情報理論（No. 4）

2016/10/8

今後の講義予定

10/15：通常（第5回）

10/22：休講（平塚祭）

10/29：通常（第6回）

講義目次

- 5.　情報源の符号化
  - 5.1 平均符号長
  - 5.2 コンパクト符号
  - 5.3 ハフマン符号
  - 5.4 他の符号化：シャノン-ファノ符号
  - 5.5 ランレングス符号化
  - 5.6 拡大情報源の符号化
  - 5.7 拡大情報源のハフマン符号化
  - 5.8 符号化の限界
  - 5.9 平均符号長の下限
  - 5.10 符号長の下限定理
  - 5.11 符号長の下限と上限
  - 5.12 シャノン第1定理：情報源符号化定理
  - 5.13 符号の効率と冗長度

5. 情報源の符号化

5.1 平均符号長
5.2 コンパクト符号
5.3 ハフマン符号
5.4 他の符号化：シャノン-ファノ符号
5.5 ランレングス符号化
5.6 拡大情報源の符号化
5.7 拡大情報源のハフマン符号化
5.8 符号化の限界
5.9 平均符号長の下限
5.10 符号長の下限定理
5.11 符号長の下限と上限
5.12 シャノン第1定理：情報源符号化定理
5.13 符号の効率と冗長度

5.1 平均符号長

- これまでの復習

  クラフトの不等式を満たすような瞬時復号可能な符号は一般に複数の実現法がある。

  これらの中から効率のよい符号を選ぶ必要がある。

  効率を測る尺度として符号語の長さがある。

- 定義：情報源シンボル$s_1, s_2, \ldots, s_q$を符号語$X_1, X_2, \ldots, X_q$に変換するとする。情報源シンボルの確率を$P_1, P_2, \ldots, P_q$、符号語長を$L_1, L_2, \ldots, L_q$とする。

  この時、符号の平均符号長は

  $$L = \Sigma_i P_i * L_i = P_1 * L_1 + P_2 * L_2 + \cdots + P_q * L_q$$

  で与えられる。

5.1 具体的符号の平均符号長

情報源Ｓ	生起確率	符号Ａ	符号Ｂ
s_1	1／4	００	０
s_2	1／4	０１	１０
s_3	1／4	１０	１１０
s_4	1／4	１１	１１１

情報源Ｈ	生起確率	符号Ａ	符号Ｂ
s_1	1／2	００	０
s_2	1／4	０１	１０
s_3	1／8	１０	１１０
s_4	1／8	１１	１１１

5.1 具体的符号の平均符号長

情報源Ｓに対しては、

符号Ａ　$L=2*(1/4)+2*(1/4)+2*(1/4)+2*(1/4)=2$

符号Ｂ　$L= 1*(1/4)+2*(1/4)+3*(1/4)+3*(1/4)=2.25$

従って符号Ａのほうが効率がよい。

情報源Ｈに対しては、

符号Ａ　$L=2*(1/2)+2*(1/4)+2*(1/8)+2*(1/8)=2$

符号Ｂ　$L= 1*(1/2)+2*(1/4)+3*(1/8)+3*(1/8)=1.75$

この場合は、符号Ｂのほうが効率がよい。

＝＞生起確率の高い記号に短い符号を割当てることで効率がよくなり、直感的にも合う。

5.2 コンパクト符号

- **定義**

情報源とその生起確率が与えられたとき、<u>平均符号長が最小になる（かつ一意復号可能な）符号</u>を<u>コンパクト符号</u>と呼ぶ。

- **基本的な問題**

与えられた情報源に対して、コンパクト符号をどのように作るか（見つけるか）？

その符号が最小の平均符号長をもつこと（コンパクト符号であること）は保証されるか？

＝＞ハフマン（Huffman）が解を与えた(1952年）

5.3 ハフマン符号

情報源	確率	S_1	S_2	S_3	S_4
s_1	0．4	0．4	0．4	0．4	0．6
s_2	0．3	0．3	0．3	0．3	0．4
s_3	0．1	0．1	0．2	0．3	
s_4	0．1	0．1	0．1		
s_5	0．06	0．1			
s_6	0．04				

作り方1（前半：縮退情報源の作成）

- 元の情報源を確率順に並べる。
- その中の最も確率の低い２つを合わせた１つ記号の少ない情報源を作る＝＞<u>縮退情報源</u>
- 確率の順に並べかえる
- 最後に２つの記号をもつ情報源まで繰り返す

5.3 ハフマン符号の構成法(A)

情報源	確率		S_1		S_2		S_3		S_4	
s_1	0.4	*1*	0.4	*1*	0.4	*1*	0.4	*1*	0.6	*0*
s_2	0.3	*00*	0.3	*00*	0.3	*00*	0.3	*00*	0.4	*1*
s_3	0.1	*011*	0.1	*011*	0.2	*010*	0.3	*01*		
s_4	0.1	*0100*	0.1	*0100*	0.1	*011*				
s_5	0.06	*01010*	0.1	*0101*						
s_6	0.04	*01011*								

