Vol. 25 No. 4 情報処理学会論文誌 July 1984

# 二値算術符号の符号化効率について†

森田啓義†† 藤元克己††
北田茂††† 有本卓†††

画像の符号化やデータ伝送など多くの応用に適したデータ圧縮符号として算術符号がよく知られている．算術符号の性能は，符号化・復号化アルゴリズムにおいて行われる算術演算の精度とシンボルの生起確率の近似法に大きく依存している．本論文では，二値算術符号の性能を測る測度として符号の効率（平均符号長に対するエントロピの比）を定義し，任意の演算精度と確率の近似法のもとで効率の評価式（上界および下界）を導いた．これらの評価式を用いた数値例から，高いデータ圧縮効果を保持したまま符号化処理時間を短縮できるような演算精度の設定および確率の近似法について詳しく検討した．

## 1. はじめに

従来，データ圧縮技法（情報源符号化法）は大規模データのコンピュータ間伝送やデータをコンパクトにファイル化する場合に広く利用されている．なかでも二値情報源の符号化法については，最近のファクシミリやコンピュータによる漢字処理の実用化に伴い，多くの研究がなされ，種々の方式が提案されている[1)~7)]．

応用の立場からすると，リアルタイムで符号化・復号化が行え，さらに，データの統計的性質の変動に即応しうる符号化法が強く望まれる．そのためには符号化・復号化時の処理時間がいずれもデータ長の線形オーダであることが必要とされる．この条件を満足する符号の一つのクラスとして算術符号がよく知られている[8)~12)]．算術符号は Elias による符号化法[3)]を一般化したものとみなせる．Elias は $[0,1)$ 区間をデータ・シンボルの生起確率の比に応じて順次分割してゆき，一つのシンボル列にある部分区間を対応させるという方法を提案した．この符号化法は符号構成が簡明であり，記憶を有するデータや確率分布の変動する場合にも適用が容易である．しかしながら残念なことに，Elias 符号は部分区間を細かく分割してゆくにつれ，より高い精度の乗算を必要とするため，全体として符号化・復号化に必要な計算手数はシンボル列の長さの2乗に比例する[5)]．この点は Pasco[8)] によって改良された．彼は浮動小数点法によって一定の有限桁で乗算を実行しても，Elias の符号化が可能なことを示した．一方，まったく別の観点から Rissanen[9)] も Pasco のと同等な符号を提案している．彼らの符号を称して算術符号と呼ぶ．

算術符号を実際に使用する場合，算術演算やシンボル確率といった符号化パラメータをいかに設定するかが符号の性能を決める上でもっとも重要である．すなわち，符号化パラメータをできるだけ精度よく表せば高い効率が得られるが符号化・復号化時の処理に多くの時間が費やされる．逆に符号化パラメータの精度を落とせば高速な処理が可能であるが効率は悪化する．そこで，効率と処理時間という相反する要求を同時に満足するように符号化パラメータを設定する必要がある．

本論文では二値算術符号の設計上の一つの指針を与えることを目的に，任意の精度で符号化パラメータを表した場合における効率の評価を行い，有用な上界・下界式を導いた．さらにそれらの評価式に基づき，あらかじめ設定した効率（90%，98%）を達成するという条件下で，その条件にもっとも適した符号化パラメータの設計を行う．

2章では算術符号の構成を示し，符号化・復号化アルゴリズムを与え，復号の一意性について論じる．さらに符号化アルゴリズムの実行時に生じる桁上りの伝播問題に関して詳しく述べる．3章では平均符号長の評価式を求め，さらに効率の上界・下界を導出する．4章では符号化パラメータの設計を行う．

なお本論文で対象とする情報源は定常無記憶二値情報源とし，シンボル0の生起する確率を $p$，シンボル1の生起する確率を $1-p$ とする．ただし，$0<p\leqq 1/2$ と仮定する．また本文中に現れる対数の底はすべ

† On Efficiency of Binary Arithmetic Codes by HIROYOSHI MORITA, KATSUMI FUJIMOTO (Department of Production System Engineering, Faculty of Engineering, Toyohashi University of Technology), SIGERU KITADA and SUGURU ARIMOTO (Department of Mechanical Engineering, Faculty of Engineering Science, Osaka University).

†† 豊橋技術科学大学工学部生産システム工学系
††† 大阪大学基礎工学部機械工学科

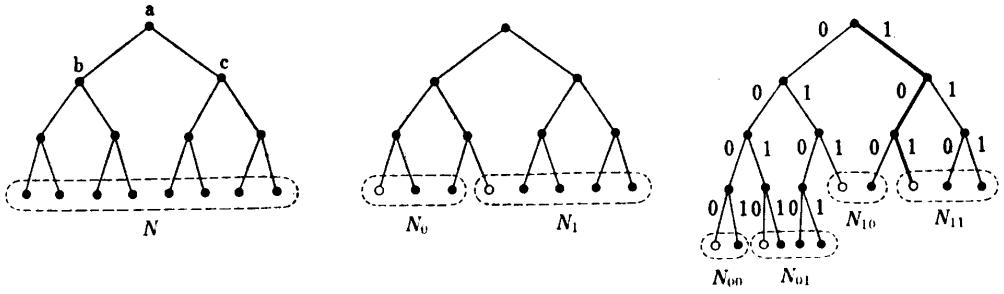


**図 1** 節点の集合 $N_s$ の構成法 ($p=0.4$, $w=3$ の場合)

Fig. 1 Construction of the set of nodes $N_s$.

て2とする.

## 2. 算術符号の構成

### 2.1 算術符号の原理

この節では算術符号における符号語の構成法を2進木を用いて説明する. まず一つの節点aが与えられているとする. 節点aから左の枝, 右の枝を出し, 両方の枝の先には新しく節点b, cを設ける. この操作を「節点aを分枝する」と呼ぼう. また節点b, cを葉と呼ぶ. 新しくできたすべての葉から分枝を次々に繰り返し, $2^w$ ($w$ は自然数) 個の葉をもつ木を作る. **図1(a)**に $w=3$ の場合を示す. 葉である節点全体の集合を $N$ とおく. 次に図1(b)に示すように, $N$ を左右二つの部分集合 $N_0$ と $N_1$ に分割する. このとき, $|N_0|$ と $|N_1|$ の比がシンボル確率の比 $p:1-p$ にできるだけ近くなるようにする. 確率の比によく合った分割が得られるかどうかは, 最初に作った木の葉数 $2^w$ によって決まる. そこで $w$ を算術符号の演算精度と呼ぶ. 次に $N_i$ $(i=0,1)$ を確率の比に応じて2分割し, $N_{00}$, $N_{01}$, $N_{10}$, $N_{11}$ なる節点の集合を作る. このとき $N_i$ に含まれる節点の数が少ないと, 確率の比に合った分割はできなくなる. そこで $|N_i|$ が $2^{w-1}$ 未満の場合は, $N_i$ に含まれる節点を分枝し, 新しくできた葉全体を改めて $N_i$ とするという操作を, $|N_i|$ が $2^{w-1}$ 以上になるまで繰り返す. 図1の例では $|N_0|=3$ なので, $N_0$ を1回分枝してから分割を行っている (図1(c)参照).

