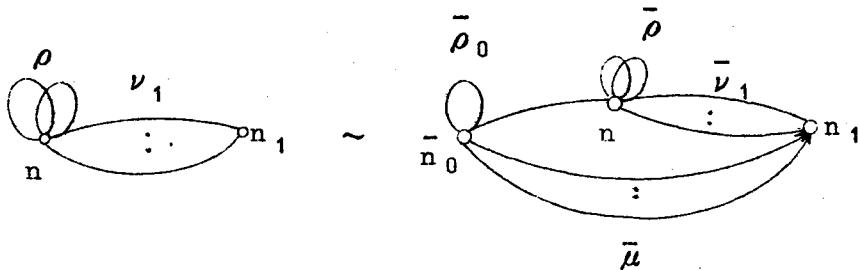


図 5.4 (a)

(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 “ \sim^s ” 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 $\nu_{\bar{\nu}_1} = \nu_{\bar{\nu}_2} = \dots = \nu_{\bar{\nu}_b} = c$ 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 $\nu_{11}, \nu_{12}, \dots, \nu_{1\nu_1}$, このとき、節点 n において prefix set $\{ \rho^c, \nu_{11}, \nu_{12}, \dots, \nu_{1\nu_1}, \rho\nu_{11}, \rho\nu_{12}, \dots, \rho\nu_{1\nu_1}, \rho^2\nu_{1\nu_1}, \dots, \rho^{c-1}\nu_{11}, \dots, \rho^{c-1}\nu_{1\nu_1}, \dots \}$ による変換を行えよ。さらに前述の (a) の証明 (iii) より $\mu_1 = \bar{\mu}_1$ とおけること、“ \sim^s ” 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} において枝数の等しいものは、かかない。

ラフ、出力グラフを G_I , G_0 とする。

補題 4.6 : G IC 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)

$$S(G_A) = S(G_I), S(G_B) = S(G_0) \quad (1)$$

こゝでまずつきの補題に注意する。

補題 4.7 : つきの性質をもつ 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 IC 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ぬ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、
 $N_L \leq Z$ 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 $\{a_1\}$ (self-loop をもたないと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 e_1, e_2, \dots, e_r とする。各 $G_A(e_i)$ 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 で述べた

* $\sigma(s_i, u_1) \in F, \sigma(s_i, u_2) \in F$

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 $G_{I0}^{(j)}$ とする。

ここで 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 $\alpha_\tau, \varepsilon_j$ 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 $\{b_k\}$ に對して、 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なる。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

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

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 に對応する G_0 の節点を $f(n)$ とかく。節点集合 $\{f(n) | n \in G_{\tau i}^{(j)}\}$ ($\{f(n_0^{(j)})\}$)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 $G_{\tau i 0}^{(j)}(G_{00})$ 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。

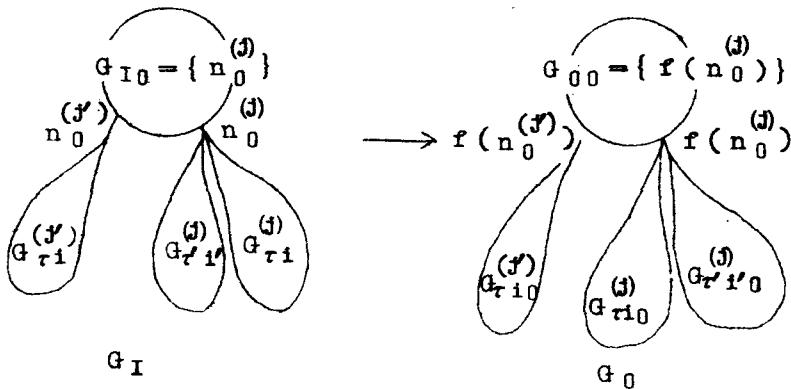


図 4.16 補題 4.8 の説明

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 $G_{A_i} \sim G_{\tau i_0}^{(j)}$ 。帰納法の仮定より

$$s(G_{A_i}) = s(G_{\tau i_0}^{(j)}) \quad (3)$$

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、 $1 \leq i \leq r$) 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 1-1 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、 $G_{A1}, G_{\tau 1}, G_{\tau i0}, G'_A$ 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