赤字の系列が符号語

作り方2（後半：符号の割り当て）

- **最後の縮退情報源（S_4）に0と1を割当てる**
- **S_4の１つ前のS_3の合成した記号を分解し0と1を割当てる**
- **以下、もとの情報源まで繰り返す**

5.3 ハフマン符号の構成法(B)（別の縮退法）

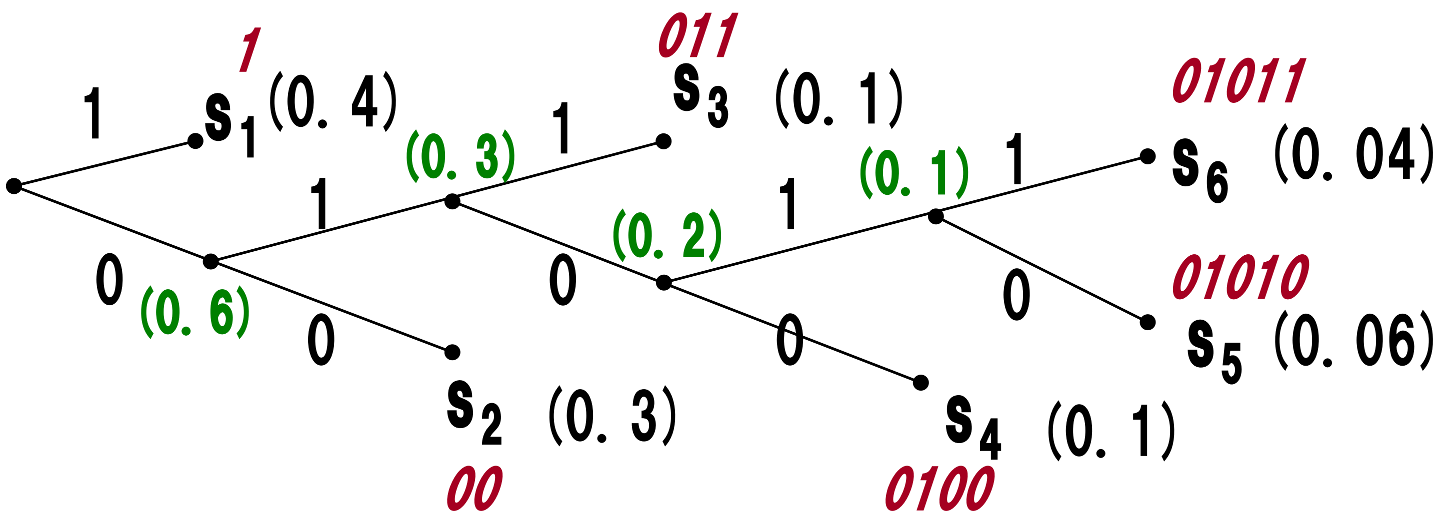
情報源	確率		S_1		S_2		S_3		S_4	
s_1	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
s_2	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
s_3	0.1	0100	0.1	011	0.2	010	0.3	01		
s_4	0.1	0101	0.1	0100	0.1	011				
s_5	0.06	0110	0.1	0101						
s_6	0.04	0111								

- 合成した記号が同じ確率をもつとき、並べ方に自由度が生じる
- 上例は、S_1に縮退させた時、S_5とS_6の合成をS_3、S_4より上に並べた場合の符号構成法