以上のように分割と分枝を繰り返していくことにより, 長さ $n$ の任意のシンボル列 $s$ に対し, 2進木上の節点の集合 $N_s$ が対応する. $N_s$ の節点のなかで, 木の上でもっとも左に位置するものを $N_s$ の代表点と呼ぶ. **図1**では各代表点を○で表している. ところで, 2進木上の各節点 $x$ に対しては, 根aから出発して $x$ に到達する道が一意に存在する. このとき左の枝へ進めば0, 右の枝へ進めば1と対応づけると, すべての道に対して0-1系列を自然に対応させることができる. たとえば, 図1(c)の太線で示した道には101が対応する.

シンボル列 $s$ の符号化においては, 根から $N_s$ の代表点への道に対応した0-1系列を $s$ に対する符号語 $C_s$ とする. 復号の際には $C_s$ に従って2進木の根から出発して枝を辿っていき, 到達した節点を代表点としてもつ $N_s$ を求めることによって, 元の系列 $s$ が復元される. 以上が算術符号化の基本的な考え方である. 具体的な符号化・復号化アルゴリズムは次節において与える.

符号語が2進木上の道に対応しているという意味では算術符号は一種の木符号である. 代表的な木符号としては, Shannon-Fano 符号[1] や Huffman 符号[2] がよく知られている. これらの符号では, あらかじめ可能なすべてのシンボル列に対し生起確率を求めておいてから符号語としての木を構成する. 一方, 算術符号では, 符号化しようとするシンボル列の生起確率だけを用いて符号語を作る符号構成になっている. この特徴ゆえに, 算術符号ではシンボル長さに比例した時間で, 符号化・復号化が行えるのである.

### 2.2 算術符号の基本算法

算術符号の演算精度は符号化・復号化を通じて2進 $w$ 桁とする. またシンボル確率 $p$ $(0<p\leqq 1/2)$ を $w$ 桁以内の2進数で近似するために, 次の2条件:

$$\begin{aligned} &1)\quad p\in(0,1/2] \text{ に対し, } 0<f(p)\leqq 1/2,\\ &2)\quad \lfloor f(p)\,2^w\rfloor^{\star}=f(p)\,2^w \end{aligned} \tag{2.1}$$

を満足する関数 $f$ を任意に定める. 具体的な $f$ の一例としては,

$$p=(.\,b_1b_2\cdots b_r\cdots)\times 2^{-q}$$

* $\lfloor x\rfloor$ は実数 $x$ 以下の整数のなかで最大のものを表す.

($b_1=1$, $b_i=0$ または 1, $i=2,3,\cdots,q$ は非負整数) に対し，$p$ を2進 $r$ 桁の浮動小数点数に丸めた，

$$t_r(p)=(.b_1b_2\cdots b_r)\times 2^{-q}+b_{r+1}2^{-q-r} \qquad (2.2)$$

がある．ここで，もし $q+r\leqq w$ ならば，$t_r(p)$ は 1), 2)を満足する．

[アルゴリズム 2.1：符号化算法]

入力を長さ $N$ のシンボル列 $s_1s_2\cdots s_N$ とし，以下に示す $A_n, C_n(n=0,1,\cdots,N)$ に関する漸化式によって正整数 $C_N$ と $L$ を出力する：

初期条件を，$A_0=2^w$, $C_0=0$, $L=w$ とおく．次に $1\leqq n\leqq N$ に対し，

$$A_n=\alpha_nA_{n-1}2^{l_n} \qquad (2.3a)$$

$$C_n=\{C_{n-1}+\lfloor A_{n-1}F_n\rfloor\}2^{l_n} \qquad (2.3b)$$

$$L=L+l_n \qquad (2.3c)$$

とおく．ここで，

$$\alpha_n=\begin{cases}\lfloor A_{n-1}f(p)\rfloor/A_{n-1} & s_n=0\\ 1-\lfloor A_{n-1}f(p)\rfloor/A_{n-1} & s_n=1\end{cases} \qquad (2.3d)$$

$$F_n=\begin{cases}0 & s_n=0\\ 1-\alpha_n & s_n=1\end{cases} \qquad (2.3e)$$

また，$1\leqq n<N$ に対し $l_n$ は

$$2^{w-1}\leqq A_n<2^w \qquad (2.3f)$$

を満たす非負整数であり，さらに $l_N=0$ とする．

(アルゴリズム終了)

アルゴリズム 2.1 をよく見ると，$A_n/2^{l_n}$ は $s=s_1s_2\cdots s_n$ に対応した $N_s$ に含まれる節点の総数であり，$C_n/2^{l_n}$ は $C_s$，すなわち，根から $N_s$ の代表点に到達する道に対応する 0-1 系列であることがわかる．したがって，$s_1s_2\cdots s_N$ に対応した符号語は $C_N$ によって与えられる．

次に符号語 $C_N$ とその長さ $L$ から元のシンボル列を再生する復号化算法を示す．

[アルゴリズム 2.2：復号化算法]

自然数 $i, j(i\leqq j)$ に対し，

$$L_{ij}=\sum_{k=i}^{j}l_k \qquad (2.4)$$

を定義する．この定義より，$L=w+L_{1(N-1)}$ であることに注意しておく．

ある整数 $m$ $(1\leqq m\leqq N)$ に対し，シンボル $s_1,\cdots,s_{m-1}$ が正確に復元されているとする．ただし $m=1$ の場合は復元シンボルは空とする．このとき $C_{m-1}, A_{m-1}, l_1,\cdots,l_{m-1}$ を正確に求めることができる．これらの値と $C_N$, $L$ を入力として，シンボル $s_m$ を次の規則に従って出力する．もし，

$$C_N-C_{m-1}2^{L_{mN}}\geqq\lfloor A_{m-1}f(p)\rfloor 2^{L_{mN}} \qquad (2.5)$$

ならば，$s_m=1$，そうでなければ $s_m=0$ とする．ここで，$L_{mN}$ は $L_{mN}=L-w-L_{1(m-1)}$ より計算されることに注意する．

(アルゴリズム終了)