$csml$ 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、 $M(L_A) \subseteq L_B$ 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 C F L 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 C F L が n^3 、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 C F L について、第 5 章では (メタ-) リニア言語をその一部として含む C F L 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、 $n^3 - \text{recognizable}$ 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、つき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。ただし $\varphi \in V_N^* - \{\Lambda\}$, $\psi \in (V_N \cup V_T)^* - \{\Lambda\}$ 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。 $\varphi_1, \varphi_2 \in V^*$ に對して、
 $\varphi_1 = w_1 u w_2$, $\varphi_2 = w_1 v w_2$ 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。 $\varphi, \psi \in V^*$ に對し、 $\varphi = \psi$ または、 $\varphi = \varphi_0, \varphi_r = \psi$ で $\varphi_1 \xrightarrow{s_1+1} \varphi_{1+1}$

であるような系列 $\varphi_0, \varphi_1, \dots, \varphi_r$ が存在するとき

$$\varphi \xrightarrow{g} \psi \quad (\text{または } \varphi \xrightarrow{\text{def}} \psi)$$

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

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

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、 $w_1 X w_2 \rightarrow w_1 x w_2$ の形のみで、 $X \in V_N$, $w_1, w_2 \in V_N^*$, $x \in V^* - \{ A \}$ 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 $X \in V_N$, $x \in V^* - \{ A \}$ 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

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

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

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

$$S \rightarrow \psi \quad (\psi \in (V - \{ S \})^* - \{ A \})$$

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

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

の形のみの場合、あるいは、

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

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 φ , ψ 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、その読み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、 $n t$ -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 C F L 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 C F L 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、

$$X \rightarrow YZ$$

$$X \rightarrow a$$

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

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

$$N(i, k) = \{Y \mid Y \rightarrow a_1 a_{i+1} \dots a_k, Y \in V_N\}$$
$$(1 \leq i \leq k)$$

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、

$$Y \rightarrow a_1 a_{i+1} \dots a_k \quad (1 + 1 \leq k)$$

ならば、

$$X_1 \rightarrow a_i a_{i+1} \dots a_j$$

$$X_2 \rightarrow a_{j+1} \dots a_k$$

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

でかつ

$$Y \rightarrow X_1 X_2 \quad P$$

なる X_1, X_2 が存在する。したがつて、

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

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

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 $N(i, k')$ ($1 \leq i \leq k' < k$) と $N(k, k)$ とが必要であり、逆にそれらがわかれば各 j ($i \leq j \leq k$) について $N_j(k, k), N_j(k-1, k), \dots, N_j(1, k)$ が順次求められ、その結果 $N(k, k), N(k-1, k), \dots, N(1, k)$ 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 $N_j(k, k), N_j(k-1, k), \dots, N_j(1, k)$ 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 $N(i, k')$ および $i = k'$ のとき限り、 $a_{k'}$ をも貯える。

には j が $k-1, k-2, \dots, 1$ の夫々の場合が順次書き加えられていく。

たとえば区画 $T_1^{(2)}$ には $N_{k-1}(k-i+1, k), N_{k-2}(k-i+1, k), \dots, N_{k-i+1}(k-i+1, k)$ が順次書き加えられる。その結果

