論　文

文脈自由言語の bottom-up 的最小誤り訂正アルゴリズムについて*

山崎　進** 外村俊之**

Abstract

On the assumption that three types of syntactic errors (the replacement of a symbol by an incorrect symbol, the insertion of an extraneous symbol, or the deletion of a symbol) debasing the strings of a language generated by a context-free grammar can occur arbitrarily many times, there are top-down algorithms that will correct any input string in the fewest possible number of errors (least error correction). However, there has not been any bottom-up algorithm correcting the above three types of errors.

In this paper a bottom-up least error correction algorithm for the three types of errors is presented which requires time proportional to the cube of the length of an input.

Also a bottom-up least error correction algorithm which works in less than cubic time is presented, in the case that the number of errors (which are in the three types) in an input can be bound out of the relation to its length.

1. まえがき

形式用語を記述する文法が与えられているときに，その言語に属する系列から誤りによって変形を受けた系列が入力された場合，その入力された系列のもつ誤りを検出したり，訂正する問題はいろいろな局面で重要である．

系列の誤りとしては，記号間の置き換え，記号の脱落，記号の挿入が記号間に独立に生起することによって生じるもの（独立誤り）が基本的であり，以下の議論ではこの独立誤りを考察の対象とする．

もとの言語が正規言語であれば，入力された系列の，有限個の誤りの検出が可能であるかどうかは可解である．もとの言語が文脈自由言語であれば，一般には有限個の誤りの検出が可能かどうかは可解でない[1]．したがって，文脈自由言語においては，入力系列の誤りを，誤り数が最小になるように訂正（最小誤り訂正）すること等が基本的な課題となる．

文脈自由言語における最小誤り訂正は，top-down的構文解析法[2]に基づく最小誤り訂正アルゴリズム[3),4)]によって，長さnの入力系列に対し，$O(n^3)$ の時間，$O(n^2)$ の記憶量で可能な（Random Access Machine-RAM で評価して）ことが示されている．一方 Younger のアルゴリズム[5]を基にした bottom-up 的手法によって最小誤り訂正法を議論した例はあるが[1),6)]，誤りとして，記号間の置き換え，記号の脱落，記号の挿入が，任意個生起し得ると仮定した場合の bottom-up 的手法についての議論はなく，この点での手法が興味をひく．

本稿では，まず文脈自由言語の最小誤り訂正の手法として，任意個の，記号間の置き換え，記号の脱落，記号の挿入によって生ずる誤りを仮定した場合の，bottom-up 的アルゴリズムを示し，長さnの系列に対し，$O(n^3)$ の時間と $O(n^2)$ の記憶量で誤り訂正が可能で（RAM で評価して）あることを述べる．つぎに，誤り数が，入力系列の長さnに無関係な定数以下に制限できる場合の bottom-up 的最小誤り訂正アルゴリズムを，Valiant によるアルゴリズム[9]を基に構成し，長さnの入力系列に対し $O(n^{2.81})$ の時間，$O(n^2)$ の記憶量で誤り訂正が可能であることを述べる．

---

* On a bottom-up least error correction algorithm for context-free languages by Susumu YAMASAKI and Toshiyuki TONOMURA (Department of Information Science, Faculty of Engineering, Kyoto University)

** 京都大学工学部情報工学教室

2. 言語の最小誤り訂正問題

以下の議論に必要な諸定義，性質を述べ，(文脈自由) 言語の最小誤り訂正問題の形式的定義を述べる．

[定義 1] 文脈自由文法 (cfg) とは，$G=(V_N, \Sigma, P, S)$ である．ここに V_N は非終端記号の有限集合，Σ は終端記号の有限集合であり，$P\subseteq V_N\times(V_N\cup\Sigma)^*$ (生成規則の有限集合) で，S は初期記号である．生成規則は $A\to\nu$, $A\in V_N$, $\nu\in(V_N\cup\Sigma)^*$ のように表現される．$\xi,\eta\in(V_N\cup\Sigma)^*$ に対して，$\xi=\xi_1A\xi_2$, $\eta=\xi_1\nu\xi_2$, $\xi_1,\xi_2\in(V_N\cup\Sigma)^*$, $A\to\nu$ ならば $\xi\underset{G}{\Rightarrow}\eta$ と表わす．$\xi\underset{G}{\Rightarrow}\eta$ または $\xi=\xi_0$, $\eta=\xi_r$, $\xi_i\underset{G}{\Rightarrow}\xi_{i+1}$, $0\leq i\leq r-1$ ならば $\xi\overset{*}{\underset{G}{\Rightarrow}}\eta$ と表わす．$L(G)=\{w\in\Sigma^*|S\overset{*}{\underset{G}{\Rightarrow}}w\}$ を G によって生成される言語という．cfg によって生成される言語を文脈自由言語 (cfl) という．

[定義 2] cfg $G=(V_N, \Sigma, P, S)$ において，$P\subseteq V_N\times\{V_N{}^2\cup\Sigma\}$ のとき，G は Chomsky 標準形にあるという．

次の定理はよく知られている．

[定理 1][7] 任意の cfg G に対し，$L(G)-\{\varepsilon\}=L(G_1)$ となる Chomsky 標準形の cfg G_1 が存在する．

[定義 3] cfg $G=(V_N, \Sigma, P, S)$ において $^\forall A\in V_N$ に対し，(i) $^\exists\xi,\eta\in(V_N\cup\Sigma)^*$, $S\overset{*}{\underset{G}{\Rightarrow}}\xi A\eta$, (ii) $^\exists w\in\Sigma^*$, $A\overset{*}{\underset{G}{\Rightarrow}}w$ なるとき，G は縮約されている (reduced) という．

つぎの性質は定理 1 と Ginsburg[8] にしたがえば得られる．

[定理 2] 任意の cfg G に対し，$L(G)-\{\varepsilon\}=L(G_1)$ となる reduced Chomsky 標準形の cfg G_1 が存在する．

本稿では，言語の誤りとしてつぎのものを扱う．

[定義 4] (言語の独立誤り) ある言語に属する系列に対し，つぎの 3 種類の誤りが，記号間に独立に起こるものとする．(i) ある記号が他の記号へ置き換わる誤り (E_R と略記する)．(ii) ある記号がもとの系列中に挿入されることによって生ずる誤り (E_I と略記する)．(iii) 系列上のある記号が削除されることにより生ずる誤り (E_D と略記する)．

系列間のハミング距離を次のように定義し，系列に独立誤りが生ずるごとに，もとの系列と誤りのある系列の距離は 1 だけ増すと考えて，1 つの系列とそれから誤りによって生じた系列とのハミング距離によって誤り数を与えるものとする．

[定義 5] $x=a_1a_2\cdots a_n(a_i\in\Sigma, 1\leq i\leq n)$, $y=b_1b_2\cdots b_m(b_j\in\Sigma, 1\leq j\leq m)$ のとき ($n=0$ または $m=0$ のときそれぞれ $x=\varepsilon$ または $y=\varepsilon$ と考える)，x と y のハミング距離，$D(x, y)$ は次のように与える．

$D(x,y)=D(x_n, y_m)$

ここに，$x_i=a_1a_2\cdots a_i$, $y_j=b_1b_2\cdots b_j$

$$D(x_i, y_j)=\min\{D(x_{i-1}, y_{j-1})+\Delta_{ij}, D(x_{i-1}, y_j)+1, D(x_i, y_{j-1})+1\}$$