[定理1] アルゴリズム 2.1 によって符号化された系列がアルゴリズム 2.2 を用いて正確に復元されるための必要十分条件は，

$$f(p)\geqq 2^{-w+1} \qquad (2.6)$$

が成立することである．

定理1の証明は付録1において行う．以下では定理1より，$p<2^{-w+1}$ なる $p$ に対して $f(p)=2^{-w+1}$ とする．

### 2.3 桁上り伝播の問題

この節では，符号化を実行する際に生じる桁上りの伝播について論じる．まず符号化時において各 $C_n(n=1,2,\cdots,N)$ を，直接算術演算する最下位 $w$ ビットのレジスタ $W$，それに続く $v$ ビットのレジスタ $V$，そして $w+v$ ビットより上位のレジスタ $M$ の三つの部分に分けて考える (**図2**)．いま $s_{n-1}$ までの符号化が行われた時点で $V$ の内容がすべて1になったとする．このときもし $s_n=1$ ならば，$A_n$ を $W$ に加えた結果として桁上りが生じる．この桁上りは $V$ を越えて $M$ の領域にまで及ぶ．したがってこのままでは，符号化結果をリアルタイムでデータ伝送する，あるいはメモリ領域に移すことはできない．そこで桁上りの伝播をレジスタ $V$ までで抑えるために，$V$ の内容がすべて1ならば，$s_n$ に関する処理を行う前に $V$ を左へ1シフトして $V$ の右端に0を挿入する．さらにシフトにより $V$ からはみ出した1は $M$ に移す．このように $V$ を操作すると，桁上りの伝播は $V$ のなかで必ず抑えることができる．この方法はビット挿入法[13] (Bit-Stuffing Method) と呼ばれる．ビット挿入法を用いた場合，符号語のなかでどのビットが挿入されたかを復号時に判定できなければならない．もし符号語において $v$ 個の連続した1の列 (ラン；run) の右隣りがつねに挿入されたビット $B$ であるならば，$B=0$ のときは $B$ を $C_n$ のなかから削除し，また $B=1$ のときは $B$ の上位ビットに1を加えてから $B$ を削除す

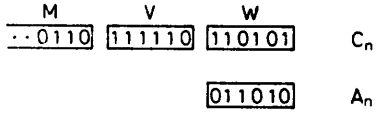


**図 2** $A_n$ と $C_n$ を表すレジスタの配置 ($v=w=6$ の場合)
Fig. 2 Arrangements of registers to represent $A_n$ and $C_n$.

| $s_n$ | $A$ | $\lfloor Af(p)\rfloor$ | $A-\lfloor Af(p)\rfloor$ | $l_n$ | $M$ | $V$ | $W$ | Comments |
|---|---|---|---|---|---|---|---|---|
| | 110110 | | | — | ……0 | 111110 | 111011 | |
| 0 | 110110 | 001101 | 101001 | 2 | ……011 | 111011 | 101100 | ; Encode $s_n$=0 & Shift |
| 1 | 110100 | 001101 | 100111 | 0 | ……011 | 111011 | 111001 | ; Encode $s_n$=1 |
| 1 | 100111 | 001001 | 011110 | 1 | ……0111 | 111000 | 000100 | ; Encode $s_n$=1 & Shift |
| 1 | 111100 | 001111 | 101101 | 0 | ……0111 | 111000 | 010011 | ; Encode $s_n$=1 |
| 1 | 101101 | 001011 | 100010 | 0 | ……0111 | 111000 | 011110 | ; Encode $s_n$=1 |
| 1 | 100010 | 001000 | 011010 | 1 | ……01111 | 110001 | 001100 | ; Encode $s_n$=1 & Shift |
| 0 | 110100 | 001101 | 100111 | 2 | ……0111111 | 000100 | 110000 | ; Encode $s_n$=1 & Shift |
| | 110100 | | | | | | | |

**図 3**　ビット挿入法の反例 ($v=w=6$, $f(p)=1/4$ の場合)

Fig. 3　A counter example of bit stuffing methods.

ればよい．ところがビット挿入法では先に述べた仮定をつねに満足するとは限らない．その反例を**図3**に示す．図3において3番目のシンボル $s_n=1$ を符号化した場合，下線を付けた部分に示すように，符号語中に連続して $v$ (=6) 個の“1”が現れる．しかしながら，$V$ レジスタの内容はすべてが“1”ではないので，桁上りを防ぐ“0”は挿入されない．一方復号の際には，符号語中に現れる連続した $v$ 個の1の次のビットは挿入されたものとして処理されるので，正しく復号されない．

この例から明らかなように，仮定を満たすためにはレジスタ $V$ 内の1ランの長さのみを監視する Langdon と Rissanen[13),15)] によるビット挿入法は不十分であり，$V$ と $M$ にまたがる1ランの長さをも監視する必要がある．

桁上りの伝播を抑止する機構を組み込んだ具体的な符号化・復号化アルゴリズムを**図4**に示す．手続き

```
    procedure ENCODE
      begin
1       C ← 0;
2       A ← 2^w;
3       L ← w;
4       COUNT ← 0;
5       for n ← 1 until N do
          begin
6           if s_n=0 then A ← ⌊Af(p)⌋
            else
              begin
7               A ← A−⌊Af(p)⌋;
8               C ← C+⌊Af(p)⌋;
9               BITTEST
              end;
10          if n<N then
11            while A<2^(w−1) do
                begin
12                if V_v=0 then COUNT ← 0
13                else COUNT ← COUNT+1;
14                L ← L+1;
15                A ← 2A;
16                C ← 2C;
17                BITTEST
                end
          end
      end
    procedure BITTEST
1     if TEST (COUNT) OR V=2^v−1 then
        begin
2         V ← V EXOR BIT (COUNT);
3         M ← 2M+1;
4         L ← L+1;
5         COUNT ← 0
        end
```

**図 4**(a)　符号化アルゴリズム

Fig. 4(a)　An encoding algorithm.

```
    procedure DECODE
      begin
1       C ← 2^(−L+w) C_N;
2       A ← 2^w;
3       COUNT ← 0;
4       for n ← 1 until N do
          begin
5           if ⌊C⌋≧⌊Af(p)⌋ then
              begin
6               s_n ← 1;
7               C ← C−⌊Af(p)⌋;
8               A ← A−⌊Af(p)⌋
              end
            else
              begin
9               s_n ← 0;
10              A ← ⌊Af(p)⌋
              end;
11          while A<2^(w−1) do
              begin
12              A ← 2A;
13              C ← 2C;
14              if W_1=0 then COUNT ← 0
15              else COUNT ← COUNT+1;
16              if COUNT=v then C ← 2C−⌊C⌋
              end
          end
      end
```

**図 4**(b)　復号化アルゴリズム

Fig. 4(b)　A decoding algorithm.