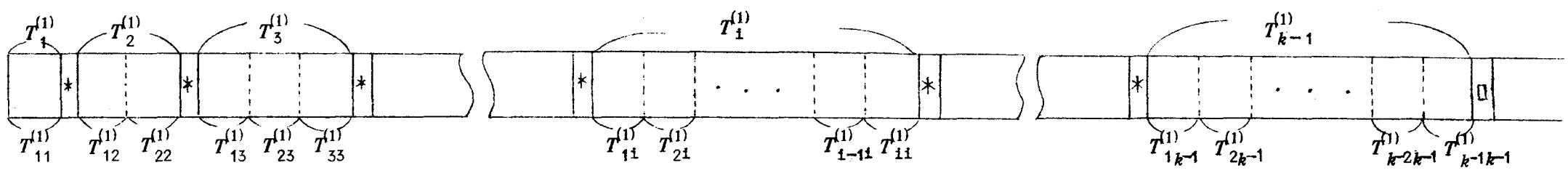
$$N(k-i+1, k) = \bigcup_{j=k-i+1}^{k-1} N_j(k-i+1, k)$$

が $T_1^{(2)}$ 上に得られる。この操作により得られた各 $T_1^{(2)}$ の内容を $T^{(1)}$ 上に移すことにより $T^{(1)}$ 上に望む結果 $N(k, k), N(k-1, k), \dots, N(1, k)$ 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ブ $T^{(1)}$ 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。ただし C_1 は k に無関係な定数である。この間のテーブ $T^{(2)}$ 上の各区画は $1 (< k)$ 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

ワーキングテーブ $T^{(1)}$ と $T^{(2)}$ をもち、それぞれは右半無限とする。

$T^{(1)}$ は $T_1^{(1)}, T_2^{(1)}, \dots, T_k^{(1)}, \dots$ 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 $T_{1k}^{(1)}, T_{2k}^{(1)}, \dots, T_{ik}^{(1)}, \dots, T_{kk}^{(1)}$ 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 nt. シンボルの集合 V_N のある部分集合、および特別の印口、△、○が同時に書き込めるものとする。 $T^{(2)}$ は $T_1^{(2)}, T_2^{(2)}, \dots, T_k^{(2)}, \dots$ 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 $T^{(1)}$

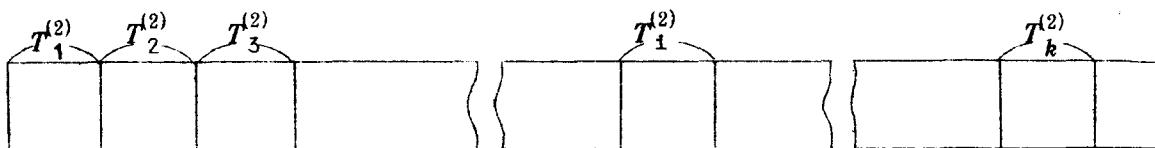


図 3.1 定理 3.1 の証明のチューリング機械
のワーキングテープ

ワーキングテープ $T^{(2)}$

の部分集合を貯え得る有限な記憶 M があるものとする。 $[T_{ik}^{(1)}]$, $[T_k^{(2)}]$ 、および $[M]$ でそれぞれ $T_{ik}^{(1)}$, $T_k^{(2)}$ 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 $[T_{ik}^{(1)}]$ は $N(i, k)$ と i -シンボルであり、 $[T_k^{(2)}]$ および $[M]$ は中間結果となる。なお記号 \square 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 $T_{k-1}^{(1)}$ 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 \triangle 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 \circ は i 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 $T_k^{(2)}$ 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 $T^{(2)}$ のコマはすべてプランク (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 $Y \rightarrow a_1$ であるようなすべてのルールの nt-シンボル Y を $T_1^{(1)} = T_{11}^{(1)}$ に書く。したがつて、

$$[T_{11}^{(1)}] := N(1, 1) \quad \{ a_1 \}$$

になる。 $T_{11}^{(1)}$ に i -シンボル S が表われれば、出力テープに出力の最初として "1" を、なければ "0" を書く。 $T_{11}^{(1)}$ 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 $T_{11}^{(1)}$ 上に、テープ $T^{(2)}$ のヘッド H_2 は $T_1^{(2)}$ 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, \dots, a_{k-1} が既に読み込まれ、