$D(x_i,\varepsilon)=i$, $D(\varepsilon, y_j)=j$, $D(x_0, y_0)=0$

$$\Delta_{ij}=\begin{cases}1, & a_i\neq b_j \text{ のとき}\\ 0, & a_i=b_j \text{ のとき}\end{cases}$$

とする．

(文脈自由) 言語の最小誤り訂正問題を次のように考える．

[定義 6] 文法 (cfg) G とそれによって生成される言語 $L(G)$ が与えられたとき，$L(G)$ の最小誤り訂正問題とは，次のことを言う．すなわち，ある系列 (入力される系列) w に対し，$L(G)_w\triangleq\{w'|D(w', w)\leq D(^\forall w'', w), w', w''\in L(G)\}$ とするとき，ひとつの $w'\in L(G)_w$ を求めること．

3. cfl の bottom-up 的最小誤り訂正アルゴリズムについて

3.1 一般的な場合のアルゴリズム

3.1.1 最小誤り訂正アルゴリズム

[アルゴリズム 1] cfg $G_0=(V_N, \Sigma, P, S)$ (定理 2 によって $L(G_0)-\{\varepsilon\}$ を生成する reduced†† Chomsky 標準形 cfg G を仮定できる) が与えられたとき，

[入力] 系列 $w=a_1a_2\cdots a_n$,

に対し，

[出力] $L(G_0)_w$ の 1 つの要素

を得るアルゴリズムはつぎの通りである．

[方法]

(1) cfg G に対し，つぎのような cfg G' を構成し，cfg G' の非終端記号が生成する最短系列を求める．

(cfg G' の構成) $G'=(V_N\cup\{I\}, \Sigma, P\cup P, S)$ とする．ここに I は $I\notin V_N$ なる記号で，P' は，$^\forall A_i\in V_N$ に対し，$A_i\to IA_i$, $A_i\to A_iI$ なる生成規則，$I\to a(^\forall a\in\Sigma)$ なる生成規則，$I\to II$ なる生成規則からな

---

† あるアルファベット V に対し，V^* は長さ 0 の系列 (空文) ε を含む，V から作られた任意の系列の集合を示し，$V^+\triangleq V^*-\{\varepsilon\}$ とする．

†† cfg が reduced であることを仮定するのは，後の手続きにおいて (認識行列の構成において)，無駄を省くためである．

る．

（非終端記号が生成する最短系列を求めること）

V_N に属する非終端記号のおのおのに対して，それが生成する最短系列とその長さを求める（これはおのおのの非終端記号に対して，生成規則を適用し，導出過程において同じ非終端記号が繰り返して現われないようにした場合に生成できる終端記号列のうち最も短いもののうちの一つを選ぶことによって求まる）．おのおのの非終端記号と，それから生成できる最短系列の長さ，非終端記号から最短系列への構文（木）を対応づけて表にまとめる．表は3つの項目からなる．これを以後，DT 表とよび，$[A, l, A\overset{*}{\underset{G}{\Rightarrow}}w_A]\in \mathrm{DT}$ のように記して，非終端記号 A から長さ l の最短系列 w_A が生成できて，DT 表に，3つの項目としてそれがまとめられていることを示す．

（I および P' を cfg G に付加して cfg G' を構成することは，入力系列に挿入誤りがあったと仮定する場合の訂正に関係する．DT 表は入力系列に削除誤りがあると仮定する場合の訂正に使う．）

（2）　2字組 $[V,e]$ を考える，ここに V は $V_N\cup\{I\}$ の要素，e は非負整数である．（2字組 $[V,e]$ は入力系列の部分系列が V に bottom-up されると予想したときの誤り数が e であることを示す．）

（3）　$\{[V,e]\mid V\in V_N\cup\{I\}$, e: 非負整数$\}$ を要素とする認識行列 $[t_{ij}]$ $(1\leqq i\leqq n,\ 1\leqq j\leqq n-i+1)$（Younger の認識行列を変形したもの）を次のように構成する．（次の（i）(b)，（ii）(b)の手続きが，従来になかった，E_D に対する処理に対応する．）

（i）（a）　おのおのの $1\leqq i\leqq n$ について，$(A\to a_i)\in P$ なる A に対して $[A,0]\in t_{i1}$，$B\to a\in P$, $a\neq a_i$ なる B に対して $[B,1]\in t_{i1}$，また $[I,1]\in t_{i1}$ とする．（b）おのおのの $1\leqq i\leqq n$ について，$[A,e]\in t_{i1}$，$[B,l,B\overset{*}{\underset{G}{\Rightarrow}}w_B]\in \mathrm{DT}$，$(C\to AB)\in P\cup P'$ または $(C\to BA)\in P\cup P'$ のとき，$[C,e+l]\in t_{i1}$ とする．

（c）（b）の手続きにおいて，2字組の前の部分が同じで，誤り数が大きいものは加えない．

（ii）（a）　$[A,e]\in t_{ik}$ $(1\leqq i\leqq n,\ 1\leqq k\leqq j-1)$，$[B,f]\in t_{i+k,j-k}$$(1\leqq i+k\leqq n, 1\leqq k\leqq j-1, i+1\leqq j\leqq n-i+1)$，$(C\to AB)\in P\cup P'$ のとき，$[C,e+f]\in t_{ij}$ とする．（b）　$[A,e]\in t_{ij}$$(1\leqq i\leqq n, 1\leqq j\leqq n-i+1)$，$[B,l,B\overset{*}{\underset{G}{\Rightarrow}}w_B]\in \mathrm{DT}$，$(C\to AB)\in P\cup P'$ または $(C\to BA)\in P\cup P'$ のとき，$[C,e+l]\in t_{ij}$ とする．（c）（a）(b)の手続きにおいて2字組の前の部分が同じで，誤り数が大きいものは加えない．誤り数の小さいものがあると，誤り数が大きいものは除く．

（iii）　t_{1n} の要素としては，初期記号 S を含むものに限る．

（4）（3）で構成した認識行列 $[t_{ij}]$$(1\leqq i\leqq n,\ 1\leqq j\leqq n-i+1)$ から，次の方法で，構文行列 $[u_{ij}]$$(1\leqq i\leqq n,\ 1\leqq j\leqq n-i+1)$ を構成する．

（i）　$t_{1n}=u_{1n}$ とする．（ii）　$[C,e+f]\in u_{ij}$$(1\leqq i\leqq n,\ 1\leqq j\leqq n-i+1)$，$[A,e]\in t_{ik}$$(1\leqq i\leqq n, 1\leqq k\leqq j-1)$，$[B,f]\in t_{i+k,j-k}$$(1\leqq i+k\leqq n,\ 1\leqq k\leqq j-1,\ i+1\leqq j\leqq n-i+1)$ で，$(C\to AB)\in P\cup P'$ ならば $[A,e]\in u_{ik}$，$[B,f]\in u_{i+k,j-k}$ とする．（iii）　$[C,e+l]\in u_{ij}$$(1\leqq i\leqq n,\ 1\leqq j\leqq n-i+1)$，$[B,l,B\overset{*}{\underset{G}{\Rightarrow}}w_B]\in\mathrm{DT}$で，$(C\to BA)\in P\cup P'$ または $(C\to AB)\in P\cup P'$ ならば，$[A,e]\in u_{ij}$ とする．

（5）　構文行列 $[u_{ij}]$$(1\leqq i\leqq n,\ 1\leqq j\leqq n-i+1)$ より，1つの構文（木）を得ながら（この方法は文献10における手法を少し変形して適用できる）誤りを訂正する．ここで，cfg G' における生成規則には番号付けが行われていて，DT 表における非終端記号から最短系列の生成も含めて，構文(木)は適用する生成規則の順序で示されるものとする．