ENCODE の 4, 9, 12, 13 および 16 行目と手続き BITTEST において桁上りの伝播を一定桁で抑えるための処理を行っている．手続き DECODE では，3, 14～16行目において ENCODE で挿入されたビットの処理を行っている．なお，アルゴリズム中，$W_1$, $V_v$ はそれぞれレジスタ $W$ の右端のビット，レジスタ $V$ の左端のビットをそれぞれ表す．また手続き BITTEST における TEST$(i)$, BIT$(i)$ $(i=1,2,\cdots,v)$ はおのおの，

$$\mathrm{TEST}(i)=\overbrace{0\cdots0}^{v-i}\ \overbrace{1\cdots1}^{i}$$

$$\mathrm{BIT}(i)=\overbrace{0\cdots0\,1}^{v-i-1}\ \overbrace{0\cdots0}^{i}$$

なる 0-1 パターンを表す．最後にアルゴリズム中の OR, EXOR は，それぞれ論理和，排他論理和を表す．

## 3. 符号化効率の解析

### 3.1 符号化効率の定義

長さ $N$ のシンボル列 $s_1s_2\cdots s_N$ に対する符号語の長さ $L_N$ は，ENCODE の動作より，$L=w+\sum_{n=1}^{N-1} l_n$ に挿入されたビット数 $b_N$ を加えたものに等しい．すなわち，

$$L_N=w+\sum_{n=1}^{N-1} l_n+b_N \tag{3.1}$$

と書ける．次にシンボル確率に関する $L_N$ の期待値を $\boldsymbol{E}L_N$ とする．このとき，長さ $N$ のシンボル列に対する二値算術符号の効率 $\eta_N$ を

$$\eta_N=\frac{Nh(p)}{\boldsymbol{E}L_N} \tag{3.2}$$

で定義する．ここで $h(p)$ はエントロピ関数[1]，

$$h(p)=-p\log p-(1-p)\log(1-p) \tag{3.3}$$

である．$\eta_N$ を評価するために，まず $\boldsymbol{E}L_N$ に注目する．(3.1)より

$$\boldsymbol{E}L_N=w+\sum_{n=1}^{N-1}\boldsymbol{E}l_n+\boldsymbol{E}b_N \tag{3.4}$$

となる．ここで，ビット挿入が生じる確率はおよそ $2^{-v}$ に等しいので[13),14)]，$\boldsymbol{E}b_N\cong 2^{-v}\sum_{n=1}^{N-1}\boldsymbol{E}l_n$ である．また(2.3)より，$l_n$ は $A_{n-1}$ と $s_n$ にのみ依存して定まる確率変数であることに注意する．実際，$l_0(m)$, $l_1(m)$ $(2^{w-1}\leqq m\leqq 2^w)$ を，それぞれ

$$\begin{aligned}&2^{w-1}\leqq\lfloor mf(p)\rfloor\,2^{l_0(m)}<2^w\\&2^{w-1}\leqq(m-\lfloor mf(p)\rfloor)\,2^{l_1(m)}<2^w\end{aligned} \tag{3.5}$$

を満足する非負整数とおくとき，もし $A_{n-1}=m$ $(n\geqq 1,\ 2^{w-1}\leqq m\leqq 2^w)$ ならば，

$$l_n=\begin{cases}l_0(m), & s_n=0\\ l_1(m), & s_n=1\end{cases} \tag{3.6}$$

となる．したがって $\boldsymbol{E}l_n$ $(n\geqq 1)$ は，

$$\begin{aligned}\boldsymbol{E}l_n&=\sum_{k_1=0}^{1}\cdots\sum_{k_N=0}^{1}\mathrm{Prob}\{s_1=k_1,\cdots,s_n=k_n,\cdots,\\&\qquad s_N=k_N\}\,l_n\\&=\sum_{k_1=0}^{1}\cdots\sum_{k_n=0}^{1}\mathrm{Prob}\{s_1=k_1,\cdots,s_n=k_n\}\,l_n\\&=\sum_{m=2^{w-1}}^{2^w}\mathrm{Prob}\{A_{n-1}=m\}\,\{pl_0(m)\\&\qquad+(1-p)l_1(m)\}\end{aligned} \tag{3.7}$$

と表せる．$\mathrm{Prob}\{A_{n-1}=m\}$ $(n\geqq 1,\ 2^{w-1}\leqq m\leqq 2^w)$ は一般に $n$ の関数として与えられるが，$n$ の増大とともに一定値 $\mathrm{Prob}\{A_\infty=m\}$ に近づく．$\boldsymbol{p}=(\mathrm{Prob}\{A_\infty=2^{w-1}\},\ \mathrm{Prob}\{A_\infty=2^{w-1}+1\},\cdots,\mathrm{Prob}\{A_\infty=2^w\})$ とおくと，(2.3)の関係から得られる $A_n$ の推移確率行列 $\boldsymbol{P}$ に対し，$\boldsymbol{p}$ は $\boldsymbol{pP}=\boldsymbol{p}$ を満足する確率ベクトルとして求まる．例として，$f(p)=t_2(p)$, $w=4$ の場合について，$t_2(p)=(.0100)$ となる $p$ に対する $\boldsymbol{P}$ と $\boldsymbol{p}$ を**表1**に示す．(3.7)の右辺の $\mathrm{Prob}\{A_{n-1}=m\}$ を $\mathrm{Prob}\{A_\infty=m\}$ に置き換えたものを，(3.4)の右辺に代入すれば，十分大きな $N$ に対して $\eta_N$ の値を近似的に評

**表 1** $f(p)=t_2(p)$ に対する $\boldsymbol{P},\boldsymbol{p}$ の一例 $(t_2(p)=0.0100,\ w=4)$

Table 1 An example of $\boldsymbol{P}$ and $\boldsymbol{p}$ for $f$ such that $f(p)=t_2(p)$.

$\boldsymbol{P}=$

| $A_{n-1}\backslash A_n$ | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 | 10000 |
|---|---|---|---|---|---|---|---|---|---|
| 1000 | $p$ | 0 | 0 | 0 | $1-p$ | 0 | 0 | 0 | 0 |
| 1001 | $p$ | 0 | 0 | 0 | 0 | 0 | $1-p$ | 0 | 0 |
| 1010 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1011 | $p$ | $1-p$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1100 | 0 | $1-p$ | 0 | 0 | $p$ | 0 | 0 | 0 | 0 |
| 1101 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1110 | 0 | 0 | 0 | $1-p$ | $p$ | 0 | 0 | 0 | 0 |
| 1111 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 10000 | $p$ | 0 | 0 | 0 | $1-p$ | 0 | 0 | 0 | 0 |

$\boldsymbol{p}=\frac{1}{S}$

| |
|---|
| $p/(1-p)+p(1-p)$ |
| 1 |
| 0 |
| $(1-p)^2$ |
| $1/(1-p)-(1-p)^2$ |
| 0 |
| $1-p$ |
| 0 |
| 0 |

$S=(1-p)/(2+(1-p)^2(1+p))$

価することができる (3.2節参照). しかしながら，十分大きな $w$ に対し $\boldsymbol{p}$ を計算することは現実には困難なので，$\eta_N$ の上界・下界を求めることが必要となる．そこでまず $\boldsymbol{EL}_N$ の評価式を以下に示す．

[定理2] $0<p\leqq 1/2$, $0<d<1-p$ に対し，

$$g(p,d)=-p\log(p+d)-(1-p)\log(1-p-d) \tag{3.8}$$

とおく．さらに

$$\Delta_f^*=\begin{cases}2^{-w} & , f(p)\leqq 2^{-w+1}\\ f(p)/2 & , 2^{-w+1}<f(p)\leqq 2^{-w+2}\\ f(p)-2^{-w+1} & , 2^{-w+2}<f(p)\end{cases} \tag{3.9}$$