$$[T_{ik'}^{(1)}] = N(i, k) \quad (1 \leq i \leq k' < k)$$

とし、 $T_{k'-1}^{(1)}$ と $T_{k'}^{(1)}$ ($2 \leq k' \leq k$) 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 $T_{k'-1}^{(1)}$ 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 $T_{k'-1, k-1}^{(1)}$ 上に、 $T^{(2)}$ のヘッド H_2 は $T_1^{(2)}$ 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 $T_1^{(2)}$ を M に書き込む。したがつて

$$[T_1^{(2)}] = [M] = N(k, k) \quad \{a_k\}$$

となる。 H_2 を $T_2^{(2)}$ へ動かし、記号 △を書く。 H_1 で $T_{k'-1, k-1}^{(1)}$ を読み、 $T_{k'-1, k-1}^{(1)}$ 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、

$$[T_2^{(2)}] = N(k-1, k) \quad (\triangle \text{は除く})$$

となる。 H_1 を $T_{k-2, k-1}^{(1)}$ に H_2 を $T_3^{(2)}$ に動かす。

(22) $T^{(2)}$ のヘッド H_2 が $T_i^{(2)}$ ($2 \leq i < k$) にあり、

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

であるし、 $T^{(1)}$ のヘッド H_1 は $T_{k+1-i, k-1}^{(1)}$ にあるとする。このとき $T_{k+1-i, k}^{(1)}$ 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 $T_i^{(2)}$ に書き込む。したがつて

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

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 $T_{i-1}^{(2)}$ ($2 \leq i \leq k$) にあり、

$$[M] = N(k', k) \quad (1 \leq k' < k)$$

* $T^{(2)}$ 上でその右側に区画が作られてない場合も右に動かす。

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

であるとする。

(2.3.1) H_1 が $T_{k-i, k'}^{(1)}$ にあるとき、 $T_{k-i, k'}^{(1)}$ 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 $T_1^{(2)}$ に書き加える。すなわち、 $T_1^{(2)}$ にあらかじめ $N_{k'}(k-i+1, k)$ 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 $T_k^{(2)}$ 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、そこには△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 $T_{1k}^{(1)}$ が求められたことになる)。すなわち

$$[T_{1k}^{(1)}] = [T_k^{(2)}]$$

である。 \circlearrowleft のある区画すなわち $T_k^{(2)}$ 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 $T_{k-1}^{(2)}$ の内容を $T^{(1)}$ 上に書き込む。 $T_{k-1}^{(2)}$ 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 $T_k^{(1)}$ として、 $T_{1k}^{(1)}, T_{2k}^{(1)}, \dots, T_{kk}^{(1)}$ が書かれたことになる。 $T_{kk}^{(1)}$ の右隣りのコマに □を書く、 H_1 は $T_{kk}^{(1)}$ に、 H_2 は $T_1^{(2)}$ におく、 $T_{1k}^{(1)}$ に 1 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 $\alpha = a_1 a_2 \cdots a_n$ ICに対しては、上の操作を $k = 1$ から $k = n$ まで行えれば終了する。

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

一般の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ゐる。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

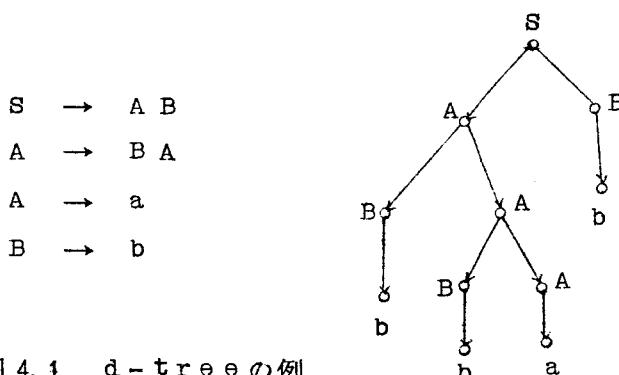


図 4.1 d-tree の例

* 構文解析をノーマル文法を使って、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 に對応して、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 ($1 \leq i, j \leq n$) に矢印が向く枝とから成り、各枝は t-シンボルあるいは nt-シンボルを名前としても