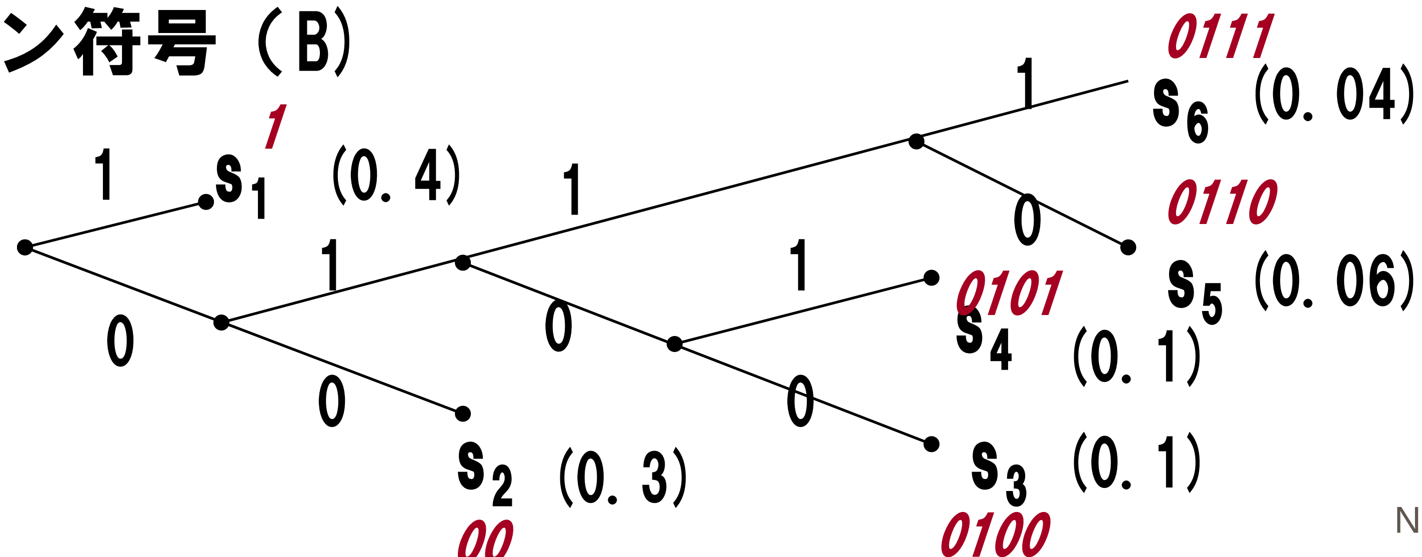
5.3 ハフマン符号の木構造表現

（　）内の数字はシンボルの生起確率
*011*など斜体の赤数字が符号語

ハフマン符号（A）



ハフマン符号（B）



5.3 ハフマン符号：ｒ元符号への適用

- ｒ元符号は、0からr-1までの数値に符号化する
- 特徴
  - **縮退情報源**（生起確率の低いシンボルを合わせて作った情報源）は、１つ前の情報源よりr-1だけシンボルが少ない。（２元符号は１個少ない）
  - **縮退した最後の情報源**は、r個のシンボルをもつことが望ましい。これは最初の情報源が、r+n(r-1)個のシンボルをもつときに限られる。（ｎは整数）
    - 最後から逆にたどり、r個、2r-1個、3r-2個、4r-3個、…のシンボルが各縮退情報源に残る必要がある。
    - もし、最初の情報源がちょうどr+n(r-1)個のシンボルをもたない時は、ダミーシンボルを付加する必要がある。

5.3 ハフマン符号：r 元符号への適用

（例）3元符号と4元符号の場合：

- 3元符号は {0, 1, 2} の３種類で符号化
  - 最後の縮退情報源は3個のシンボル、
  - 最後から1つ前の縮退情報源は、3+2=5個のシンボル
  - その前の縮退情報源は、3+2+2=7個のシンボル
  - ・・・
  - 最初の情報源は、3+2n個のシンボルであればダミー不要
- 4元符号は {0, 1, 2, 3} の４種類で符号化
  - 最後の縮退情報源は4個のシンボル、
  - 最後から1つ前の縮退情報源は、4+3=7個のシンボル
  - その前の縮退情報源は、4+3+3=10個のシンボル
  - ・・・
  - 最初の情報源は、4+3n個のシンボルであればダミー不要

5.3 ハフマン符号：４元符号の例

赤字の数値列が符号語（4元符号）

情報源	確率		S１		S２		S３	
s１	0. 22	2	0. 22	2	0. 23	1	0. 40	0
s２	0. 15	3	0. 15	3	0. 22	2	0. 23	1
s３	0. 12	00	0. 12	00	0. 15	3	0. 22	2
s４	0. 10	01	0. 10	01	0. 12	00	0. 15	3
s５	0. 10	02	0. 10	02	0. 10	01		
s６	0. 08	03	0. 08	03	0. 10	02		
s７	0. 06	11	0. 07	10	0. 08	03		
s８	0. 05	12	0. 06	11				
s９	0. 05	13	0. 05	12				
s10	0. 04	100	0. 05	13				
s11	0. 03	101						
s12	0. 00	102						
s13	0. 00	103						

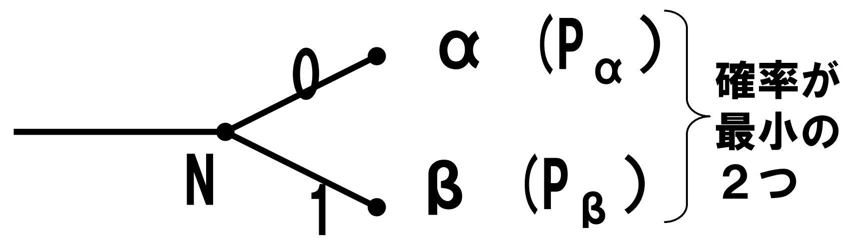
４元では縮退情報源を作る毎に３つづつ減る（4−1=3）

ダミーシンボル：s11, s12
（起き得ない確率は 0 に設定）

5.3 ハフマン符号のコンパクト性証明(1/4)

（補助定理）

- コンパクト符号の符号木において、根から最も遠い位置にある葉（最高次の葉）は、少なくとも2枚あり、最後の節点で分岐した2つの枝の先にある。この2枚の葉は、確率の最も小さい2つの情報源記号に対応している。