に対し，関数 $g_L(p)$, $g_U(p)$ を次のように定める．

$$g_L(p)=\begin{cases}g(p,2^{-w}-p) & , p<2^{-w}\\ g(p,0) & , 2^{-w}\leqq p<2^{-w+1}\\ g(p,f(p)-p) & , p\geqq 2^{-w+1}, f(p)-p<0\\ g(p,0) & , p\geqq 2^{-w+1},\\ & \quad f(p)-p\geqq 0,\ \Delta_f^*-p<0\\ g(p,\Delta_f^*-p) & , p\geqq 2^{-w+1},\\ & \quad f(p)-p\geqq 0,\ \Delta_f^*-p\geqq 0\end{cases}$$

$$g_U(p)=\begin{cases}g(p,2^{-w+1}-p) & , p<2^{-w}\\ \max\{g(p,2^{-w}-p),\ g(p,2^{-w+1}-p)\}, & \\ \qquad 2^{-w}\leqq p<2^{-w+1} & \\ \max\{g(p,f(p)-p),\ g(p,\Delta_f^*-p)\}, & \\ \qquad p\geqq 2^{-w+1} & \end{cases}$$

このとき，

$$(N-1)g_L(p)-1\leqq\sum_{n=1}^{N-1}\boldsymbol{El}_n\leqq(N-1)g_U(p) \tag{3.10}$$

が成立する．

定理2の証明は付録2において行う．定理2より $\eta_N$ に関しては次の評価が得られる．

[系1]

$$\frac{h(p)}{w/N+(1-1/N)(1+2^{-v})g_U(p)}\leqq\eta_N\leqq\frac{h(p)}{(w-1)/N+(1-1/N)(1+2^{-v})g_L(p)} \tag{3.11}$$

ただし，(3.11)においてビット挿入が起こる確率は $2^{-v}$ とおいた．

### 3.2 効率の評価式の有効性について

$g_U(p)$ と $g_L(p)$ の定義から明らかなように，$\eta_N$ の上界と下界の差は，とくに $w$ に依存し，$w$ が小さくなるほどその差は大きくなる．そこで評価式がどの程度信頼できるかを調べるために，アルゴリズム 2.1, 2.2 に基づく符号化シミュレーションを行った．その結果得られた符号化効率を**図5**に示す．ここで，$f(p)=t_2(p)$, $v=16$, $w=4, 8, 16$ とした．同図には $w=8, 16$ における(3.11)の上界，下界も合わせて示している．$w=16$ の場合は，$2^{-11}\leqq p\leqq 2^{-1}$ の全域で上

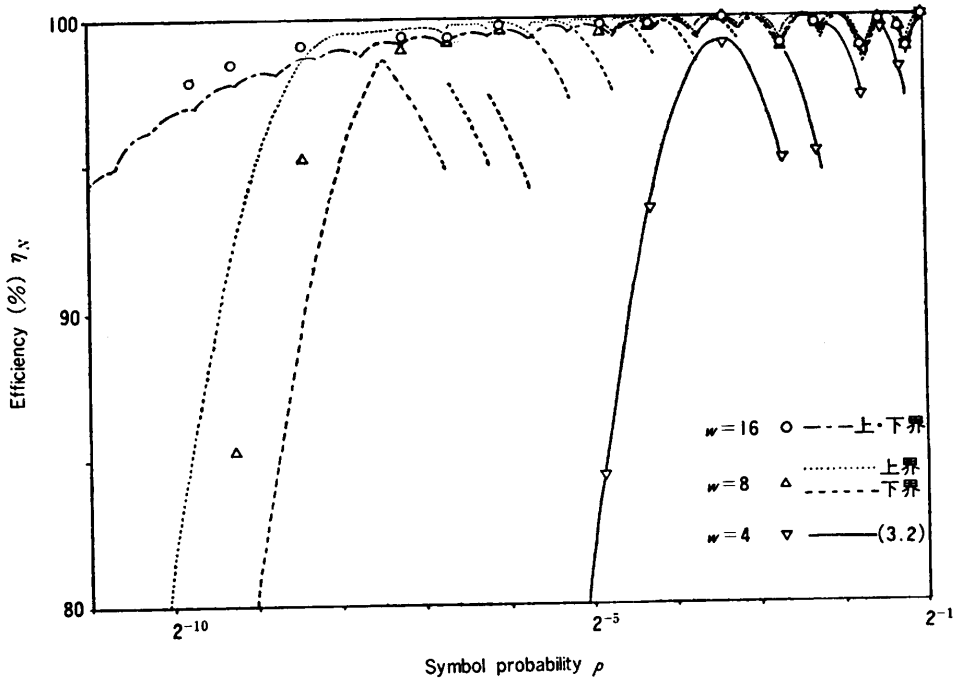


**図5** $\eta_N$ の評価式と符号化シミュレーション結果の比較 ($v=16$, $N=2^{16}$, $w=4, 8, 16$)

Fig. 5 Comparison of evaluated bounds of $\eta_N$ and results of coding simulations.

界と下界はほぼ一致しており，シミュレーション結果ともよく合っている．$w=8$ の場合では，上界と下界の差は $p<2^{-8}$ においては顕著であるが，$p\geqq 2^{-8}$ の部分ではその差もたかだか5%であり，$p$ に対するシミュレーション結果の挙動をよく把えている．一方，$w=4$ とすると，図には示していないが，上限と下限の差は $p\geqq 2^{-4}$ の範囲においても10%以上となり，効率をよく評価するとはいえない．しかしながら，この場合は，$A_n$ の生起確率が表1に示した例のように，容易に計算できるので，$\eta_N$ を直接求めることができる．実際，シミュレーション結果は，この $\eta_N$ の値とよく合う（図5の実線部分を参照）．以上のことより，小さな $w$ に対しては，評価式を用いるより直接 $\eta_N$ を計算するほうがよいが，直接 $\eta_N$ を求めるのが困難な，比較的大きな $w$ の範囲（$w\geqq 8$）では，(3.11)の評価式は有効であるといえる．