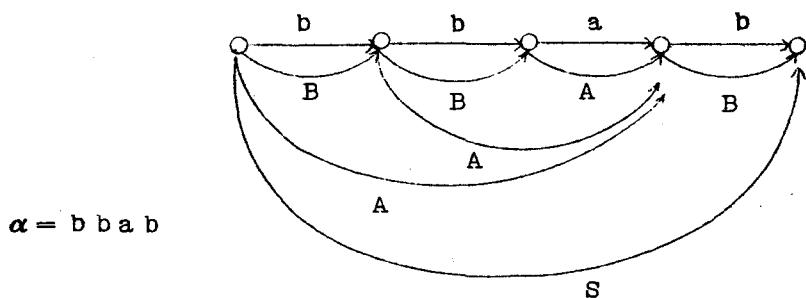


図 4.2 グラフ D の例

つ。そのうち入力系列を作るシンボル a_i ($1 \leq i \leq n$) は枝 $\overrightarrow{v_{i-1} v_i}$ の一つに相当し、他の枝はすべて nt-シンボルである。さらに g が適用されたルールのとき、 $\gamma(g)$ を枝の名前の系列としてもつ道の両端の節点が名前 $\ell(g)$ 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 に對応するが、同時に d-seq. とも 1 対 1 に對応し、ある入力系列 α に對して d 個の異なる d-seq. があるとき、入力系列 α に對して、 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、任意の入力系列に對して、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 - 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 $L(G)$ についてつきのような認識のアルゴリズムが得られる。

入力系列を $\alpha = a_1, a_2, \dots, a_n$ 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 $r(g)$ 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 $r(g)$ をもつ向きをもつ枝を書き入れる。（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 $R_i = R_{i1}$ 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 t -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。ここで 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 $d\text{-seq.}$ 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 $\ell(g)$ を枝の名前としたが、本節では便宜上 $\ell(g)$ 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 $r(g)$ はさらに、 $r_1(g)$, $r_2(g)$ 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。 $r_2(g) = 1$ 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_j^i で表わし、 E_j^i を貯える記憶部分には、 $g \in E_j^i$ および記号 \triangle 、 \circ 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、 $r_1(g)$ と $r_2(g)$ とがそれぞれ枝の名前の系列 g' , g'' の $\ell(g)$, $\ell(g'')$ に一致するとき、 g' および g'' に關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、 \circ は PD に何も入っていないときの $[PD]$ 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) $n \rightarrow i$, $n \rightarrow j$ とする。
- (2) E_j^i の内、 \triangle がなく、 $\ell(g)$ ($g \in E_j^i$) が S である枝に \triangle を附し、
 $d.s.=g$ とする。 $r_1(g) \rightarrow \alpha$, $r_2(g) \rightarrow \beta$ 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) $1 \rightarrow k$ として、 $\alpha = \ell(g')$, ($g' \in E_k^k$) なる g' を求め、 \circ を附す。ただし、既に g' に \triangle があれば \circ はつけず E_k^k 内の他の枝の名前に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 $k+1 \rightarrow k$ として、

(3.1) をくり返す。

(3.2) $\beta = \ell(g'')(g'' - E_{i-k}^i)$ なる g'' を求め \triangle を附し、 E_k^k 内の \circ を \triangle に書きかえる。ただし g'' に既に \triangle があれば、 E_{i-k}^i 内の他の枝についてこの操作を行う。 E_{i-k}^i 内にそのような枝がなければ $k+1 \rightarrow k$ とし E_k^k 内の \circ を消し (3.1) を行う。 $r_2(g')$ がなければ (5) を行う。

