社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.
IT2001-46(2001-11)

形式言語と圧縮に関する一考察

中澤　真[†]　松嶋　敏泰[††]　平澤　茂一[††]

† 早稲田大学メディアネットワークセンター
〒169-8050 東京都新宿区戸塚町 1-104
†† 早稲田大学理工学部経営システム工学科
〒169-8555 新宿区大久保 3-4-1

E-mail: †nakazawa@mn.waseda.ac jp

あらまし　近年，SEQUITUR や MPM などの文法に基づく符号化法が盛んに研究されている．本研究では，Chomsky の階層における狭い文法のクラスがユニバーサル情報源符号化において果たす役割について明らかすることを目的とする．またこれに関連して，正則文法と決定性文脈自由文法についての個々の系列に対する圧縮限界として Kolmogorov complexity を示し，形式文法の階層においてどのような性質を持つのか明らかにする．

キーワード　正則文法，決定性文脈自由文法，無歪み圧縮，Kolmogorov コンプレクシティ

A Note on Formal Languages and Data Compression

Makoto NAKAZAWA[†], Toshiyasu MATSUSHIMA[††], and Shigeichi HIRASAWA[††]

† Media Network Center, Waseda University
Totsuka-cho 1–104, Shinjuku-ku, Tokyo, 169-8050 Japan
†† School of Science and Engineering, Waseda University
Ohkubo 3–4–1, Shinjuku-ku, Tokyo, 169–8555 Japan

E-mail: †nakazawa@mn.waseda.ac.jp

Abstract　Recently, grammar based codes are researched. In this paper, it aims to clarify the role that the hierarchy of the formal grammar plays in the universal source coding. In relation to this, it does not know what property Kolmogorov complexity has in the hierarchy of the grammar but we will clarifies it.

Key words　Formal grammars, Lossless compression, Kolmogorov complexity

1. まえがき

近年の急激なコンピュータテクノロジーの進歩により，CPU パワー，記憶装置の容量などは毎年飛躍的に向上している．また通信環境の整備が進み伝送速度も向上している．これらの進歩により，あらゆる情報が電子化され，情報量も増加の一途をたどっている．このため情報を効率よく蓄積・伝送するということは依然として重要な要求であり，データ圧縮技術の必要性は高まるばかりである．

情報源符号化の研究の中でも特に無歪み圧縮において，1990 年代後半から文法を用いた圧縮法の研究が行われてきた．この文法に基づく符号には SEQUITUR [7] [8] や MPM [2] [3] などいくつかの研究があるが，いずれも情報源系列を言語として受理する文法を構成し，この文法を算術符号などによりバイナリ表現することにより無歪み圧縮を行う手法である．復号側は受け取った文法の生成規則に従って情報源系列を再生成することができる．

ここで用いられている文法は形式言語を受理するような文法で，計算量理論の分野で古くから研究されてきたものである．この形式文法のクラスは Chomsky の階層 [13] とよばれる階層構造を持っている．0 型文法から 3 型文法までの 4 つの基本文法について階層が示され，またこれ以外の文法についても，包含関係や性質についての証明が試みられている．先の SEQUITUR や MPM で用いられている文法はいずれも決定性文脈自由文法である．しかし，これら以外の文法のクラスがデータ圧縮においてどのような役割を果たすのか明らかではない．

文法と圧縮の関係についての研究として Kolmogorov complexity [10] がある．これは入力系列を出力するユニバーサルチューリングマシン上の最小のプログラム長という形で系列の複雑さを定義している．Chomsky の階層において最も広いクラスである句構造文法はチューリングマシンと等価であることが示されているので，この文法については Kolmogorov complexity の議論がそのまま用いることが可能である．しかし，計算機モデルとしてはチューリングマシンは汎用性が高すぎるため，実際のアルゴリズムとの乖離が大きいと考えられる．一般に 3 型（正則）文法や 2 型（文脈自由）文法はそれぞれ有限状態オートマトンやプッシュダウンオートマトンなどの計算機モデルとの等価性が示されており，これらはいずれもチューリングマシンが受理する言語のクラスに真に含まれることが証明されている．

本研究では，このような Chomsky の階層における狭い文法のクラスが文法に基づく符号においてどのような振る舞いをするか考察することが目的である．特に，正則文法と決定性文脈自由文法について，これらのクラスと Kolmogorov complexity との関係を明らかにする．さらにそれぞれの文法によって模倣することが可能な圧縮アルゴリズムを示し，どのような構造を持つ情報源系列に対して文法が適切な記述をするか示す．