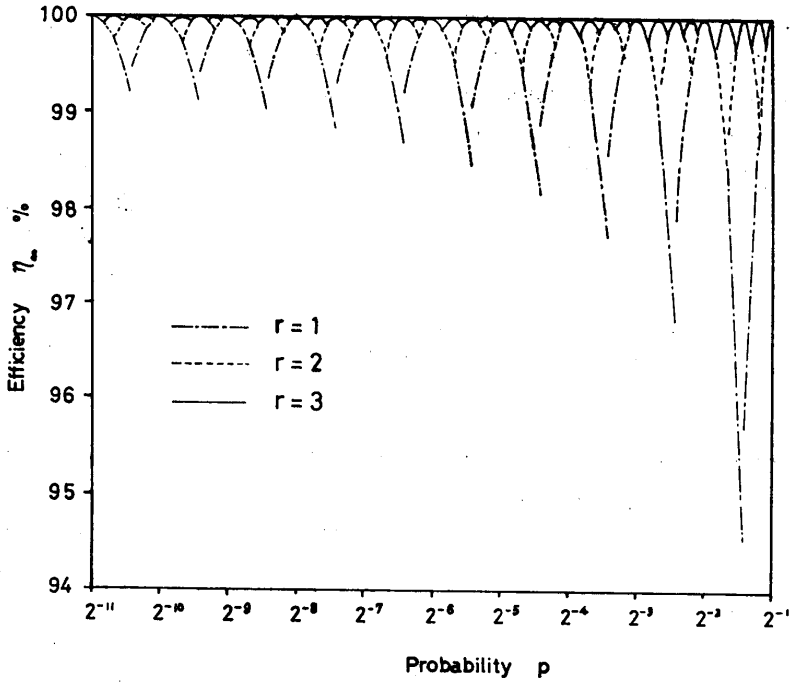


**図 6** $f(p)=t_r(p)$ $(r=1,2,3)$ の場合の効率 $\eta_\infty$ のグラフ
Fig. 6 A graph of $\eta_\infty$ in case of $f(p)=t_r(p)$ $(r=1,2,3)$.

## 4. 効率と符号化パラメータとの関係

(3.11)から明らかなように，$\eta_N$ は $N$ のみならず，演算精度 $w$，ビット挿入機構のレジスタ長さ $v$，さらに確率 $p$ の近似関数 $f(p)$ に依存している．この章では，あらかじめ設定した効率（90%, 98%）を達成するという条件のもとで，条件にもっとも適した $f(p)$, $w$, $v$ の設計を行う．

### 4.1 $f(p)$ の設計

まず $f(p)$ と効率の関係を明らかにするために，理想の効率として

$$\eta_\infty=\lim_{\substack{N,v,w\to\infty\\ w/N\to 0}}\eta_N \tag{4.1}$$

を定義する．系1より $\eta_\infty$ は存在して

$$\eta_\infty=h(p)/g(p,f(p)-p) \tag{4.2}$$

である．さらに以下の議論では，$f(p)$ として2章で定義した $t_r(p)$ を用いる．$r=1,2,3$ の場合について，$\eta_\infty$ のグラフを**図6**に示す．この図より，$r=2$ に対し，$2^{-11}\leqq p\leqq 2^{-1}$ の範囲で98%以上の効率が達成されている．それゆえ，$f(p)=t_2(p)$ と定める．

### 4.2 $w$ と $v$ の設計

与えられた $w$ と $v$ に対し，あらかじめ設定された効率を達成するシンボル確率の範囲を，$\eta_N$ あるいは $\eta_N$ の評価式を用いて求めた．このときの各パラメータの値を**表2**に示す．結果を**表3**にまとめておく．表3では，$2^{-m}\leqq p\leqq 2^{-1}$，$m=2,3,\cdots,10$ なる $p$ の範囲において，設定された効率を達成する $w, v$ のなかで $w+v$ が最小となる組が示されている．表3より，$v$ については効率の設定値およびシンボル確率 $p$ によらず，$v=8$ とすれば十分であること，また $w$ に関してはシンボル確率が大きくなるほど，用いる $w$ の値は小さ

**表 2** 効率の評価に用いたパラメータの値
Table 2 List of parameters used in evaluating coding efficiency.

| | |
|---|---|
| $w$ と $v$ の値 | 4, 8, 12, 16 |
| シンボル長さ | $2^{10}$, $2^{16}$ |
| 効率の設定値 | 90%, 98% |
| $f(p)$ | $t_2(p)$ |
| $p$ | $2^{-10}\leqq p\leqq 2^{-1}$ |
| $\eta_N$ の評価式 | $w=4$ のとき：$A_n$ の推移行列から(3.2)を近似計算したもの<br>$w>4$ のとき：(3.11)の下界 |

表 3 効率の評価式から各 $p$ に対して設計された $w$ と $v$ の値

Table 3 The values of $w$ and $v$ designed for each $p$ with the bounds of efficiency.

| $p$ | 90% | | 98% | |
|---|---|---|---|---|
| | $w$ | $v$ | $w$ | $v$ |
| $2^{-2}$ | 4 | 4 | — | — |
| $2^{-3}$ | 4 | 8 | — | — |
| $2^{-4}$ | 4 | 8 | — | — |
| $2^{-5}$ | 8 | 8 | — | — |
| $2^{-6}$ | — | — | — | — |

$N=2^{10}$
(a)

| $p$ | 90% | | 98% | |
|---|---|---|---|---|
| | $w$ | $v$ | $w$ | $v$ |
| $2^{-2}$ | 4 | 4 | 8 | 8 |
| $2^{-3}$ | 4 | 8 | 8 | 8 |
| $2^{-4}$ | 4 | 8 | 8 | 8 |
| $2^{-5}$ | 8 | 4 | 12 | 8 |
| $2^{-6}$ | 8 | 8 | 12 | 8 |
| $2^{-7}$ | 8 | 8 | 12 | 8 |
| $2^{-8}$ | 8 | 8 | 12 | 8 |
| $2^{-9}$ | 12 | 4 | 16 | 8 |
| $2^{-10}$ | 12 | 8 | — | — |

$N=2^{16}$
(b)

くて済むことがわかる．最後に表3の(a),(b)の比較より，高い効率を得るためには，入力系列の長さをまず十分大きくとることが必要であることに注意しておく．

## 5. むすび

二値算術符号の演算精度やシンボル確率の近似法などの符号化パラメータが符号化効率に与える影響を示す評価式を導出し，設定した効率を達成する符号化パラメータの設計を行った．本論文では情報源を定常無記憶二値情報源に限定して議論したが，記憶のある情報源や確率分布が未知である情報源に対しても拡張は容易である[12)~14)]．したがって，今後の算術符号の応用として，データ伝送・画像の符号化のみならず，データ・ベースに蓄積された，必ずしも一様な統計的性質をもたない膨大なデータの圧縮に効果的であると思われる．ただ，算術符号はノイズの影響に非常に弱く，符号語のうち1ビットでも誤ると，一般に復号結果は元のシンボル列とは大きく異なる．この点を克服するのが今後の課題である．