（補助定理の証明）

（仮定）

- 右図において、情報源記号のαとβの確率はP_α、P_βとし、$P_\alpha \leqq P_\beta$とする。

5.3 ハフマン符号のコンパクト性証明(2/4)

- 必ず最後の節点で分岐がある証明
  - もし最後の節点Nで分岐していない（即ち最後の節点Nから出る枝が1本である）場合、Nから出る枝を除きNを葉として情報源記号αに対応させても瞬時復号可能性は失われず平均符号長は短くなる。従って節点Nでは必ず分岐している。
- 分岐する葉が確率最小の記号に対応する証明
  - もし節点Nから分岐する葉より低次の（即ち枝の長さが短い）葉に確率P_βより小さい確率をもつ記号γが割り当てられていれば、βとγを入れ替えて平均符号長を短くできる。しかしこれはコンパクト性の仮定に反するのでそのようなγは存在せず、αとβが確率最小の２つの記号に対応する。　【証明終】

5.3 ハフマン符号のコンパクト性証明(3/4)

［定理］ハフマン符号はコンパクト符号である

［証明］

- ハフマン符号の木において、第 j 番目の段階の縮退情報源をS_jとする。もしS_jがコンパクトな符号割り当てならば1つ前のS_{j-1}もコンパクトであることを証明すればよい。
- S_{j-1}の平均符号長をL_{j-1}とする。 S_{j-1}の確率最小の2枚の葉の確率を、P_α、P_βとすると、

  $$L_{j-1}=L_j+P_\alpha+P_\beta$$

  が成り立つ。何故ならば、L_{j-1}ではこの２つの葉に0, 1がそれぞれ余計に付加されるので、L_jはこの1ビット分平均符号長が長くなるから。
- ここで、S_jがコンパクトであるが、S_{j-1}がコンパクトでなかったと仮定する。
- すると、別のコンパクト符号木S_{j-1}' が存在する。それに対する平均符号長L_{j-1}'とすると、 $L_{j-1}' < L_{j-1}$が成り立つ。

5.3 ハフマン符号のコンパクト性証明(4/4)

- 補助定理により、 S_{j-1}' には確率最小の2枚の葉が存在し、それらをまとめた新しい符号木S_j'を作ってみる。
- この木は、 S_jと同じ葉をもち、

  $L_{j-1}'=L_j'+P_\alpha+P_\beta<L_{j-1}= L_j+P_\alpha+P_\beta$
- となる。しかしこれは、S_jがコンパクトであるという仮定に矛盾する。従って、S_{j-1}もコンパクトでなければならない。　　　　【証明終】