（i）　$g(i,j,[C,e+f])$$(1\leqq i\leqq n,\ 1\leqq j\leqq n-i+1)$ を，$C\overset{*}{\underset{G'}{\Rightarrow}}u(u\in\Sigma^*)$，$D(u,a_ia_{i+1}\cdots a_{i+j-1})=e+f$ なるような A から u への最左導出構文(木)を与え，$a_ia_{i+1}\cdots a_{i+j-1}$ を u に訂正するルーチンとし，次のように帰納的に定義される．

（a）　$j=1$ のとき（$[C,e+f]\in u_{i1}$ を仮定して）$(a-1)$ から $(a-4)$ のどれかを実行する；$(a-1)$ $(C\to c)\in P$ ならば，その生成規則の番号を与え，$c\neq a_i$ のとき a_i を c に訂正する（$c=a_i$ のときはそのまま），$(a-2)C=I$ のとき，$I\to a_i$の生成規則の番号を与え a_i を ε に訂正する，$(a-3)(C\to AB)\leqq P\cup P'$，$[A,f,A\overset{*}{\underset{G}{\Rightarrow}}w_A]\in\mathrm{DT}$ のとき，$A\overset{*}{\underset{G}{\Rightarrow}}w_A$ の構文の次に $C\to AB$ の番号を与え，a_i の前に w_A を挿入し，$g(i,1,[B,e])$ をよぶ．ここで $C\to AB$ なる規則は適当に1つ選べば良い．$(a-4)(C\to BA)\in P\cup P'$，$[A,f,A\overset{*}{\underset{G}{\Rightarrow}}w_A]\in\mathrm{DT}$ のとき，$C\to BA$ の番号を与え，$g(i,1,[B,e])$ をよぶ，そして $g(i,1,[B,e])$ の実行のあとに，$A\overset{*}{\underset{G}{\Rightarrow}}w_A$ の構文を与え，$g(i,1,[B,e])$ によって a_i の前後に施される訂正のすぐ後に w_A を挿入する．ここで $C\to BA$ なる生成規則は適当に1つ選べば良い．

（b）　$j>1$ のとき（$[C,e+f]\in u_{ij}$ を仮定して），

次の $(b-1)$，$(b-2)$，$(b-3)$ のどれかを実行する；$(b-1)[A,e]\in u_{ik}, [B,f]\in u_{i+k,j-k}, (C\to AB)\in P\cup P'$ となる適当な k を求め，$C\to AB$ の番号を与え，$g(i,k,[A,e])$，次に，$g(i+k,\ j-k,[B,f])$ を実行する．ここで $C\to AB$ なる規則は適当に1つ選べば良い，$(b-2)(C\to AB)\in P\cup P', [A,f,A\overset{*}{\underset{G}{\Rightarrow}}w_A]\in \mathrm{DT}$ のとき，$C\to AB$ の番号，つづいて $A\overset{*}{\underset{G}{\Rightarrow}}w_A$ の構文を与えて，入力系列 $a_ia_{i+1}\cdots a_{i+j-1}$ の前に w_A を挿入し，$g(i,\ j,[B,\ e])$ を実行する．$(b-3)(C\to BA)\ni P\cup P', [A,\ f,\ A\overset{*}{\underset{G}{\Rightarrow}}w_A]\in \mathrm{DT}$ のとき，$C\to BA$ の番号を与えて，$g(i,j,[B,e])$ を実行する，そしてその後に $A\overset{*}{\underset{G}{\Rightarrow}}w_A$ の構文を与えて，入力系列 $a_ia_{i+1}\cdots a_{i+j-1}$ が訂正された系列の後に w_A を挿入する．

(ii) $g(1,n,[S,e])([S,e]\in u_{1n})$ を実行する．

(6) $g(1,n,[S,e])$ が w' を出力したとする．次の手続きを行う．(i) $D(w',w)\leqq(w$ の長さ$)$ か $L(G_0)\not\ni\{\varepsilon\}$ のときは，w' を求める訂正系列とする．(ii) $D(w',w)>(w$ の長さ$)$ かつ $L(G_0)\ni\{\varepsilon\}$ のときは，ε を求める訂正系列とする（構文は無視する）．

[例 1] cfg $G=(V_N,\Sigma,P,S)$: $V_N=\{S,A,B,C\}$ $\Sigma=\{a,b\}$，$P=\{$①$S\to SS$，②$S\to AB$，③$S\to AC$，④$C\to SB$，⑤$A\to a$，⑥$B\to b\}$ とし，$w=aabab$ を入力系列とした場合のアルゴリズム1の適用を例示する．

G は reduced Chomsky 標準形になっている．$G'=(V_N\cup\{I\},\Sigma,P\cup P',S)$: $P'=\{$⑦$S\to IS$，⑧$S\to SI$ ⑨$A\to IA$，⑩$A\to AI$，⑪$B\to IB$，⑫$B\to BI$，⑬$C\to IC$，⑭$C\to CI$，⑮$I\to II$，⑯$I\to a$，⑰$I\to b\}$ となる．DT 表は **Table 1** のようになる．

認識行列 $[t_{ij}]$ $(1\leqq i\leqq 5,\ 1\leqq j\leqq 5-i+1)$ は **Table 2** のようになる．'*' のついた要素は，アルゴリズム (3) の (i)(b)，(ii)(b) の手続きによっても生成されるものである．

構文行列 $[u_{ij}]$ $(1\leqq i\leqq 5,\ 1\leqq j\leqq 5-i+1)$ は Table 2 の $[t_{ij}]$ において '+' のついた要素を除いたものである．

Table 1 Nonterminal and the shortest string generated by it

非終端記号	最短系列長	最短系列の構文（最左導出）
S	2	$S\overset{*}{\Rightarrow}ab$ (②⑤⑥)
A	1	$A\to a$ (⑤)
B	1	$B\to b$ (⑥)
C	3	$C\overset{*}{\Rightarrow}abb$ (④②⑤⑥⑥)

Table 2 Recognition matrix t_{ij} and parsing matrix u_{ij} for $w=aabab$

[A, 0] [B, 1] [I, 1] *[S, 1] *[C, 2]+	[A, 1] [B, 2]+ [I, 2]+ [S, 1]+ [C, 2]+	[A, 2]+ [B, 2]+ [C, 1]+ [S, 1] [I, 3]+	[A, 3]+ [B, 3]+ *[C, 2]+ [S, 1]+ [I, 4]+	[S, 1]
[A, 0] [B, 1]+ [I, 1] *[S, 1] *[C, 2]+	[A, 1] [B, 1] [C, 1] [S, 0] [I, 2]+	[A, 2]+ [B, 2]+ [C, 1]+ [S, 1] [I, 3]+	[A, 3]+ [B, 3]+ *[C, 1] [S, 1] [I, 4]+	
[A, 1] [B, 0] [I, 1] *[S, 1]+ *[C, 2]+	[A, 1]+ [B, 1] [C, 2]+ *[S, 2]+ [I, 2]+	[A, 2]+ [B, 2]+ *[C, 2]+ [S, 1]+ [I, 3]+		
[A, 0] [B, 1]+ [I, 1] *[S, 1] *[C, 2]	[A, 1]+ [B, 1] [S, 0] [C, 1]+ [I, 2]+			
[A, 1]+ [B, 0] [I, 1]+ *[S, 1]+ *[C, 2]+				

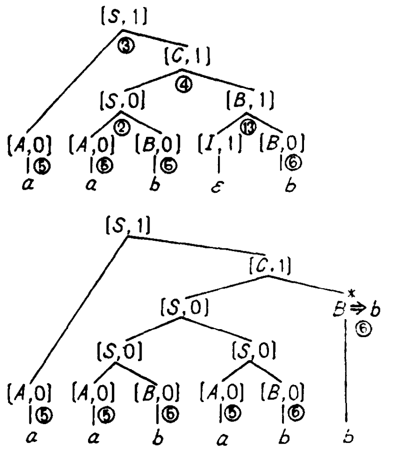

Fig. 1 Example of least error corrections for $w=aabab$

$g(1,5,[S,1])$ を実行すると **Fig. 1** のようになり（2例を示す―これは $g(i,j,[c,e+f])$ の定義において選択の余地があるからであり，実際上は優先度をつけて択一にする），最小誤り訂正系列 $aaab$ または $aababb$ を得る．Fig. 1 での番号は生成規則の番号で

ある.