2. 準備

まず初めに形式文法のための基本的な準備をする．有限かつ非空なアルファベットを Σ，Σ 上のすべての語(注1)の集合を Σ^* で表す．この Σ 上の語の順序対の有限集合を P とする．P の要素 (μ,ω)，$\mu,\omega \in \Sigma^*$ を生成規則と呼び，$\mu \to \omega$ と表す．このとき $RW = (\Sigma, P)$ を書き換えシステムという．このとき文法 G を以下のように定義する．

［定義 1］ Σ_N を非終端アルファベット，Σ_T を終端アルファベットとし，$\Sigma = \Sigma_N \cup \Sigma_T$，$\Sigma_N \cap \Sigma_T = \phi$ を満足する集合とする．このとき文法 G を $G = (RW, \Sigma_N, \Sigma_T, S)$ と定義する．ただし，S は開始記号で $S \in \Sigma_N$ である．

便宜上，非終端アルファベット Σ_N の要素を大文字のアルファベット $A, B, C, \ldots$ で，終端記号 Σ_T の要素を小文字のアルファベット $a, b, c, \ldots$ で表すこととする．$\alpha, \beta, \nu_1, \nu_2 \in \Sigma^*$ に対して，書き換えシステム RW の P の要素である生成規則 $\mu \to \omega$ について $\alpha = \nu_1\mu\nu_2$，$\beta = \nu_1\omega\nu_2$ が成り立つとき $\alpha \Rightarrow \beta$ と表し，RW に従って α から β が直接導出されるという．また，Σ 上の語の有限列 $\omega_1, \omega_2, \ldots, \omega_k$ $(k \geqq 0)$ について $\alpha = \omega_1 \Rightarrow \omega_2$，$\omega_2 \Rightarrow \omega_3, \ldots, \omega_{k-1} \Rightarrow \omega_k = \beta$ が成り立つとき $\alpha \Rightarrow^* \beta$ と表し，RW に従って α から β が導出されるという．

［定義 2］ 文法 G によって生成される言語 $L(G)$ を

$$L(G) \triangleq \{u \in \Sigma_T^* : S \Rightarrow^* u\} \tag{1}$$

と定義する．

文法 G に基づく符号は情報源系列 $x \in \Sigma^*$ に対して，

$$L(G) = \{x\} \tag{2}$$

なる文法を構成し，この文法を算術符号などによってバイナリ表現して符号化する方法である．復号器

(注1)：0 個以上の Σ 上の要素（記号）からなる有限列．

はこの文法から言語 $L(G)$ を生成することが可能なため，式 (2) より情報源系列 x を復号できることになる．ただし文法を符号として利用するためには一意復号性などいくつかの条件を満足する必要がある．このため文法の性質に関する定義をする．

［定義 3］ 以下の条件を満足するとき文法 G は許容可能（admissible）[2] であるという．

(i) G が決定性文法である．

(ii) G の任意の生成規則 $u \to w$ において $w \neq \lambda$ (注2)．

(iii) $L(G) \neq \phi$.

(iv) G は生成規則の中に一度も現れない記号を持たない．

3. 形式文法と Kolmogorov complexity

許容可能な文法を考えることにより，文法に基づく符号化法の必要条件が整った．しかし，許容可能性を満たす文法は無数に存在する．ここでは形式文法のクラスについて考え，このクラスが圧縮にどのような影響を及ぼすか考える．

形式文法においては Chomsky の階層 [13] が広く知られている．この階層の定義を以下に示す．

［定義 4］ 文法 G の生成規則集合 P の各要素が次の条件 $0, 1, \cdots, i$, を満たすとき，type i 文法，$i = 0, 1, 2, 3$ という．

(条件 0) $\mu \to \omega,\ \mu, \omega \in \Sigma^*$

(条件 1) $\mu \to \omega,\ |\mu| \leq |\omega|,\ \mu, \omega \in \Sigma^*$ (注3)

(条件 2) $A \to \omega,\ A \in \Sigma_N, \omega \in \Sigma^*$

(条件 3) $A \to \alpha B$ または $A \to \alpha$

ただし，$A, B \in \Sigma_N, \alpha \in \Sigma_T^*$

表 1 Chomsky による言語階層

生成文法	言語型	言語名
type 0	L_0	句構造言語
type 1	L_1	文脈依存言語
type 2	L_2	文脈自由言語
type 3	L_3	正則言語