5.3 ハフマン符号の平均符号長

5. 1節で定義した「平均符号長」を計算する：

- 最初のハフマン符号（A）の平均符号長

  L=1*0. 4+2*0. 3+3*0. 1+4*0. 1+5*0. 06+5*0. 04=2. 2

- 別のハフマン符号（B）の平均符号長

  L=1*0. 4+2*03+4*0. 1+4*0. 1+4*0. 06+4*0. 04=2. 2

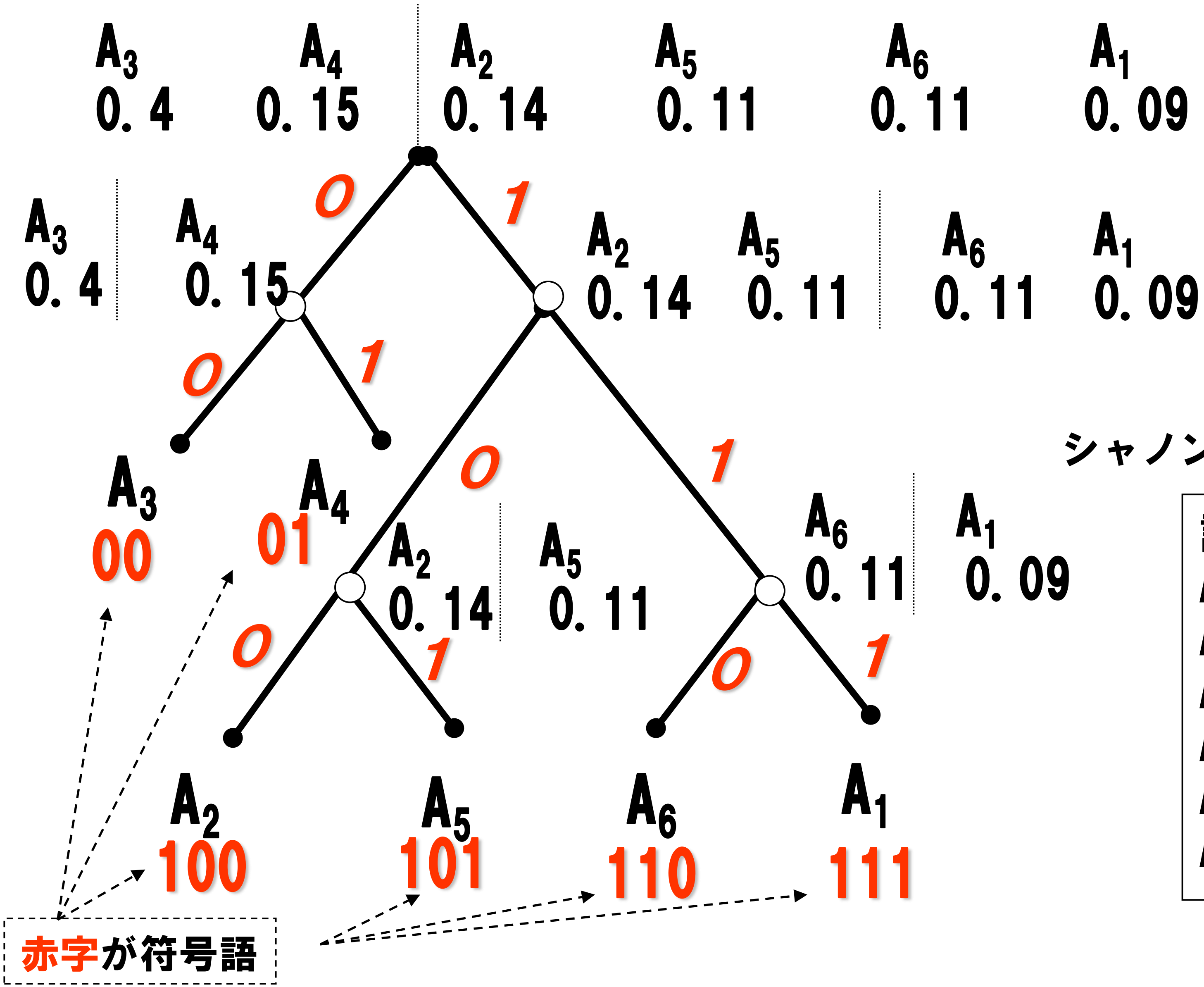
どちらも同じ。つまり作り方に自由度があっても、最終的にできたハフマン符号は同じ平均符号長になる。

複数個の記号をまとめてハフマン符号化すると、どうなるか
＝＞これを<u>拡大情報源の符号化</u>と言う

5.4 他の符号化：シャノン-ファノ符号

- ハフマン符号に似た高能率の符号化法
- 符号化のアルゴリズム（２元符号の場合）
  ①情報源シンボルを生起確率の高い順に並べ替える
  ②並べたシンボルの列を、それぞれの確率の和がほぼ等しくなるよう２分する
  ③２分した部分集合ごとに、一様に記号０または１を割り当てる
  ④上記の分割した集合をさらに、確率の和が均等になるように２分割し、それぞれの集合に記号０または１を割り当てる
  ⑤上記の手順を、もうそれ以上分割できないところまで繰り返す

5.4 シャノン-ファノ符号化ツリー（例）



シャノンーファノ符号

記号	符号語
A3	00
A4	01
A2	100
A5	101
A6	110
A1	111

5.4 シャノン-ファノ符号化の効率

- 情報源Aのエントロピ－H（A)

（前ページのツリー図に示す順に並べて計算すると）

H(A)=-0.4 log0.4-0.15 log0.15-0.14 log 0.14
- 0.11 log0.11-0.11 log 0.11-0.09 log0.09=2.35

- シャノン－ファノ符号化の平均符号長L

（符号語より確率×符号語長を総和：A3, A4, A2, A5, A6, A1の順

L=0.4*2+0.15*2+0.14*3+2*0.11*3
+0.09*3=2.45

- 効率η

η＝H(A)/L=2.35/2.45=0.96

(96%で1.0=100%, に達しておらず，完全なコンパクト符号化ではない）

5.4 シャノン-ファノ符号化：別の表現 (1/2)

符号化アルゴリズム

1. 情報源シンボルを確率の高いものから順番に並べる。いま、$s_1, s_2, s_3, ..., s_q$の順となったとする。
2. このとき、s_iに、

   L_i=ceiling (log$_2$ (1/p_i))

   なる符号長を割り当てる。ceiling (x)は、x以上の最小整数を表わす。即ち、x≦ceiling(x)<x+1を満たす。

   例：ceiling (1.563)=2, ceiling(1)=1, ceiling(-2.3)=-2
3. 次に、$p_1, p_2, p_3, ..., p_q$から、$\alpha_1, \alpha_2, \alpha_3, ..., \alpha_q$を計算する。
   - $\alpha_1=0$
   - $\alpha_2=p_1$
   - $\alpha_3=p_1+p_2=\alpha_1+p_2$
   - $\alpha_4=p_1+p_2+p_3=\alpha_3+p_3$
   - .....
   - $\alpha_{i+1}=\alpha_i+p_i$

ceiling (x)： 天井関数(ceiling function)
floor (x)： 床関数(floor function)
反対に、実数xに対してx以下の最大の整数として定義される関数。