3.1.2 アルゴリズム1の正当性とその評価

[定理 3] アルゴリズム1によって最小誤り訂正が可能である.

[証明] (1)アルゴリズム1は, 誤り E_R (誤りのない場合も含めて) に対しては, Younger の認識行列, 構文行列の構成[5], 文献 10 の構文を出す方法に, 誤り数の加算という手続きを付加したものである. (2)誤り E_I に対しては, cfg G' を構成し, $I\to a(^\forall a\in\Sigma)$, $A\to IA$, $A\to AI\to P'(^\forall A\in V_N)$, $I\to II\in P'$ とすることによって, 正しい構文解析が入力系列中の挿入誤りの影響によって不可能になることを防いでいる. ルーチン $g(i,j,-)$ の実行においては I から ε を生成するとして, 挿入された記号を消すことになる. (3)誤り E_D に対して検討してみると, 削除された部分系列は P によってどれかの非終端記号に bottom-up されるはずである. どの非終端記号に bottom-up されたかによって, それから上のレベルの構文が決まる. したがって, 削除された部分系列が bottom-up される非終端記号を, 構文解析の適当な位置に付加してゆけば, 削除された部分系列を含む構文が得られる. 入力系列に対して, 正しいと予想する系列は削除されたと考える部分系列の分だけ長くなるので, 最小誤り訂正においては, 構文上正しい最短系列を付け加えれば良い. それはアルゴリズム1において, DT 表を作成すること, (3)の(i)(b), (ii)(b), (4)の(ii)(iii)の手続き, (5)の(i)(a-3), (a-4), (b-2), (b-3)の手続きに対応している.

(4) 誤り数に関する手続きは, 文献 1), 6)の考え方を(3)において E_D に対してまで拡張していて, 非終端記号が同じ誤り数が大きいものは, 認識行列構成のとき除く手続きは, アルゴリズム1の(2)で定義した2字組の直観的意味と系列間のハミング距離が Dynamic Programming によって定義されていることから必要である.

(5) $g(1,n,[S,e])$ が ε を与えるときの処理はアルゴリズム1の(6)で行われる.

(1)〜(5)から, アルゴリズム1の正当性が結論できる. (証明終)

[定理 4] アルゴリズム1によって, 入力系列の長さが n のとき, $O(n^3)$ の時間, $O(n^2)$ の記憶量で, 最小誤り訂正が可能である. ここに計算の複雑さは RAM によって評価し, 非負整数の加算減算は1ステップと見ている.

[証明] アルゴリズム1において(1)の手続きは有限回で終了し, (3)(i)(b), (ii)(b), (4)(ii), (iii)(5)(a-3)(a-4), (b-2)(b-3) は, DT 表が有限で, (3)(i)(c), (ii)(c) の制限があることにより, t_{ij}, u_{ij}, $g(i,j,-)$ のおのおのにつき, 有限の操作で終了するので, (3)(4)の手続きは時間 $O(n^3)$, (5)の手続きは時間 $O(n^2)$ で可能であることが, 文献 5), 10) との対応で示せる.

記憶量については t_{ij}, u_{ij} の要素の異なる個数は有限個しかないので, 全体として $O(n^2)$ あれば良い. (証明終)

3.2 誤り数が制限される場合の最小誤り訂正法

本節においては, 入力系列の誤り数がその系列の長さに無関係なある数で押えられる (と言える) 場合やあるいは, ある一定の(入力系列の長さに関係のない)誤り数を越えるときは誤り訂正が不要で, その一定の誤り数以下の範囲での最小誤り訂正だけを問題にしたい場合においては, 入力系列の長さ n に対し, $O(n^{2.81})$ の時間と $O(n^2)$ の記憶量で可能な最小誤り訂正法を述べる.

3.2.1 最小誤り訂正法

[アルゴリズム 2] cfg G_0 ($L(G_0)-\{\varepsilon\}$ を生成する reduced Chomsky 標準形 G を仮定する) が与えられたとき,

[入力] 系列 $w=a_1a_2\cdots a_n$, および一定の定数 γ (n と無関係な定数) に対して,

[出力] $L(G_0)_w{}^\gamma \triangleq \{w' \mid w'\in L(G_0)_w \& D(w',w)\leqq\gamma\}\neq\phi$ のとき, $L(G_0)_w{}^\gamma$ の1つの要素を与え, $L(G_0)_w{}^\gamma=\phi$ のとき 'No' をかえす,

アルゴリズムは次の通りである.

[方法]

(1) アルゴリズム1の(1)の手続きを行う.

(2) アルゴリズム1の(2)と同様の2字組を考える.

(3) 2字組 $[V,e]$ の集合を要素としてもつ $(n+1)\times(n+1)$ 行列を(iii)で述べるように定義し ((iii)(a)), (i)で定義する $\boxtimes_\gamma$ 演算の下での推多的閉包 ((ii)で定義) を求める ((iii)(b)).

(i) (行列演算 $\boxtimes_\gamma$ について) $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ が $(n+1)\times(n+1)$ 行列のとき $\boxtimes_\gamma$ をつぎのように定義する. すなわち,

$$\boldsymbol{A}\boxtimes_\gamma\boldsymbol{B}=\boldsymbol{C}\Leftrightarrow\boldsymbol{C}_{ij}=\mathop{\vee_\gamma}_{k=1}^{n+1}(\boldsymbol{A}_{ik}\otimes\boldsymbol{B}_{kj})\quad(1\leqq i,j\leqq n+1)$$

⊗は(a)で定義し, $\vee_\gamma$ は(b)で定義する.