正則文法は正則言語を受理する Chomsky の階層における最も制約が厳しい文法である．この正則文法に基づく符号を構成した場合，任意の情報源系列について符号化することが可能である

［補題 1］ 許容可能な正則文法のクラスを $\mathcal{G}_{RG}$ とする．このとき，任意の有限系列 $\forall x \in \Sigma^*$ に対し，

$$^{\exists}G \in \mathcal{G}_{RG},\ L(G) = \{x\}. \qquad (3)$$

［証明］ 一般性を失うことなく $\Sigma_T = \{0, 1\}$ と仮定してよい．ここでは情報源系列が $x = 01110001$ の場合を考える．この場合 $L(G) = \{x\}$ が成り立つように $S \to 01110001$ という生成規則ただ一つからなる文法 G が構成できる．明らかにこの文法は許容可能な正則文法である． □

どのような有限な情報源系列も許容可能な正則文法で生成可能であることが補題 1 から明らかになったが，正則文法という文法のクラスが圧縮という視点から考えた場合，適切であるのであろうか．補題 1 の証明のように正則文法を構成し，このまま 2 値符号化したのでは，まったく圧縮されないことは明らかである．また，Chomsky の階層から正則文法を含む表現能力のより高い文法が存在するが，それらを用いた符号化との差異はどのようになるか考えてみる．

そのための準備として個々の系列（individual sequence）の複雑さの尺度である Kolmogorov complexity を定義する．

［定義 5］ ψ を部分帰納的関数とする．このとき $x \in \Sigma_T^*$ の系列の複雑さ $\mathcal{C}$ を以下のように定義する．p をユニバーサルチューリングマシン上のプログラムとするとき

$$\mathcal{C}_\psi(x) \triangleq \min\{|p| : \psi(p) = x\}. \qquad (4)$$

ただし，上記の条件を満足する p が存在しない場合は $\mathcal{C}_\psi(x) = \infty$ とする．

また，ψ がユニバーサル(注4) な帰納的関数の場合 ψ_0 と表し，$\mathcal{C}_{\psi_0}(x)$ を Kolmogorov complexity とよび，$\mathcal{C}(x)$ で表す．

チューリングマシンと 0 型文法の等価性から p を句構造文法を表記したものとも考えることができる．この場合，ψ_0 はこの文法のインタープリタである．先に述べたようにより制限されたクラスにおいて同様の複雑度を定義する．決定性文脈自由文法（DCFG），正則文法（RG）を解釈する部分帰納的関数をそれぞれ ψ_{DCFG}，ψ_{RG} とすると式 (4) と同様に $\mathcal{C}_{DCFG}(x)$，$\mathcal{C}_{RG}(x)$ を定義することができる．このとき以下の定理が成り立つ．

(注2)：λ は空列を表す．

(注3)：$|\cdot|$ は記号列の長さを表す．

(注4)：任意の帰納的関数 $\psi : \{0,1\} \to \Sigma^*$ に対して，次の 2 つの条件を満足する $i \in \{0,1\}^*$ が存在するとき ψ_0 をユニバーサルな帰納的関数という．i) $\psi(p)$ が定義されるならば $\psi_0(i, p)$ も定義される．ii) ψ の定義域において $\forall p, \psi_0(i, p) = \psi(p)$.

［定理 1］ 任意の $x \in \Sigma_T^*$ に対して，

$$\mathcal{C}_{DCFG}(x) \leq \mathcal{C}_{RG}(x) + \tau. \qquad (5)$$

ただし，τ は DCFG と RG にのみ依存して決定される定数である．

［証明］ DCFG が RG を真に包含することが証明されている [13]．これは任意の RG を DCFG が模倣可能であることを示している．DCFG のクラスに含まれる文法を枚挙した場合，RG を模倣する文法がいつか必ず現れ，これを特定するためのインデックスの複雑さが定数 τ に相当する．明らかに τ は DCFG と RG にのみ依存し，x には依存しない． □