(4) $d \cdot s. = g'$, $r_1(g) \rightarrow \alpha$, $r_2(g'') \rightarrow \beta$ 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 $d\text{-seq.}$ になる。したがつてつぎに異なる $d\text{-seq.}$ を求めるため $\mathcal{C}(1)$ を行う。 $[PD] \neq \sigma$ ならば、 $[PD]$ のルール名 g_p を $d \cdot s.$ とし、 $r_2(g_p)$ が存在しなければ (5) を行う。存在すれば $r_1(g_p) \rightarrow \alpha$, $r_2(g_p) \rightarrow \beta$ 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 $I \rightarrow i$, $J \rightarrow j$ として (3.1) を行う。

以上の操作は、つぎのことを行つてゐることであり、あらゆる $d\text{-seq.}$ 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 $d\text{-tree}$ 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、 $r_1(g_1)$ を導くルール g_2 を求めて $r_2(g_1)$ を導くルールを PD に貯えておく。さら $\mathcal{C} r_1(g_1)$ を導くルール g_3 を求めて $r_2(g_2)$ 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 $d\text{-seq.}$ になつてい

る。一つの $d\text{-seq.}$ が求められたあと、 i -シンボル S の位置に戻り、 Δ 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 $d\text{-seq.}$ 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 Δ が附されれば操作は終り全ての異なる $d\text{-seq.}$ 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、 $d\text{-seq.}$ の一つのルールを求めるのに $C_2 n$ だから一つの $d\text{-seq.}$ を求めるには $c_3 n^2$ となり、 N 個の異なる $d\text{-seq.}$ 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 $E_j^{\frac{1}{n}}$ に割りあてるものとしている。したがつて各 $E_j^{\frac{1}{n}}$ 内にはたかだか nt -シンボルの数が貯えられ、 $0 \leq i \leq n-1$, $1 \leq j \leq n$ 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 $c n^2$ である。

§ 4.4 CSLのある部分族の認識への拡張

ルールが

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

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

の他に

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

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 L_S とする。このとき、 $L \in L_S$ の認識のアルゴリズムは、 つきのよう Δ にグラフ的に書くことができる。

$a_i \in V_T$ ($1 \leq i \leq n$) 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 $v_{i-10}^{(k_0)}$ 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。ただし、 $A_i \rightarrow a_i$ 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 k_{j-1} ($1 \leq j < i$, $3 \leq i$) まで作られたとする。各ルール g について $v_{ij-1}^{k''_{j-1}}$ ($k''_{j-1} \leq k_{j-1} \leq 1, 2, \dots$) 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 $v_{ij-1}^{k_j}$ 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 $v_{ij}^{k_j}$ ($k_j = 1, 2, \dots$) 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とはしない。この結果得られたグラフを k_{i0} とする。

(4) $k_1 = k_{10}$ ただし $i = 1, 2$

$= k_{11}$ ◇ $i = 3$

$= k_{11-3}$ ◇ $i \geq 4$

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 -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_s に含まれることになり、そのような枝がなければ、 α は L_s に含まれない。このことは CFL の認識の際と同じよう α は L_s に含まれる。このことは CFL の認識の際と同じよう α は $d\text{-tree}$ 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る。

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

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

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 $n (\geq 3)$ 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}_B になつてゐる。

§ 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 $c_1 n^2$ 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 $c_2 n^3 + c_3 n^2 N$ で終了することが示された。ここで c_1 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 $c_4 n(n-1)/2$ より極めて少い場合も稀ではないことを考えて、単位記憶内各 i, j, k , をそのまま貯えるようすれば、明らかに記憶は $c_5 n^3$ 必要で計算のための手数の上限は $c_6 n^3 + c_7 n \log n \cdot N$ 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 $1 \leq i, j, k \leq n$ が貯えられるために $\log n$ 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、C S L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 C S L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。