## 参考文献


1) Shannon, C. E.: A Mathematical Theory of Communication, *Bell Syst. Tech. J.*, Vol. 27, No. 3, pp. 379-423 and No. 4, pp. 624-656 (1948).
2) Huffman, D. A.: A Method for the Construction of Minimum-Redundancy Codes, *Proc. IRE*, Vol. 40, No. 9, pp. 1098-1101 (1952).
3) Elias, P.: Unpublished result explained by 5).
4) Lynch, T. J.: Sequence Time Coding for Data Compression, *Proc. IEEE*, Vol. 54, No. 10, pp. 1490-1491 (1966).
5) Jelinek, F.: Buffer Overflow in Variable Length Coding of Fixed Rate Sources, *IEEE Trans. Inform. Theory*, Vol. IT-14, No. 3, pp. 490-501 (1968).
6) Cover, T. M.: Enumerative Source Encoding, *IEEE Trans. Inform. Theory*, Vol. IT-19, No. 1, pp. 73-77 (1973).
7) Ohnishi, R., Ueno, Y. and Ono, F.: Optimization of Facsimile Data Compression, *NTC*, Vol. 49, No. 1, pp. 1-6 (1977).
8) Pasco, R. C.: Source Coding Algorithms for Fast Data Compression, Ph. D. thesis, Dept. of Electrical Engineering, Stanford Univ., CA (1976).
9) Rissanen, J.: Generalized Kraft Inequality and Arithmetic Coding, *IBM J. Res. Dev.*, Vol. 20, No. 3, pp. 198-203 (1976).
10) Rissanen, J. and Langdon, G.: Arithmetic Coding, *IBM J. Res. Dev.*, Vol. 23, No. 2, pp. 149-162 (1979).
11) Guazzo, M.: A General Minimum-Redundancy Source-Coding Algorithm, *IEEE Trans. Inform. Theory*, Vol. IT-26, No. 1, pp. 15-25 (1980).
12) Rissanen, J. and Langdon, G.: Universal Modeling and Coding, *IEEE Trans. Inform Theory*, Vol. IT-27, No. 1, pp. 12-22 (1981).
13) Langdon, G. and Rissanen, J.: Compression of Black White Images with Arithmetic Coding, *IEEE Trans. Comm.*, Vol. COM-29, No. 6, pp. 858-867 (1981).
14) 森田啓義，北田 茂，有本 卓：算術符号化方式の符号化効率について，情報理論とその応用研究会，第5回，pp. 111-118 (1982).
15) Langdon, G. and Rissanen, J.: A Simple General Binary Source Codes, *IEEE Trans. Inform. Theory*, Vol. IT-28, No. 5, pp. 800-803 (1982).


## 付録 1 定理1の証明

まず $A_n$, $C_n$ に対し次の関係式が成立することに注意しておく．

[補題1] $0<m<n\leqq N$ なる整数 $m$, $n$ に対し，

$$1)\quad C_n=C_{m-1}2^{L_{mn}}+\sum_{i=m}^{n}\lfloor A_{i-1}F_i\rfloor 2^{L_{in}} \qquad (A.1)$$

$$2)\quad A_n=A_m 2^{L_{(m+1)n}}\prod_{i=m+1}^{n}\alpha_i \qquad (A.2)$$

(補題の証明) (2.3b)を展開すると

$$C_n=\sum_{i=1}^{n}\lfloor A_{i-1}F_i\rfloor 2^{L_{in}}$$

となる．この式はさらに以下のように変形される．

$$C_n=\sum_{i=1}^{m-1}\lfloor A_{i-1}\rfloor 2^{L_{in}}+\sum_{i=m}^{n}\lfloor A_{i-1}F_i\rfloor 2^{L_{in}}$$
$$=\{\sum_{i=1}^{m-1}\lfloor A_{i-1}F_i\rfloor 2^{L_{i(m-1)}}\}2^{L_{mn}}+\sum_{i=m}^{n}\lfloor A_{i-1}F_i\rfloor 2^{L_{in}}$$
$$=C_{m-1}2^{L_{mn}}+\sum_{i=m}^{n}\lfloor A_{i-1}F_i\rfloor 2^{L_{in}}$$

よって(A.1)が示された．(A.2)は $A_n$ の定義から明らかである．(証明終了)

以上の準備のもとに，定理1を証明しよう．

1) 必要性：復号化が正確に行われているならば，任意の $n\geqq 1$ に対し，$A_n>0$ が成立しなければならない．なぜなら，もしある $n$ に対し，$A_n=0$ ならば，それ以降の $m>n$ に対し，$A_m=0$ となり，元のシンボル列を一意に復元できないからである．

$A_n>0$ と同値な条件は，

$$A_{n-1}f(p)\geqq 1$$

で，これが任意のシンボル列 $s_1s_2\cdots s_{n-1}s_n$ に対して成立するためには

$$2^{w-1}f(p)\geqq 1$$

が成立することが必要である．

2) 十分性：部分シンボル列* $s_1\cdots s_{m-1}(1\leqq m<N)$ がすでに正確に復元されているとする．もし $s_m=1$ ならば，補題1の(A.1)から，

$$C_N=C_{m-1}2^{L_{mN}}+\lfloor A_{m-1}f(p)\rfloor 2^{L_{mN}}+\sum_{i=m+1}^{N}\lfloor A_{i-1}F_i\rfloor 2^{L_{iN}}$$
$$\geqq C_{m-1}2^{L_{mN}}+\lfloor A_{m-1}f(p)\rfloor 2^{L_{mN}}$$

が成立する．なぜなら，任意の $i$ に対し $\lfloor A_{i-1}F_i\rfloor\geqq 0$ であるから．このとき (2.5) より $s_m$ は正確に1として復元される．一方，$s_m=0$ の場合は

$$\lfloor A_{m-1}f(p)\rfloor 2^{L_{mN}}>\sum_{i=m+1}^{N}\lfloor A_{i-1}F_i\rfloor 2^{L_{iN}} \qquad (A.3)$$

を示せばよい．(A.3)の右辺は(A.2)を用いて次のように変形できる．

$$\sum_{i=m+1}^{N}\lfloor A_{i-1}F_i\rfloor 2^{L_{iN}}\leqq\sum_{i=m+1}^{N}A_{i-1}F_i\,2^{L_{iN}}$$
$$=A_mF_{m+1}2^{L_{(m+1)N}}+\sum_{i=m+2}^{N}\left(\prod_{j=m+1}^{i-1}\alpha_jA_m2^{L_{(m+1)(i-1)}}\right)F_i2^{L_{iN}}$$
$$=A_m2^{L_{(m+1)N}}\left(F_{m+1}+\sum_{i=m+2}^{N}\prod_{j=m+1}^{i-1}\alpha_jF_i\right) \qquad (A.4)$$

また(A.3)の左辺は次のように変形される．

$$\lfloor A_{m-1}f(p)\rfloor 2^{L_{mN}}=\lfloor A_{m-1}f(p)\rfloor 2^{t_m}2^{L_{(m+1)N}}=A_m2^{L_{(m+1)N}} \qquad (A.5)$$