（a） 演算$\otimes$は(a-1)，(a-2)をまとめて(a-3)のように定義する．(a-1) $[A,e]\in\boldsymbol{A}_{ik}$, $[B,f]\in\boldsymbol{B}_{kj}$ かつ $(C\to AB)\in P\cup P'$ ならば $[C,e+f]\in\boldsymbol{A}_{ik}\otimes\boldsymbol{B}_{kj}$ とする．(a-2) $[A,e]\in\boldsymbol{C}_{ij}$, $[B,l,B\overset{*}{\underset{G}{\Rightarrow}}w_B]\in\mathrm{DT}$, $(C\to AB)\in P\cup P'$ または $(C\to BA)\in P\cup P'$ のとき，$[C,e+l]\in\boldsymbol{C}_{ij}$ とする．(a-3)(a-3-1) cfg G' の $P\cup P'$ と DT 表を参照しながら表（DT 1 表とよぶ）を構成する．すなわち，$V_N\cup\{I\}$の要素Aに対して，第1列目にAを書き，$(A_1\to AB_1)\in P\cup P'$ または $(A_1\to B_1A)\in P\cup P'$で，$A_1\neq A$ なる $A_1, B_1\in V_N([B_1,l_1,B_1\overset{*}{\Rightarrow}w_1]\in\mathrm{DT})$ が存在するとき，同じ行の第2列目に (A_1,l_1) を書く．そして同じ行の第 i 列目に (A_i,l_i) があって $(A_{i+1}\to A_iB_{i+1})\in P\cup P'$ または $(A_{i+1}\to B_{i+1}A_i)\in P\cup P'$ で $A_{i+1}\neq A, A_1, A_2,\cdots A_i$ なる A_{i+1}, $B_{i+1}\in V_N([B_{i+1},l_{i+1},B_{i+1}\overset{*}{\underset{G}{\Rightarrow}}w_{i+1}]\in\mathrm{DT})$ が存在するとき，第 $i+1$ 列目に (A_{i+1},l_{i+1}) を書く．これを，同じ行に同じ非終端記号が現われない限り続ける（有限回の操作で終る）．そして行の非終端記号の並びが異なるときは異なる行を構成するとして（第1列目が同じでも異なる行を構成することが可能），すべての $V_N\cup\{I\}$ 要素について上の手続きを行う（DT 1 表の作成，この作成は有限回で終る）．ある行が，$A,(A_1,l_1),(A_2,l_2),\cdots,(A_i,l_i)$ のとき，以後 $[A,(A_1,l_1),\cdots(A_i,l_i)]\in\mathrm{DT}1$ と書くことにする．(a-3-2) $[A,e]\in\boldsymbol{A}_{ik}$, $[B,f]\in\boldsymbol{B}_{kj}$, かつ $(C\to AB)\in P\cup P'$, $[C,(C_1,l_1),\cdots,(C_m,l_m)]\in\mathrm{DT}1$ のとき，$[C,e+f]$, $[C_1,e+f+l_1]$, $\cdots$, $[C_m,e+f+l_1+\cdots+l_m]\in\boldsymbol{A}_{ik}\otimes\boldsymbol{B}_{kj}$ とする．

（b） $\dot{\vee}_\gamma$ は，2字組の要素で，2字組の2番目の要素が γ を越えないで最小になるものだけを集める集合演算で，要素間の次の演算を基にする．

$$\begin{aligned}&[A,e]\dot{\vee}_\gamma[B,f]\\&\quad=[A,e]\cup[B,f](A\neq B, e,f\leqq\gamma \text{ の時}),\\&\quad\ [A,e](e\leqq\gamma<f \text{ の時}),\\&\quad\ [B,f](f\leqq\gamma<e \text{ の時}),\ \phi(e,f>\gamma \text{ の時}).\\&[A,e]\dot{\vee}_\gamma[A,f]\\&\quad=[A,e](e\leqq\gamma<f,\ e\leqq f\leqq\gamma \text{ の時}),\\&\quad\ [A,f](f\leqq\gamma<e,\ f\leqq e\leqq\gamma \text{ の時}),\\&\quad\ \phi(e,f>\gamma \text{ の時}).\end{aligned}$$

$$\phi\dot{\vee}_\gamma[A,e]=[A,e]\dot{\vee}_\gamma\phi=\begin{cases}[A,e] & e\leqq\gamma \text{ の時}\\ \phi & e>\gamma \text{ の時}\end{cases}$$

（$\dot{\vee}_\gamma$ 定義は誤り数が γ を越えるものは除くという点とアルゴリズム1の(3)(i)(c)，(ii)(c)の手続きを行うことに対応している．）

（ii） 2字組を要素とする行列 $\boldsymbol{C}$ の $\boxtimes_\gamma$ 演算の下での推移的閉包 ${}^\gamma\boldsymbol{C}^+$ とは，

$${}^\gamma\boldsymbol{C}^+={}^\gamma\boldsymbol{C}^{(1)}\dot{\vee}_\gamma{}^\gamma\boldsymbol{C}^{(2)}\dot{\vee}_\gamma\cdots$$

である．ここに，${}^\gamma\boldsymbol{C}^{(i)}=\overset{i-1}{\underset{j=1}{\dot{\vee}_\gamma}}({}^\gamma\boldsymbol{C}^{(j)}\boxtimes_\gamma{}^\gamma\boldsymbol{C}^{(i-j)}$, ${}^\gamma\boldsymbol{C}^{(1)}=\phi\dot{\vee}_\gamma\boldsymbol{C}$ とする．

（iii） （a）入力系列 $w=a_1a_2\cdots a_n$ に対し，次の $(n+1)\times(n+1)$ 行列 $\boldsymbol{R}$ を定義する．(a-1) $\boldsymbol{R}_{i,i+1}\ni[A,\delta]$．ここに $(A\to a_i)\in P$ のとき $\delta=0$．$(A\to a_i)\in P'$ のとき．$\delta=1$．

(a-2) $B\to a\in P$, $a\neq a_i$ なるすべての $B\in V_N$ に対し，$[B,1]\in\boldsymbol{R}_{i,i+1}$

(a-3) $[A,\delta]\in\boldsymbol{R}_{i,i+1}$ で，$(A,(A_1,l_1),\cdots,(A_k,l_k))\in\mathrm{DT}1$ のとき $[A_1,\delta+l_1],\cdots,[A_k,\delta+l_1+\cdots+l_k]\in\boldsymbol{R}_{i,i+1}$．(a-4) $\boldsymbol{R}_{i,i+1}$ は(a-1)(a-2)(a-3)に限る．ただしここでは'$\in$'は $\phi\dot{\vee}_\gamma$ の意味で構成する．(a-5) $\boldsymbol{R}_{ij}=\phi$, $j\neq i+1$.

（明らかに $\boldsymbol{R}_{i,i+1}=t_{i1}$ である；t_{i1} はアルゴリズム1の(3)で定義したものである．）

（b） $\boxtimes_\gamma$ の下での推移的閉包 ${}^\gamma\boldsymbol{R}^+$ を求める．${}^\gamma\boldsymbol{R}^+$ を求める手順は Valiant[9] の手法を適用して次のようになる．

(b-1) $\boldsymbol{R}$ に対して，$2^{m-1}<n+1\leqq2^m$ なる m を求めて，$\boldsymbol{R}_{ij}'=\boldsymbol{R}_{ij}(1\leqq i,j\leqq n+1)$, $\boldsymbol{R}_{ij}'=\phi(n+2\leqq i\leqq2^m$ または $n+2\leqq j\leqq2^m)$ なる $\boldsymbol{R}'$ を構成する（明らかに ${}^\gamma\boldsymbol{R}_{ij}^+={}^\gamma\boldsymbol{R}_{ij}'^+$, $1\leqq i,j\leqq n+1$ である）．