* C S L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、少なくとも 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 $V_{T_0} = \{A_1, A_2, \dots, A_{t_0}\}$ を t-シンボルの集合とし、Sを i-シンボル、 V_{N_0} 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 ($1 \leq p \leq t_0$) とする t_0 個のリニア文法 $G_p = \{V_{N_p}, V_{T_p}, P_p, A_p\}$ ($1 \leq p \leq t_0$) を考える。ここ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_{N_0} および $G_{p'}$ ($p \neq p'$, $1 \leq p' \leq t_0$) の nt-シンボルとも異なるものとする。このとき

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

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 を \mathcal{L} についてつきの定理が成立する。

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ャ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

$$Y \rightarrow XA$$

$Y \rightarrow a$

($X, X \in V_{N_0}, a \in V_{T_0}$)

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

$Y \rightarrow X a$

$Y \rightarrow a X$

$Y \rightarrow a$

($Y, X \in V_{N_p}, a \in V_{T_p} : 1 \leq p \leq t_0$)

のいずれかであるものとできる。[※] なお簡単のため

$$V_{T_p} = V_{T_{p'}} = V_T \quad (p \neq p', 1 \leq p \leq t_0, 1 \leq p' \leq t_0)$$

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、

$$N(i, k) = \{ Y \mid Y \xrightarrow{G_p} a_i a_{i+1} \cdots a_k, Y \in V_{N_p}, 1 \leq p \leq t_0 \} \\ (1 \leq i \leq k)$$

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

$$X \xrightarrow{G_p} a_i \cdots a_{k-1}, Y \rightarrow X a_k \in P$$

または

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

したがつて、 $1 \leq i < k$ 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 $N(i, k-1)$ ($1 \leq i \leq k-1$) と $N(k, k)$ とがわかれれば、 $N(k-1, k), N(k-2, k), \dots, N(1, k)$ 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。つきに、

$$M(k) = \{ Y \mid Y \xrightarrow{G_p} a_1 a_2 \cdots a_k, Y \in V_{N_0} \}$$

ここで $Y \xrightarrow{G_p} \varphi$ は $Y \xrightarrow{G_0} \varphi_1, \varphi_1 \xrightarrow{G_p} \varphi$ 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。 $Y \xrightarrow{G_p} a_1 a_2 \cdots a_k$ ($Y \in V_{N_0}$) ならば、ある i について

$$X \xrightarrow{G_0} a_1 a_2 \cdots a_i \quad (X \in V_{N_0})$$

$$Y \rightarrow X A_p \in P_0$$

$$A_p \xrightarrow{G_p} a_{i+1} \cdots a_k$$

である。したがつて $M(i)$ と $N(i+1, k)$ ($1 \leq i \leq k-2$) とがわかれれば $M(k)$ 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 $N(i, k')$ ($1 \leq i \leq k' < k$) を貯えておく必要はない、 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 $M(i)$ ($1 \leq i \leq k-1$) を貯える必要がある。これらを貯えるテーブ T は明らかに k 回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 ($1 \leq i \leq k-1$) 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 ($1 \leq i \leq k-1$) 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。これで

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

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 -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 $C_3 k$ である。ただし 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 $C_4 n^2$ 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。

たとえばテーションボルを 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。^{❀❀}

* $L_Y = \{ S\varphi_1 S\varphi_2 S\varphi_3 SR(\varphi_2) S\varphi_4 S | \varphi_1, \varphi_2, \varphi_3, \varphi_4 \in \{0, 1\}^* \}$ とかける。ただし $R(\varphi_2)$ は φ_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 $\varphi_2, R(\varphi_2)$ に、そして Z は φ_3 に直接関連するシンボルである。