［定理 2］ $|G|$ を文法 G の生成規則の右辺の長さの総和とする．$x \in \Sigma_T^*$ に対し，$L(G) = \{x\}$ となるような許容可能な正則文法 $G \in \mathcal{G}_{RG}$ について次式が成り立つ．

$$|G| = \Omega(|x|) \ ^{(注5)}. \qquad (6)$$

［証明］ 正則文法は右線形（あるいは左線形）であり，許容可能性からこの文法 G は決定性文法であるため，生成規則の右辺で同じ非終端記号が 2 回以上現れることはない．このため，x の部分列で繰り返されるパターンを非終端記号に置き換えるという文法を構成できない．結果として生成規則の右辺には x のすべての部分列が現れることになる． □

定理 2 はスタック機能を持つ決定性プッシュダウンオートマトンと持たない有限状態オートマトンとの差異によるものと考えることができる．

4. 決定性文脈自由文法

決定性文脈自由文法は正則文法と文脈自由文法との間に位置し，真の包含関係をなすことが証明されている [13]．文法に基づく符号として研究された SEQUITUR や MPM の文法もこの決定性文脈自由文法である．ここではまず，辞書法として知られている LZ78 符号や LZW 符号を生成規則による表現に書き換えて，これが決定性文脈自由文法であることを示す．

4.1 LZ78 符号

ユニバーサル符号の代表的なものの一つに 1978 年に Lempel と Ziv により研究され実用化された LZ78 符号 [6] がある．この符号は増分分解によって情報源系列を分解し，分解された部分列を再帰的に参照することによって圧縮を行う符号化法である．この増分分解に関する部分を文法における生成規則として表現した場合の例を次に示す．

［例 1］ 終端アルファベット，非終端アルファベットをそれぞれ $\Sigma_T = \{0, 1\}, \Sigma_N = \{A, B, \ldots\}$ とする．このとき情報源系列が $x = 010101010101$ であった場合，増分分解された部分列ごとに生成規則を新たに作成する．

$$\begin{aligned} &A \to 0 \\ &B \to 1 \\ &C \to A1 \\ &D \to C0 \\ &E \to B0 \\ &F \to E1 \\ &S \to ABCDEF \end{aligned}$$

生成規則の右辺に増分分解された部分列がすべて再帰的に表現され，開始記号 S をもつ生成規則がこれらをつなぎ合わせる役割を果たしている．この生成規則から x が生成できるのは明らかである．

4.2 LZW 符号

LZ78 符号の改良版アルゴリズムの一つである LZW 符号 [14] についても，LZ78 と同様に生成規則によるアルゴリズムの表現が可能である．情報源系列についての分解方法の概念は基本的に同じである．

［例 2］ 例 1 と同様にアルファベットを $\Sigma_T = \{0, 1\}$, $\Sigma_N = \{A, B, \ldots\}$ とし，$x = 010101010101$ の場合を考える．LZW ではあらかじめ 1 文字からなる部分列をすべて辞書に前もって登録する．これを生成規則で表すと次のようになる．

$$\begin{aligned} &A \to 0 \\ &B \to 1 \end{aligned}$$

符号化対象の部分列を辞書内に登録されていない未一致系列になるまで伸長し，これを新たな部分列として辞書に登録する．上記の場合先頭の 01 が辞書に登録されていない未一致系列となるので登録することになる．これを一つの生成規則に対応させると以下のようになる．

$$\begin{aligned} &C \to A1 \\ &D \to B0 \\ &E \to C0 \\ &F \to E1 \\ &S \to ABCEDF \end{aligned}$$

(注5)：関数 f, g に対し，$\exists c, d > 0, \forall x, f(x) \leq cg(x) + d$ が満足するとき，$f = \Omega(g)$．

これらの結果から次の定理を導くことができる．

［定理 3］ LZ78 符号，LZW 符号は決定性プッシュダウンオートマトンにより模倣可能である．

LZ78 符号，LZW 符号は共に決定性文脈自由文法によってそのアルゴリズムを模倣することが可能であることを示したが，構成された文法をどのように符号化するかという点について，二つの符号には差が生じる．LZW 符号では生成規則の構成方法がその時点までに既に得られた部分列により一意に定まるため，生成規則を逐一符号化する必要はない．必要な情報は $S \rightarrow ABCEDF$ という開始記号を含む生成規則のみである．一方 LZ78 符号は生成規則の右辺についての情報(注6)をすべて符号化する必要がある．そのため構成された文法はどちらも非常によく似ているが，文法の符号化部分において効率性に差が生じることになる．