(b-2) ${}^\gamma\boldsymbol{R}_{ij}'^+[1\leqq i,\ j\leqq2^m-2^m/k]$, ${}^\gamma\boldsymbol{R}_{ij}'^+[2^m/k<i,j\leqq2^m]$ が既知のとき，${}^\gamma\boldsymbol{R}_{ij}'^+[1\leqq i\leqq2^m/k,\ 2^m-2^m/k<j\leqq2^m]$ を求める手順を $OP_k(2^m)$ と書き，$OP_2(2^m)$, $OP_3(2^m)$, $OP_4(2^m)$ を帰納的に次のように定義する．$OP_2(2^m)$：㋑ ${}^\gamma\boldsymbol{R}_{ij}'^+[2^m/4<i,\ j\leqq3/4\times2^m]$ に $OP_2(2^{m-1})$ を適用し，㋺ $OP_3(3/4\times2^m)$ を ${}^\gamma\boldsymbol{R}_{ij}'^+[1\leqq i,\ j<3/4\times2^m]$ と ${}^\gamma\boldsymbol{R}_{ij}'^+[1/4\times2^m<i,j\leqq2^m]$ に適用し，㋩ $OP_4(2^m)$を適用する．㋭ $OP_3(3/4\times2^m)$を ${}^\gamma\boldsymbol{R}_{ij}'^+[1\leqq i,j\leqq3/4\times2^m]$ に適用することは（ホ-1）${}^\gamma\boldsymbol{R}_{ik}'^+[1\leqq i\leqq1/4\times2^m,\ 1/4\times2^m<k\leqq1/2\times2^m]\boxtimes_\gamma{}^\gamma\boldsymbol{R}_{kj}'^+[1/4\times2^m<k\leqq1/2\times2^m,\ 1/2\times2^m<j\leqq3/4\times2^m]$ を行い，（ホ-2）${}^\gamma\boldsymbol{R}_{ij}'^+[1\leqq i,\ j\leqq1/4\times2^m]$ と ${}^\gamma\boldsymbol{R}_{ij}'^+[1/2\times2^m<i,\ j\leqq3/4\times2^m]$ とに $OP(1/2\times2^m)$ を適用し，（ホ-3）$1/4\times2^m$ 回の $\dot{\vee}_\gamma$ 演算（（ホ-1），（ホ-2)の結果に適用）を行うことである，㋬ $OP_3(3/4\times2^m)$ を ${}^\gamma\boldsymbol{R}_{ij}'^+[1/4\times2^m<i,\ j\leqq2^m]$ に適用することは（ヘ-1）${}^\gamma\boldsymbol{R}_{ik}'^+[1/4\times2^m<i\leqq1/2\times2^m,\ 1/2\times2^m<k\leqq3/4\times2^m]\boxtimes_\gamma{}^\gamma\boldsymbol{R}_{kj}'^+[1/2\times2^m<k\leqq3/4\times2^m,\ 3/4\times2^m<j\leqq2^m]$ を行い，（ヘ-2）${}^\gamma\boldsymbol{R}_{ij}'^+[1/4

$\times 2^m<i, j\leqq 1/2\times 2^m]$ と $\boldsymbol{R}_{kj}'^{+}[3/4\times 2^m<i, j\leqq 2^m]$ とに $OP_2(1/2\times 2^m)$ を適用し，(ヘ-3) $1/4\times 2^m$ 回の $\vee_\gamma$ 演算（(ヘ-1)，(ヘ-2)の結果に適用)を行うことである．

㋣ $OP_4(2^m)$ を ${}^{\gamma}\boldsymbol{R}_{ij}'^{+}[1\leqq i, j\leqq 3/4\times 2^m]$，${}^{\gamma}\boldsymbol{R}_{ij}'^{+}[1/4\times 2^m<i, j\leqq 2^m]$ に適用することは (ト-1) ${}^{\gamma}\boldsymbol{R}_{ij}'^{+}[1\leqq i\leqq 1/4\times 2^m,\ 1/4\times 2^m<k\leqq 1/2\times 2^m]\boxtimes_\gamma {}^{\gamma}\boldsymbol{R}_{kj}'^{+}[1/4\times 2^m<k\leqq 1/2\times 2^m,\ 3/4\times 2^m<j\leqq 2^m]$ を行い，(ト-2) ${}^{\gamma}\boldsymbol{R}_{ik}'^{+}[1\leqq i\leqq 1/4\times 2^m,\ 1/2\times 2^m<k\leqq 3/4\times 2^m]\boxtimes_\gamma {}^{\gamma}\boldsymbol{R}_{kj}'^{+}[1/2\times 2^m<k\leqq 3/4\times 2^m,\ 3/4\times 2^m<j\leqq 2^m]$ を行い，(ト-3) (ト-1)，(ト-2)の結果に $\vee_\gamma$ を適用し，(ト-4) ${}^{\gamma}\boldsymbol{R}_{ij}'^{+}[1\leqq i, j\leqq 1/4\times 2^m]$，${}^{\gamma}\boldsymbol{R}_{ij}'^{+}[3/4\times 2^m<i, j\leqq 2^m]$ に $OP(1/2\times 2^m)$ を適用し，(ト-5) $1/4\times 2^m$ 回の $\vee_\gamma$ 演算（(ト-3)，(ト-4)の結果に適用）を行うことである．

（4） ${}^{\gamma}\boldsymbol{R}_{1n+1}{}^{+}=\phi$ のときは，'No' をかえす．${}^{\gamma}\boldsymbol{R}_{1n+1}{}^{+}\neq\phi$ のときは，${}^{\gamma}\boldsymbol{R}_{ij}{}^{+}=t_{i,j-i}(1\leqq i\leqq n,\ i+1\leqq j\leqq n+1)$ として，アルゴリズム1の（5）の手続きを行う．ここで $g(i, j, -)$ の定義においては，'$\in u_{ij}$' は '$\in t_{ij}$' とおきかえる．

（5） アルゴリズム1の（6）と同じ手続きを行い，訂正系列が w'（または ε）となるとき，$D(w', w)\leqq\gamma$（または（w の長さ)$\leqq\gamma$）であれば，w'（または ε）を答とし，そうでなければ 'No' をかえす．

［例 2］ 例1の cfg G, $w=aabab$, $\gamma\geqq 4$ に対しては，${}^{\gamma}\boldsymbol{R}_{ij}{}^{+}=t_{i,j-i}(1\leqq i\leqq n,\ i+1\leqq j\leqq n+1)$ となる．${}^{\gamma}\boldsymbol{R}^{+}$ の計算は次のように行う．

$$
{}^{\gamma}\boldsymbol{R}'^{+}=\begin{bmatrix}
\phi & {}^{\gamma}\boldsymbol{R}_{12}{}^{+} & ? & ? & ? & ? & \phi & \phi \\
\phi & \phi & {}^{\gamma}\boldsymbol{R}_{23}{}^{+} & ? & ? & ? & \phi & \phi \\
\phi & \phi & \phi & {}^{\gamma}\boldsymbol{R}_{34}{}^{+} & ? & ? & \phi & \phi \\
\phi & \phi & \phi & \phi & {}^{\gamma}\boldsymbol{R}_{45}{}^{+} & ? & \phi & \phi \\
\phi & \phi & \phi & \phi & \phi & {}^{\gamma}\boldsymbol{R}_{56}{}^{+} & \phi & \phi \\
\phi & \phi & \phi & \phi & \phi & \phi & \phi & \phi \\
\phi & \phi & \phi & \phi & \phi & \phi & \phi & \phi \\
\phi & \phi & \phi & \phi & \phi & \phi & \phi & \phi
\end{bmatrix}
$$

とし，

$$
\begin{bmatrix}
\phi & {}^{\gamma}\boldsymbol{R}_{12}{}^{+} & ? & ? \\
\phi & \phi & {}^{\gamma}\boldsymbol{R}_{23}{}^{+} & ? \\
\phi & \phi & \phi & {}^{\gamma}\boldsymbol{R}_{34}{}^{+} \\
\phi & \phi & \phi & \phi
\end{bmatrix},
\begin{bmatrix}
\phi & {}^{\gamma}\boldsymbol{R}_{34}{}^{+} & ? & ? \\
\phi & \phi & {}^{\gamma}\boldsymbol{R}_{45}{}^{+} & ? \\
\phi & \phi & \phi & {}^{\gamma}\boldsymbol{R}_{56}{}^{+} \\
\phi & \phi & \phi & \phi
\end{bmatrix}
$$