5.4 シャノン・ファノ符号化：別の表現(2/2)

4. α_i を2進少数に展開して、小数点以下のL_i桁を情報源シンボルs_iに対応する符号語にする

- この方法で作られる符号は、$L < H(S)+1$を満たす
  - $\log_2 (1/p_i) \leqq L_i < \log_2 (1/p_i) +1$

    p_iを乗じて和をとると（和は1から、情報源シンボル数=q、まで）
  - $\Sigma p_i \log_2 (1/p_i) \leqq \Sigma p_i L_i < \Sigma p_i (\log_2 (1/p_i) +1)$
  - $H(S) \leqq \Sigma p_i L_i < H(S)+1$
- また、s_iに対する符号語とs_{i+1}に対する符号は、少なくとも最初のL_i桁のいずれかが異なるので、瞬時復号可能符号である

5.4 シャノン・ファノ符号化：例への適用

- 情報源Aが、下記のように、生起確率の順に並び替えて、A’として与えられているとする

$$A' = \begin{pmatrix} A_3 & A_4 & A_2 & A_5 & A_6 & A_1 \\ 0.4 & 0.15 & 0.14 & 0.11 & 0.11 & 0.09 \end{pmatrix}$$

- 符号語の長さL_iは、次のように決まる
  - $L_1=2$ $(\log_2(1/0.4)=1.32...)$
  - $L_2=3$ $(\log_2(1/0.15)=2.73...)$
  - $L_3=3$ $(\log_2(1/0.14)=2.83...)$
  - $L_4=4$ $(\log_2(1/0.11)=3.18...)$
  - $L_5=4$ $(\log_2(1/0.11)=3.18...)$
  - $L_6=4$ $(\log_2(1/0.09)=3.47...)$
- 各p_iからα_iを計算し、さらに2進小数展開する
  - $\alpha 1=0$ →0.00
  - $\alpha 2=0.4$ →0.011...
  - $\alpha 3=0.55$ →0.1000...
  - $\alpha 4=0.69$ →0.1011...
  - $\alpha 5=0.8$ →0.1100...
  - $\alpha 6=0.91$ →0.1110...

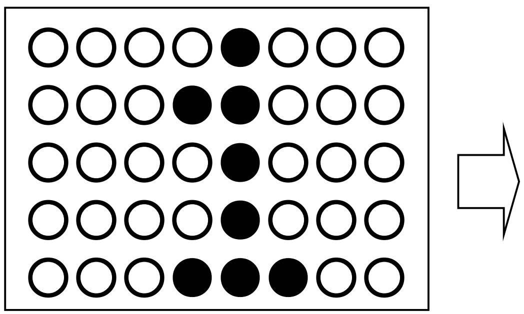
符号語

A3→00
A4→011
A2→100
A5→1011
A6→1100
A1→1110

平均符号長L=2.91、となりコンパクト符号と比較して効率悪いが、H(S)+1、より小さくなる

5.5 ランレングス符号化(1)

- 2元無記憶情報源S={0, 1}で、0と1の生起確率のいずれかが、ほぼ1に等しく、他方は0に近い場合。例えば、0の生起確率が1に近い場合、出力系列は圧倒的に0が多い：



00001000
00011000
00001000
00001000
00011100

- 0000100000011000000001000　（例：24ビット）

$\underbrace{0000}_{4}1\underbrace{000000}_{6}1\underbrace{}_{0}1\underbrace{0000000}_{7}1\underbrace{000}_{3}$

- 0の長さのみを示すと、上記のように，4, 6, 0, 7, 3, となる（注）長さ0は「11」と続く場合で「0」がゼロ長であることを意味する

5.5 ランレングス符号化(2)

- 符号化法
  - 連続する0（連(run)という）の長さを2進数字で表わせば大幅に系列の長さを短縮できる。この原理で符号化した符号を「ランレングス符号」という。上記例の系列に対しては、
    - 0000→100（=4)；0000000→110（=6)；11→0（=0)；0000000→111（=7)；000→011（=3)

    と次のように符号化できる.

    100　110　0　111　011　　　（=13ビットに圧縮）

  - しかし，実際は上のような空白はないので区切りがわからず，受信側で

    10 0 11 0 0 111・・・

    と解釈されると，連長が，2，0，3，0，0，7，. . . と誤って復元される可能性がある.