ところで(2.6)が成立していることから，任意の $m$ に対して $A_m>0$ である．したがって，(A.4)と(A.5)から

$$F_{m+1}+\sum_{i=m+2}^{N}\prod_{j=m+1}^{i-1}\alpha_jF_i<1 \qquad (A.6)$$

が示されれば，(A.3)が成立する．そこで，任意の $i$ に対して，

$$F_i<1,\quad F_i+\alpha_i\leqq 1$$

が成立することに注意する．これらの不等式を用いて(A.6)の左辺を変形してゆくと次のようになる．

$$F_{m+1}+\sum_{i=m+2}^{N}\prod_{j=m+1}^{i-1}\alpha_jF_i$$
$$=F_{m+1}+\sum_{i=m+1}^{N-1}\prod_{j=m+1}^{i}\alpha_jF_{i+1}$$
$$=F_{m+1}+\sum_{i=m+1}^{N-2}\prod_{j=m+1}^{i}\alpha_jF_{i+1}+\prod_{j=m+1}^{N-1}\alpha_jF_N$$
$$<F_{m+1}+\sum_{i=m+1}^{N-2}\prod_{j=m+1}^{i}\alpha_jF_{i+1}+\prod_{j=m+1}^{N-1}\alpha_j$$
$$=F_{m+1}+\sum_{i=m+1}^{N-3}\prod_{j=m+1}^{i}\alpha_jF_{i+1}+\prod_{j=m+1}^{N-2}\alpha_j(F_{N-1}+\alpha_{N-1})$$
$$\leqq F_{m+1}+\sum_{i=m+1}^{N-3}\prod_{j=m+1}^{i}\alpha_jF_{i+1}+\prod_{j=m+1}^{N-2}\alpha_j \qquad (A.7)$$

(A.7)を繰り返し変形してゆくと最終的に

$$F_{m+1}+\sum_{i=m+2}^{N}\prod_{j=m+1}^{i-1}\alpha_jF_i<F_{m+1}+\alpha_{m+1}\leqq 1$$

を得る．これは(A.6)にほかならない．(証明終了)

## 付録2 定理2の証明

(3.1)の右辺の第2項 $\sum_{n=1}^{N-1}l_n$ に注目し，(2.3a)を用いて変形してゆくと次のようになる．

$$\sum_{n=1}^{N-1}l_n=\sum_{n=1}^{N-1}\log(A_n/(\alpha_nA_{n-1}))$$
$$=\log\left[\prod_{n-1}^{N-1}(A_{n-1}/(\alpha_nA_{n-1}))\cdot(A_{N-1}/A_0)\right]$$

* $m=1$ の場合は部分シンボル列は空とする．

$$=-\sum_{n=1}^{N-1}\log\alpha_n+\log(A_{N-1}/A_0) \quad (A.8)$$

ここで，$2^{w-1}\leqq A_{N-1}\leqq 2^w-1<A_0$ と(A.8)より，

$$-1-\sum_{n=1}^{N-1}\boldsymbol{E}\log\alpha_n\leqq\sum_{n=1}^{N-1}\boldsymbol{E}\,l_n<-\sum_{n=1}^{N-1}\boldsymbol{E}\log\alpha_n \quad (A.9)$$

が導かれる．次に $\boldsymbol{E}(-\log\alpha_n)$ を(3.7)と同様に変形すると，

$$\boldsymbol{E}(-\log\alpha_n)=\sum_{m=2^{w-1}}^{2^w-1}\mathrm{Prob}\{A_{n-1}=m\}\,g(p,\Delta_m) \quad (A.10)$$

となる．ここで $g(p,\Delta_m)$ は(3.8)で定義された関数であり，$\Delta_m=-p+\lfloor mf(p)\rfloor/m$ とおいた．

一般に $g(p,\delta)$, $0\leqq p\leqq 1/2$, $-p<\delta<1-p$ は $\delta$ に関して凸関数であり，かつ $\delta=0$ で最小値 $g(p,0)=h(p)$ をとることに注意しておく．

(1) $p<2^{-w+1}$ の場合

$f(p)=2^{-w+1}$ なので，$2^{w-1}\leqq m<2^w$ に対し $\lfloor mf(p)\rfloor=1$. よって，$\Delta_m=-p+1/m$. したがって，$\Delta_m$ は

$$-p+2^{-w}<\Delta_m\leqq -p+2^{-w+1}$$

の範囲をとりうる．したがって，$p\leqq 2^{-w}$ ならば，

$$g(p,-p+2^{-w})\leqq\boldsymbol{E}(-\log\alpha_n)\leqq g(p,-p+2^{-w+1})$$

$2^{-w}\leqq p<2^{-w+1}$ ならば，

$$g(p,0)\leqq\boldsymbol{E}(-\log\alpha_n)\leqq\max[g(p,2^{-w}-p),\ g(p,2^{-w+1}-p)]$$

となる．

(2) $p\geqq 2^{-w+1}$ の場合

一般に，実数 $x$ に対し，$x-1<\lfloor x\rfloor\leqq x$ より，

$$f(p)-1/m<\lfloor mf(p)\rfloor/m\leqq f(p)$$

また $f(p)$ の定義から，$2^{w-1}\leqq m<2^w$ に対し，つねに

$$\lfloor mf(p)\rfloor/m\geqq 1/m$$

が成立する．したがって，

$$\min_{2^{w-1}\leqq m<2^w}\max[1/m,\ f(p)-1/m]\leqq\lfloor mf(p)\rfloor/m\leqq f(p) \quad (A.11)$$

(A.11)の最左辺の式に対し，実際に min, max 演算を行うと，

$$\min_{2^{w-1}\leqq m<2^w}\max\left[\frac{1}{m},\ f(p)-\frac{1}{m}\right]=\begin{cases}=2^{-w} & f(p)<2^{-w+1}\\ \geqq f(p)/2 & 2^{-w+1}\leqq f(p)\leqq 2^{-w+2}\\ =f(p)-2^{-w+1} & f(p)>2^{-w+2}\end{cases} \quad (A.12)$$

となる．(A.12)の右辺を $\Delta_f^*$ とおくと，$\Delta_m$ は

$$-p+\Delta_f^*\leqq\Delta_m\leqq -p+f(p)$$

の範囲をとりうる．したがって

$$\boldsymbol{E}(-\log\alpha_n)\leqq\max\{g(p,-p+\Delta_f^*),g(p,-p+f(p))\}$$

となる．また $p$ のとりうる値により，

i) $p\geqq f(p)$ ならば，$\boldsymbol{E}(-\log\alpha_n)\geqq g(p,-p+f(p))$

ii) $\Delta_f^*\leqq p<f(p)$ ならば，$\boldsymbol{E}(-\log\alpha_n)\geqq g(p,0)$

iii) $p<\Delta_f^*$ ならば，$\boldsymbol{E}(-\log\alpha_n)\geqq g(p,-p+\Delta_f^*)$

となる．各 $p$ の範囲に対し，上で求めた $\boldsymbol{E}(-\log\alpha_n)$ の上界および下界をそれぞれ，$g_U(p)$, $g_L(p)$ として(A.9)に代入すれば，(3.9)が得られる．

(証明終了)

(昭和58年8月15日受付)
(昭和59年2月14日採録)