に $OP_2(1/2\times 8)$ を適用して，

$$
\begin{aligned}
&{}^{\gamma}\boldsymbol{R}_{12}{}^{+}\boxtimes_\gamma {}^{\gamma}\boldsymbol{R}_{23}{}^{+}=\boldsymbol{R}_{12}\boxtimes_\gamma \boldsymbol{R}_{23}={}^{\gamma}\boldsymbol{R}_{13}{}^{+},\\
&{}^{\gamma}\boldsymbol{R}_{23}{}^{+}\boxtimes_\gamma {}^{\gamma}\boldsymbol{R}_{34}{}^{+}=\boldsymbol{R}_{23}\boxtimes_\gamma \boldsymbol{R}_{34}={}^{\gamma}\boldsymbol{R}_{24}{}^{+},\\
&{}^{\gamma}\boldsymbol{R}_{34}{}^{+}\boxtimes_\gamma {}^{\gamma}\boldsymbol{R}_{45}{}^{+}=\boldsymbol{R}_{34}\boxtimes_\gamma \boldsymbol{R}_{45}={}^{\gamma}\boldsymbol{R}_{35}{}^{+},\\
&{}^{\gamma}\boldsymbol{R}_{45}{}^{+}\boxtimes_\gamma {}^{\gamma}\boldsymbol{R}_{56}{}^{+}=\boldsymbol{R}_{45}\boxtimes_\gamma \boldsymbol{R}_{56}={}^{\gamma}\boldsymbol{R}_{46}{}^{+},\\
&({}^{\gamma}\boldsymbol{R}_{12}{}^{+}\boxtimes_\gamma {}^{\gamma}\boldsymbol{R}_{24}{}^{+})\vee_\gamma({}^{\gamma}\boldsymbol{R}_{13}{}^{+}\boxtimes_\gamma {}^{\gamma}\boldsymbol{R}_{24}{}^{+})={}^{\gamma}\boldsymbol{R}_{14}{}^{+},\\
&({}^{\gamma}\boldsymbol{R}_{34}{}^{+}\boxtimes_\gamma {}^{\gamma}\boldsymbol{R}_{46}{}^{+})\vee_\gamma({}^{\gamma}\boldsymbol{R}_{35}{}^{+}\boxtimes_\gamma {}^{\gamma}\boldsymbol{R}_{56}{}^{+})={}^{\gamma}\boldsymbol{R}_{36}{}^{+}
\end{aligned}
$$

を得る．次に，

$$
\begin{bmatrix}
\phi & {}^{\gamma}\boldsymbol{R}_{12}{}^{+} & {}^{\gamma}\boldsymbol{R}_{13}{}^{+} & {}^{\gamma}\boldsymbol{R}_{14}{}^{+} & ? & ? \\
\phi & \phi & {}^{\gamma}\boldsymbol{R}_{23}{}^{+} & {}^{\gamma}\boldsymbol{R}_{24}{}^{+} & ? & ? \\
\phi & \phi & \phi & {}^{\gamma}\boldsymbol{R}_{34}{}^{+} & {}^{\gamma}\boldsymbol{R}_{35}{}^{+} & {}^{\gamma}\boldsymbol{R}_{36}{}^{+} \\
\phi & \phi & \phi & \phi & {}^{\gamma}\boldsymbol{R}_{45}{}^{+} & {}^{\gamma}\boldsymbol{R}_{46}{}^{+} \\
\phi & \phi & \phi & \phi & \phi & {}^{\gamma}\boldsymbol{R}_{56}{}^{+} \\
\phi & \phi & \phi & \phi & \phi & \phi
\end{bmatrix}
$$

に $OP_3(3/4\times 8)$ を適用し，

$$
\begin{bmatrix}
{}^{\gamma}\boldsymbol{R}_{13}{}^{+} & {}^{\gamma}\boldsymbol{R}_{14}{}^{+} \\
{}^{\gamma}\boldsymbol{R}_{23}{}^{+} & {}^{\gamma}\boldsymbol{R}_{24}{}^{+}
\end{bmatrix}\boxtimes_\gamma
\begin{bmatrix}
{}^{\gamma}\boldsymbol{R}_{35}{}^{+} & {}^{\gamma}\boldsymbol{R}_{36}{}^{+} \\
{}^{\gamma}\boldsymbol{R}_{45}{}^{+} & {}^{\gamma}\boldsymbol{R}_{46}{}^{+}
\end{bmatrix}\triangleq
\begin{bmatrix}
\boldsymbol{R}_{15}{}^{(1)} & \boldsymbol{R}_{16}{}^{(1)} \\
\boldsymbol{R}_{25}{}^{(1)} & \boldsymbol{R}_{26}{}^{(1)}
\end{bmatrix}
$$

を得，

$$
\begin{bmatrix}
\phi & {}^{\gamma}\boldsymbol{R}_{12}{}^{+} & ? & ? \\
\phi & \phi & \boldsymbol{R}_{25}{}^{(1)} & ? \\
\phi & \phi & \phi & {}^{\gamma}\boldsymbol{R}_{56}{}^{+} \\
\phi & \phi & \phi & \phi
\end{bmatrix}
$$

に $OP_2(1/2\times 8)$ を適用し，

$$
\begin{aligned}
&{}^{\gamma}\boldsymbol{R}_{12}{}^{+}\boxtimes_\gamma \boldsymbol{R}_{25}{}^{(1)}\triangleq \boldsymbol{R}_{15}{}^{(2)},\quad \boldsymbol{R}_{25}{}^{(1)}\boxtimes_\gamma {}^{\gamma}\boldsymbol{R}_{56}{}^{+}\triangleq \boldsymbol{R}_{26}{}^{(2)},\\
&({}^{\gamma}\boldsymbol{R}_{12}{}^{+}\boxtimes_\gamma \boldsymbol{R}_{26}{}^{(2)}\vee_\gamma(\boldsymbol{R}_{15}{}^{(2)}\boxtimes_\gamma {}^{\gamma}\boldsymbol{R}_{56}{}^{+})\triangleq \boldsymbol{R}_{16}{}^{(2)} \text{ とし，}\\
&\boldsymbol{R}_{15}{}^{(1)}\vee_\gamma \boldsymbol{R}_{15}{}^{(2)}={}^{\gamma}\boldsymbol{R}_{15}{}^{+},\quad \boldsymbol{R}_{26}{}^{(1)}\vee_\gamma \boldsymbol{R}_{26}{}^{(2)}={}^{\gamma}\boldsymbol{R}_{26}{}^{+},\\
&\boldsymbol{R}_{16}{}^{(1)}\vee_\gamma \boldsymbol{R}_{16}{}^{(2)}={}^{\gamma}\boldsymbol{R}_{16}{}^{+}
\end{aligned}
$$

を得る．$\boldsymbol{R}'_{ij}(i\leqq 6$ または $j\leqq 7)=\phi$ だから $OP_4{}^{(8)}$ の適用は不要である．