5.5 ランレングス符号化(3)，Wyle符号

- これを解決するため区切りを入れた符号化がワイル符号
- ワイル（Wyle）符号の符号化法
  - 長さnのランレングスの場合，まず1を引く（0は0）
  - n-1を2進数に直したのち，Wyle符号（下表）に変換する

ランレングス（連長）	符号	符号長
1～4	0**	3
5～8	10***	5
9～16	110****	7
17～32	1110*****	9
33～64	11110******	11
65～128	111110*******	13

5.5 ランレングス符号化(4)，Wyle符号

- Wyle符号化の例
  - ランレングス4の場合
    - 1を引いた数=3，を2進数に直し11，Wyle符号「0**」の「**」に代入して011となる
  - ランの長さが，4，6，1，7，3の場合は下表のとおり

元のランレングス（n）	4	6	1	7	3
n−1	3	5	0	6	2
2進数	11	101	00	110	10
Wyle符号	011	10101	000	10110	010
最終的な符号化系列	01110101000101100010				

5.6 拡大情報源の符号化（ブロック符号化）

情報源
（さいころ）

→ 出現系列(例)

s1, s2, s1, s1, . . .

上記で、s1は表、s2は裏を表すと仮定する

元情報源（Ｓ）の符号化　　１記号ずつ符号化

s1, s2, s1, s1, s2, s2, . . . → 010011. . .

二次拡大情報源（S^2）の符号化　　２記号ずつ符号化

s1s2, s1s1, s2s2, . . . → 010011. . .

三次拡大情報源（S^3）の符号化　　３記号ずつ符号化

s1s2s1, s1s2s2, . . . → 010001. . .

5.7 拡大情報源のハフマン符号化：２次拡大

- 元の情報源（簡単のために２元情報源）

記号	確率	符号
s1	3/4	*0*
s2	1/4	*1*

L＝<u>1/</u>記号

- ２次拡大情報源

記号	確率	符号				
s1s1	9/16	*0*	9/16	*0*	9/16	*0*
s1s2	3/16	*11*	4/16	*10*	7/16	*1*
s2s1	3/16	*100*	3/16	*11*		
s2s2	1/16	*101*				

赤字が符号語

L＝1★(9/16)+2★(3/16)+3★(3/16)+3★(1/16)=27/16＝1.688=<u>0.844/</u>記号

5.7 拡大情報源のハフマン符号化：３次拡大

・３次拡大情報源

S	確率	符号語
s1s1s1	27/64	1
s1s1s2	9/64	001
s1s2s1	9/64	010
s2s1s1	9/64	011
s1s2s2	3/64	00000
s2s1s2	3/64	00001
s2s2s1	3/64	00010
s2s2s2	1/64	00011

L＝1★27/64+3★9/64X3+5★3/64X3+5★1/64=158/64=2.468=<u>0.823</u>/記号

5.8 符号化の限界

- 平均符号長はどこまで短く出来るか？
  （符号化の限界はどこか？）
  例：Ｌ＝１→0.844→0.823→・・・？

- 結論としてはシャノンの情報源符号化定理で示された：具体的には下記の内容
  「情報源の平均符号長はそのエントロピーの値より小さくはならない。このエントロピーにいくらでも近い符号（最良な符号）を構成することができる。」
- 上の結論にたどり着くため、
  - Ｌの下限（最小値）がどうなるか？
  - Ｌの上限（最大値）がどうなるか？
  - これらの限界を満たす符号の作り方はあるか？

  について調べる

5.9 平均符号長の下限（最小値）

- 無記憶情報源Ｓが、

$$S = \begin{pmatrix} s_1 & s_2 & s_3 \cdots s_q \\ p_1 & p_2 & p_3 \cdots p_q \end{pmatrix}$$

で与えられるとき、エントロピーは

$H(S) = -\sum_i p_i \log_r p_i$ で与えられる。

- **平均符号長の下限に関する定理**

この情報源をｒ元符号に符号化したときの平均符号長Ｌの下限はＨ（Ｓ）となる。すなわち、

$H_2(S)/\log_2 r \leqq L$	($H_2(S)$は2元単位のエントロピー)
$H_r(S) \leqq L$	($H_r(S)$はr元単位のエントロピー)

5.10 符号長の下限定理：例とその意味(1)

- 例1：情報源S＝{s_1、s_2},$p(s_1)=3/4$, $p(s_2)=1/4$

$$H_2(S)= -3/4 \log_2 3/4 -1/4 \log_2 1/4= 0.811 \text{ (bit)}$$

- この情報源の2元符号は、平均符号長が0．811以上でなければならない。前例の符号では、L=1（1次拡大）、0.844（２次拡大）、0.823（3次拡大）、などが得られた。しかしH(S)に等しい符号は示されていない。

5.10 符号長の下限定理：例とその意味(2)

- 例2：情報源$S=\{s_1,s_2,s_3,s_4\}$, $p(s_1)=1/2$, $p(s_2)=1/4$, $p(s_3)=1/8$, $p(s_4)=1/8$

  $H_2(S)= -(1/2)\log(1/2)-(1/4)\log(1/4)-((1/8)\log(1/8))*2=1.75$ (bit)である。

これに対するハフマン符号は下記の通り

s1	1/2	0	1/2	0	1/2	0
s2	1/4	10	1/4	10	1/2	1
s3	1/8	110	1/4	11		
s4	1/8	111				

この平均符号長は下記のようになる：

$L=1*(1/2)+2*(1/4)+3*(1/8)+3*(1/8)=1.75$

従って、下限を満たしている符号となっている

5.11 符号長の下限と上限

- $H_r(S) \leqq L$の等号を満たす符号（最も効率の良い符号）は、情報源の記号の生起確率が$(1/r)^{L_i}$の形をしている必要がある
  - 例2の情報源がその例（r=2の２元符号）
- 各符号長$L_i = \log_r(1/P_i)$となる