5. 正 則 文 法

正則文法は補題 1 で示したように，何の工夫もないまま文法に基づく符号に用いてしまうと意味をなさない．しかし，情報源の制約条件によっては効率的な符号を構成できる．

5.1 ブロックソート符号（BS 符号）

BS 符号 [4] は情報源系列からソートテーブルを作成し，類似文脈内における相対的な位置を示すことによって符号化する方法である．これを文法として考えた場合どのように記述されるか以下の例によって示す．

［例 3］ 終端アルファベット，非終端アルファベットをそれぞれ $\Sigma_T = \{a, b, c, \dots\}, \Sigma_N = \{A, B, C \dots\}$ とする．このとき情報源系列が $x = abacabd$ であった場合，以下のような生成規則が構成される．まず，x を $|x|$ 回巡回シフトし，作られた $|x|$ 個の系列を辞書式順序でソートする．ソートされた系列の第 1 ビットのみを取り出して構成した系列を μ，最終ビットのみを取り出して構成した系列を ω とする．この場合，$\mu = a_1a_2a_3b_1b_2cd$，$\omega = dcb_1a_1a_2a_3b_2$ (注7) となる．

μ と ω のそれぞれの系列の位置を示すために μ の第 i 文字を μ_i，同様に ω の第 i 文字を ω_i とする．ω_i と μ_i は情報源系列において隣接した記号対である．これらの記号対から $\omega_i \rightarrow \omega_i\mu_i$ という生成規則を作る．ただし，左辺と右辺の第 2 文字目を対応がつくような非終端記号に置き換える．この場合以下のような文法が構成される．

$$D \rightarrow dA_1$$
$$C \rightarrow cA_2$$
$$B_1 \rightarrow bA_3$$
$$A_1 \rightarrow aB_1$$
$$A_2 \rightarrow aB_2$$
$$A_3 \rightarrow aB_C$$
$$B_2 \rightarrow bD$$

ソートテーブルにおける x の行番号の情報から開始記号 S の位置を決めて，上記の生成規則の一部を変更する．

$$D \rightarrow d\lambda$$
$$S \rightarrow aB_1$$

この文法は明らかに x を生成する．

［定理 4］ BS 符号は許容可能な正則文法に基づく符号化法によって模倣可能である．

［証明］BS 符号は常に情報源系列を記号対からなる部分列に分解し，この文脈をソートすることにより圧縮を行っている．そのため生成規則の右辺は必ず 2 文字以下のアルファベットで構成される．またこのとき，右線形となっていることからこの文法が正則文法であることが示される． □

BS 符号を上記のような方法で文法を構成した場合，問題となるのは文法の符号化である．この符号化において送るべき情報はソートテーブルにおける x の行番号と ω のみである．この情報から一意に上記の文法が定まる．情報源によっては ω の系列で連続するアルファベットの個数が多くなり Recency Rank(RR) 符号などを用いて効率的に符号化できる．

逆に，BS 符号において圧縮が不利になる系列は隣接する 2 文字対がすべて異なるような系列である．例えば，$x = abcdbacadbc$ という系列を分解した場合 $ab, bc, cd, db, ba. ac, ca, ad, db, bc$ となり，ω は同じ文字が連続する可能性が非常に小さくなる．このような場合 RR 符号を用いて符号化することは効率的とは言い難い．

6. 文法の符号化とエントロピー

4 節で述べたように，同じ DCFG に属する文法でも文法の符号化方法において圧縮率に差が生じる．文法をどのように記述すべきか最適な決定をするのは一般的に困難であるが，Kolmogorov complexity の視点から考えた場合は漸近的にその差は無視でき

(注6)：開始記号 S で始まる生成規則を除く．
(注7)：終端記号についた添字は識別のためにつけたインデックスで，出現順による．

ることになる．その結果，定理1からも明らかなように情報源に制約がなく，計算量についても考慮しないのであれば，できる限り広いクラスに含まれる任意の文法を用いれば十分であるという結論が導かれる．

しかし，情報源に制約が加えられたり，計算量を考慮した場合は別である．5節で述べたBS符号の場合，文法Gの生成規則の集合Pの要素数は一般的に大きくなってしまい，この正則文法は非効率的であるが，有限状態情報源などを仮定した場合は漸近最良性が示されている．この事からも文法によって記述することができる情報源系列の構造と，文法自身の符号化，およびエントロピーについて考える必要がある．