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

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

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

$$P : \quad s \rightarrow sXs, \quad X \rightarrow 0X, \quad X \rightarrow 1X$$

$$X \rightarrow sX, \quad X \rightarrow X0, \quad X \rightarrow X1$$

$$X \rightarrow Xs, \quad X \rightarrow sys,$$

$$Y \rightarrow 0Y0, \quad Y \rightarrow 1Y1, \quad Y \rightarrow sys$$

$$Y \rightarrow sZs,$$

$$Z \rightarrow 0Z, \quad Z \rightarrow 1Z, \quad Z \rightarrow sz$$

$$Z \rightarrow Z0, \quad Z \rightarrow Z1, \quad Z \rightarrow Zs$$

$$Z \rightarrow 0, \quad Z \rightarrow 1, \quad Z \rightarrow s$$

§ 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 $T_1, T_2, \dots, T_i, \dots$ 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 の各コマは最初プランクとする。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 は各 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。 $[T_{hi}]$ ($h=0, 1, i=1, 2, \dots$) 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 , 1 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、 1 はヘッドが T_1 にあるとき、 $N(1, k)$ が求められることを示す。さらに $N(1, k)$ 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行進する。

(1) 入力テープから最初の入力シンボル a_1 を読み、 $T_{01} \leftarrow *a_1$ と $FSG \quad G_0$ で $Y \rightarrow a_1$ 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 ($1 \leq p \leq t_0$) $Y \rightarrow a_1$ であるようなルール \leftarrow について、すべての nt-シンボル Y を $T_{11} \leftarrow$ 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 を含めれば出力テープ $\leftarrow 1 \rightleftharpoons$ を、含まれなければ $\leftarrow 0 \rightleftharpoons$ を書く。ヘッドを $T_{02} \leftarrow$ 動かす。

$$(2) [T_{01}] = *a_1$$

$$[T_{0k}] = a_k \cup M(k'-1) \quad (1 < k' < k)$$

$$[T_{0k}] = M(k-1)$$

$$[T_{1k}] = N(k', k-1) \quad (1 \leq k' < k)$$

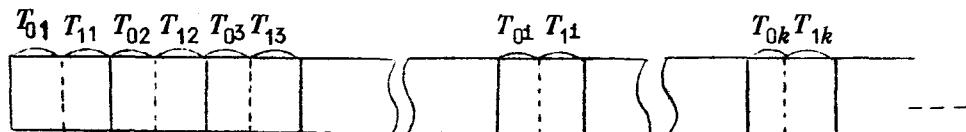


図 5.1 定理 5.1 の証明のチューリング機械の
ワーキングテープ T

であり、ヘッドが $T_{0,k}$ 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 $T_{0,k}$ に貯える。文法 G_p 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 を $T_{1,k}$ と M_2 に貯える。したがつて

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

である。ヘッドを再び $T_{0,k}$ 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 $X \rightarrow X_1 A_p$ であるようなすべてのルールの nt-シンボル X を M_3 に貯える。このときの $[M_3]$ は $M_{k-1}(k)$ である。ヘッドを $T_{1,k-1}$ に動かす。

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

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

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

$$[M_0] = a_k$$

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

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

であるとする。ここで $[M_3]$ 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。

$T_{1,i}$ 内の Y_1 について文法 G_p で $Y \rightarrow Y_1 a_k$ 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 $Y \rightarrow a_i Y_1$ 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 C F L 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-) リニア言語をその一部に含む C F L 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。

C F L の構文解析はランダム・アクセス記憶の必要量が $c_3 n^2$ で $c_1 n^3 + c_2 n^2 N$ を上限とする手数で終了することを示した。ここで n は文章の長さ、 N は異なる derivation の数、そして c_1 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 であるある CFL の部分族が n^2 -recognizable であることがわかつたが、 $\lim_{n \rightarrow \infty} T(n)/(\ln n)^2 = 0$ 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 $(\ln n)^2$ 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ゝや、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。

本研究に關し、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-

- (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 Mathematical 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

Pg. 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.Elbourne and W.H.Ware,"The evolution 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. 1964, 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.Petrick, "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 "Structure 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 computing time for recognition of languages generated by linear grammars," Rep.of Coordinated Science Lab.Univ.of Illinois, 1966. To appear 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