【符号長の上限定理】

$\log_r(1/P_i) \leqq L_i < \log_r(1/P_i)+1$を満たす符号を作ると、

$H_r(S) \leqq L < H_r(S) + 1$を満たす

下限より高々１大きい平均符号長の非常に効率のよい符号が構成可能である

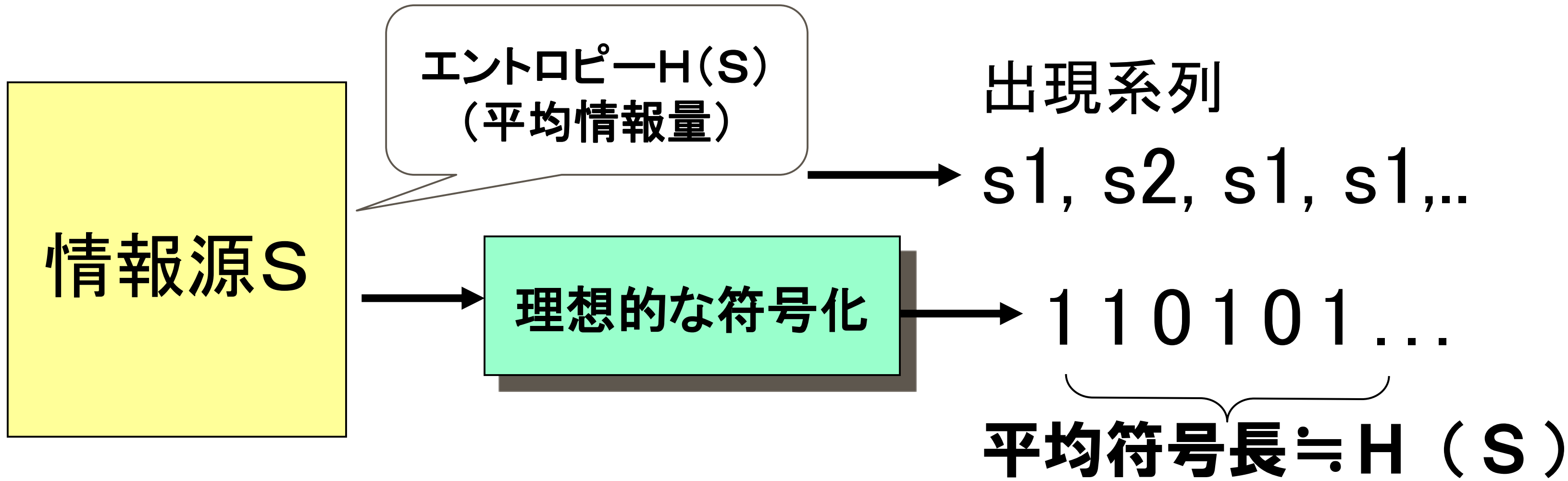
5.12 シャノン第1定理：情報源符号化定理

- $H_r(S) \leqq L < H_r(S) + 1$を、n次拡大情報源に適用すると、$H_r(S^n) \leqq L_n < H_r(S^n)+1$。無記憶情報源の場合、$H_r(S^n)=n\,H_r(S)$となる
- $H_r(S) \leqq L_n/n < H_r(S) + (1/n)$
  - nを大きくするとL_n/nが$H_r(S)$に近づく
  - $\lim_{n\to\infty} L_n/n = H_r(S)$

これより、次のことが成り立つ

- 情報源Sは、エントロピー$H_r(S)$にいくらでも近い平均符号長の符号に符号化できる

5.12 シャノン第1定理：その意味



- 情報源の情報量は、それを符号化するのに必要な2元シンボル数（即ち、平均符号長）に等しい。
- 言い換えると、情報源の量はそれを最適に符号化したときの符号の長さに反映される。情報源がもつ量はそれ以上符号長を小さくできない限界を決める量である。

5.13 符号の効率と冗長度

- 符号の良さを測る尺度の定義として、符号効率(又は冗長度)がある:
- 符号効率 η （イータ）

  ＝理想的符号の長さ/その符号長＝H(S)/L

  $0 \leqq \eta \leqq 1$
- 冗長度d

  $= 1 - \eta = (L - H(S))/L$
- 具体例(前例の情報源S)
  - １記号ずつの符号化：η=H(S)/L=0.811, d=0.189
  - ２記号ずつの符号化：η=0.811/0.844=0.961, d=0.039
  - ３記号ずつの符号化：η=0.811/0.823=0.985, d=0.015

  さらに、まとめることで η は１に近づく