7. むすび

文法に基づく符号は個々の系列の構造を抽出し，その文法においてその構造を容易に記述することができるような規則性を見つけだすことが鍵となる．文法に対応した規則性を見つけだすことに成功した場合，生成規則の長さ，個数は一般的にいずれも小さくなり，生成される生成規則にも偏りが生じることになり，文法の符号化も効率的に記述される．

本研究では，形式文法の階層とKolmogorov complexityの関係について明らかにし，文法に基づく符号化法における文法の構成について一つの示唆を与えた．また，正則文法や決定性文脈自由文法によって模倣することが可能な圧縮法について示し，文法が持つ表現能力によって情報源系列の構造の抽出に大きな差があることを示した．

しかし，文法を具体的にどのように記述するかという符号化の問題については深く踏み込むことができなかった．これを明らかにするためには，MPMや修正SEQUITURが有限状態情報源に漸近最良性を持つことが示されているように，情報源に対して制約条件を仮定する必要があるだろう．特に情報源についても計算機モデルと考え，このクラスと文法のクラスとを対比させることが問題解決の糸口になるであろう．

それぞれの文法における導出のための計算量，生成規則の適用ための計算量との関係も明らかにする必要がある．

謝辞　著者の一人中澤は本研究に際して貴重な御助言をいただきました横浜商科大学の浮田善文先生と早稲田大学の小林学氏に深く感謝いたします．また暖かく支援してくださった法政大学の西島利尚先生に深く感謝いたします．なお，本研究の一部は科学研究補助金（課題番号12875172）の助成による．

文　献

[1] J.C.Kieffer and E.Yang, "Sequential Codes, Lossless Compression of Individual Sequences, and Kolmogorov Complexity," IEEE Trans. Inform. Theory, vol.42, No.1, pp.29-39, January 1996.

[2] J.C.Kieffer and E.Yang, "Grammar-Based Codes: A New Class of Universal Lossless Source Codes," IEEE Trans. Inform. Theory, vol.46, No.3, pp.737-754, May 2000.

[3] J.C.Kieffer, E.Yang, G.Nelson, and P.Cosman, "Universal Lossless Compression via Multilevel pattern Matching," IEEE Trans. Inform. Theory, vol.46, No.4, pp.1227-1245, July 2000.

[4] M.Burrows and D.J.Wheeler, "A Block-Sorting Lossless Data Compression Algorithm," SRC Research Report 124, Digital Systems Research Center, May 1994.

[5] J.Ziv and A.Lempel, "A Universal Algorithm for Sequential Data Compression," IEEE Trans. Inform. Theory, vol.23, No.3, pp.337-343, May 1977.

[6] J.Ziv and A.Lempel, "Compression of Individual Sequences via Variable-Rate Coding," IEEE Trans. Inform. Theory, vol.24, No.5, pp.530-536, September 1978.

[7] C.G.Nevil-Manning, I.H.Witten, and D.L.Maulsby, "Compression by Induction Hierarchical Grammars," Proc. IEEE DCC'94, pp.244-253, Snowbird, Utah, USA, March 1994.

[8] C.G.Nevil-Manning and I.H.Witten, "Compression and Explanation Using Hierarchical Grammars," The Computer Journal, vol.40, No.2/3, pp.104-116,1997.

[9] C.G.Nevil-Manning and I.H.Witten, "Identifying Hierarchical Structure in Sequences: A Linear-Time Algorithm," J. of Artificial Intelligence Research, vol.7, pp.67-82, 1997.

[10] M.Li and P.M.B.Vitányi, "An Introduction to Kolmogorov Complexity and Its Application," Springer-Verlag, 1993.

[11] M.Li and P.M.B.Vitányi, "A New Approach to Formal Language Theory by Kolomogorov Complexity," SIAM J. Comput. vol.24, No.2, pp.398-410, 1995.

[12] J.E.Hopcroft and J.D. Ullman, "Introduction to Automata Theory, Languages and Computation I, " Addison Wesley College, 2000.

[13] J.E.Hopcroft and J.D. Ullman, "Introduction to Automata Theory, Languages and Computation II," Addison Wesley College, 2000.

[14] T.A.Welch, "A Tchnique for High-Performance Data Comprsesion," IEEE Computer, vol.17, No.6, pp.8-19, June 1984.