3.2.2 アルゴリズム2の正当性とその評価

［定理 5］ アルゴリズム2によって，誤り数が一定値 γ 以下の最小誤り訂正が仮定できるとき，最小誤り訂正が可能である．

［証明］ ${}^{-}\boldsymbol{R}^{+}$ の定義と，アルゴリズム1における t_{ij} の構成を対応づけると，${}^{-}\boldsymbol{R}_{i,j-i}=t_{ij}(1\leqq i\leqq n, 1\leqq j\leqq n-i+1)$ が主張できて，${}^{\gamma}\boldsymbol{R}^{+}$ が，2字組の誤り数が γ を越えないように制限したときの $\boldsymbol{R}$ の推移的閉包であることからアルゴリズム2の（3）までで，入力系列に対して，誤り数 γ を越えない範囲での構文解析が完了するということが主張できる．手続き（4）において，'No' をかえさずに，1つの最小誤り訂正を行うときは，構文行列 $[u_{ij}]$ を作成することを省いてある点を除いてアルゴリズム1の場合と同じである．ところで $g(i, j, [A, e])$ のルーチンに入るときは，$[A, e]$ が $t_{1n}=u_{1n}$ の要素と構文的に結合されているので，実際上 $[u_{ij}]$ 作成は省略して良い．以上から定理の妥当性が言える．　(証明終)

［定理 6］ アルゴリズム2は入力系列の長さ n に対して $O(n^{2.81})$ の時間，$O(n^2)$ の記憶量で可能である．

［証明］ アルゴリズム2において必要な記憶量が $O(n^2)$ であることは明らかであり，アルゴリズム2の（4）の手続きは $O(n^2)$ の時間で可能だから，（3）の手続きが $O(n^{2.81})$ の時間で可能であることを示せば

よい．

${}^{\gamma}\boldsymbol{R}^{+}$，${}^{\gamma}\boldsymbol{R}'^{+}$ を求めるのに要する時間を $T_{\gamma}(n)$，$T_{\gamma}(2^m)$ とする，((3)の手続きは $T_{\gamma}(n)+O(n^2)$ 以下で可能である．)

$2^m\times 2^m$ 行列の $\boxtimes_{\gamma}$ 演算の時閉が $M_{\gamma}(2^m)$ で可能とし，$2^m\times 2^m$ 行列に $OP_k(2^m)(k=2,3,4)$ を適用したときに必要な時間を $T_k(2^m)$ とするとき，Valiant[9]の定理2の証明に対応づけて，$2^t\cdot M_{\gamma}(2^m)\leqq M_{\gamma}(2^{m+1})(t>2)$ が仮定できるとき，$T_{\gamma}(2^m)\leqq 2T_2(2^m)+O(2^{2m})$，$T_2(2^m)\leqq M_{\gamma}(2^m)\cdot\mathrm{const.}$ を得ることができる．よって $2^t\cdot M_{\gamma}(2^m)\leqq M_{\gamma}(2^{m+1})(t>2)$ ならば，$T_{\gamma}(n)\leqq M_{\gamma}(n)\cdot\mathrm{const.}+O(n^2)$ となる．

$n\times n$ のブール行列の乗算に必要な計算時間を $BM(n)$ とし，$M_{\gamma}(n)\leqq BM(n)\cdot\mathrm{const.}$ を以下に示す．

$V_N\cup\{I\}$ の要素に番号付けし，それらを $A_1, A_2,\cdots A_h$ とかく (h は有限)．$2h(\nu+1)$ のブール行列を次に定義する；$\boldsymbol{A}[l,\alpha]_{ik}=1\Leftrightarrow[A_l,\alpha]\in\boldsymbol{A}_{ik}$，$\boldsymbol{B}[l,\beta]_{kj}=1\Leftrightarrow[A_l,\beta]\in\boldsymbol{B}_{kj}(1\leqq l\leqq h,\ 0\leqq\alpha,\ \beta\leqq\gamma)$．$h^2(\gamma+1)^2$ のブール行列を次に定義する；$\boldsymbol{C}[l,m,\alpha+\beta]=\boldsymbol{A}[l,\alpha]\cdot\boldsymbol{B}[m,\beta](1\leqq l,m\leqq h,\ 0\leqq\alpha+\beta\leqq 2\gamma)$．このとき，$[A_p,\alpha+\beta]\in(\boldsymbol{A}\boxtimes_{\gamma}\boldsymbol{B})_{ij}\Leftrightarrow\alpha+\beta=\min_{l,m}\{\alpha'+\beta'|\boldsymbol{C}[l,m,\alpha'+\beta']_{ij}=1\}\leqq\gamma\ \&\ (A_p\to A_lA_m)\in P\cup P'$ だから，$M_{\gamma}(n)\leqq BM(n)\cdot\mathrm{const.}$ が結論できる．$BM(n)=O(n^{\log 7})$ が示されているので $M_{\gamma}(2^{m+1})\leqq 2^t\cdot M_{\gamma}(2^m)(t>2)$ が満足され，$T_{\gamma}(n)\leqq O(n^{2.81})$，よって(3)の手続きは $O(n^{2.81})$ で実行可能である．（証明終）

4. むすび

本稿では，言語の誤りとして記号間の置き換え，記号の挿入，記号の削除が記号間に独立に生ずる独立誤りを仮定したときの，文脈自由言語の最小誤り訂正を，bottom-up 的に行うアルゴリズムとして，誤りが任意個生じ得る一般的な場合の方法を示し，計算時間が $O(n^3)$，記憶量が $O(n^2)$ で可能であることを述べた．次に，誤り数が一定値以下に制限できる場合は，最小誤り訂正法として，$O(n^{2.81})$ の時間で可能なアルゴリズムを示した．この点から，誤り数が制限できるときには，bottom-up 的手法が有効である．誤り数に制限がない場合は，許容誤り数が入力系列の長さに近づき，あるいはそれを越えうるので，誤り数の解析を伴う bottom-up 的構文解析を行列演算に帰着させることは得策でないと思われる．

謝辞　御援助頂く本学工学部堂下教授，矢島教授，日頃有益な御助言を賜わる本学工学部上林助教授，議論頂いた研究室諸氏に深謝する．

参考文献

1) 岩元，沢野：正則言語，文脈自由語の誤り訂正，電子通信学会論文誌，Vol. 56-D, No. 12, pp. 675～680 (1973)
2) J. Earley: An efficient context-free parsing algorithm, Comm. ACM, Vol. 13, No. 2, pp. 94～102 (1970)
3) A. V. Aho & T. G. Peterson: A minimum distance error-correcting parser for context-free languages, SIAM J. on Computing, Vol. 1, No. 4, pp. 305～312 (1972)
4) G. Lyon: Syntax-directed least errors analysis for context-free languages, Comm. ACM, Vol. 17, No. 1, pp. 3～14 (1974)
5) D. H. Younger: Recognition and parsing of context-free language in time n^3, Information and Control, Vol. 10, pp. 189～208 (1967)
6) L. W. Fung & K. S. Fu: Maximum-likelihood syntactic decoding, IEEE Trans. on Information Theory, Vol. IT-21, pp. 423～430 (1975)
7) J. E. Hopcroft & J. D. Ullman: Formal Languages and Their Relation to Automata, Addison-Wesley Publishing Company, (1969)
8) S. Ginsburg: The Mathematical Theory of Context-Free Languages, McGraw-Hill, (1966)
9) L. G. Valiant: General context-free recognition in less than cubic time, Journal of Computer and System Sciences, Vol. 10, pp. 308～315 (1975)
10) A. V. Aho & J. D. Ullman: The Theory of Parsing, Translation, and Compiling (Vol. 1. Parsing), Prentice-Hall, Inc., (1972)

(昭和51年9月14日受付)
(昭和51年12月15日再受付)