情 報 技 術 基 礎

2011.5.14 改訂

弘前大学　小山智史

氏　　名：　　　　　　　　　　　　　　（　　　　　@stu.hirosaki-u.ac.jp）

第1章　情報の表現

1.1　集合

1.1.1　集合の表記

集合 (set) は**要素** (element) の集まりのことで、

$$\text{集合名} = \{\ \text{要素}, \text{要素}, ..., \text{要素}\ \} \tag{1.1}$$

のように表します。A という名前の集合が 1, 2, 3, 4, 5 の 5 つの要素からなる場合、

$$A = \{1, 2, 3, 4, 5\}$$

と書きます。また、ある要素が集合に属しているとき、$\in$ の記号を用いて

$$3 \in A$$

と書きます。ここで、次の集合 B を考えます。

$$B = \{2, 3, 4\}$$

この例のように、B のすべての要素が A に属しているとき、B は A の**部分集合** (subset) であるといい、$\subset$ の記号を用いて

$$B \subset A$$

と書きます。あらゆる集合は、それ自体の部分集合になっています。

要素のない集合を**空集合**といい、ϕ という記号で表します。

$$\phi = \{\ \}$$

ϕ は空集合以外のあらゆる集合の部分集合になっています。

全体集合を U とし、今

$$U = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

とします。このとき、A の**補集合** A^c は

$$A^c = \{x | x \notin A \text{ かつ } x \in U\} = \{0, 6, 7, 8, 9\}$$

となります。

1.1.2　集合の演算

ここで、次の集合 C を考えます。

$$C = \{0, 1, 2, 3\}$$

和集合 $A \cup C$ は A, C のどちらかに属している要素の集合です。

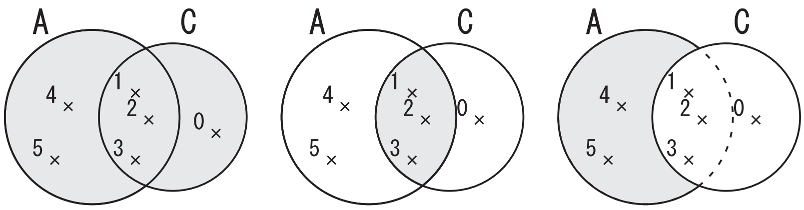
$$A \cup C = \{x | x \in A \text{ または } x \in C\} = \{0, 1, 2, 3, 4, 5\}$$

積集合 $A \cap C$ は A, C のどちらにも属している要素の集合です。

$$A \cap C = \{x | x \in A \text{ かつ } x \in C\} = \{1, 2, 3\}$$

差集合 $A - C$ は A の要素から C の要素となっているものを除いた要素の集合です。

$$A - C = \{x | x \in A \text{ かつ } x \notin C\} = \{4, 5\}$$



(a)$A \cup C$　(b)$A \cap C$　(c)$A - C$

図 1.1: 集合の演算とベン図

（練習） 前出の例で、集合 A と集合 B の関係をベン図で示しなさい。

プログラム　set.htm

以下のプログラムは

A について「grape banana lemon」
B について「banana orange」

のように集合の要素を「半角または全角のカンマまたはスペース」で区切って入力すると演算結果を表示します。

```
<script>
A=new Array(); a=prompt("A=","").split(/[,，　 ]/); // a[0]="grape", ...
for(i=0;i<a.length;i++) A[a[i]]=""; // A["grape"]="", ...
B=new Array(); b=prompt("B=","").split(/[,，　 ]/); // b[0]="lemon", ...
for(i=0;i<b.length;i++) B[b[i]]=""; // B["lemon"]="", ...

with(document){
    write("<p>A={"); for(x in A) write(x+","); write("}</p>");
    write("<p>B={"); for(x in B) write(x+","); write("}</p>");
    write("<p>A ∩ B={"); for(x in A)if(x in B) write(x+","); write("}</p>");
    write("<p>A ∪ B={");
    for(x in A) write(x+",");
    for(x in B) if(!(x in A)) write(x+",");
    write("}</p>");
    write("<p>A － B={");for(x in A)if(!(x in B)) write(x+","); write("}</p>");
    write("<p>B － A={");for(x in B)if(!(x in A)) write(x+","); write("}</p>");
}
</script>
```

1.2 2 進数による情報の表現

1.2.1 情報の単位

コンピュータの内部は電子回路で実現されていて、データは 2 値、すなわち「電圧が高いか低いか」のいずれであるかによって表現されています。ある信号線の電圧が高い場合を 2 進数字「1」、低い場合を 2 進数字「0」に対応づけると、その 1 本の信号線で 2 進数 1 桁を表わすことができます。

2 進数 1 桁では 2 つの中のいずれであるかを表すことができます。2 進数 2 桁では「00 01 10 11」の $2^2 = 4$ 種類のデータのいずれか、2 進数 8 桁では「00000000 〜 11111111」の $2^8 = 256$ 種類のデータのいずれかを表すことができます。この 2 進数の桁数の単位を **bit**(binary digit) と呼びます。bit はしばしば記憶容量 (入れ物の大きさ) の単位としても用いられます。2 bit のメモリは「00 01 10 11」のいずれかを記憶できます[1]。

では、2 進数で表したデータが何を意味するかとなると、例えば、「01000111」は、ある時は 256 種類の文字の中のひとつ「G」を意味し、ある時は 0〜255 までの数値の中のひとつ「71」を意味し、またある時は 256 種類の命令の中のひとつを意味します。このように、データは、**送る側と受ける側の間に共通の了解がある**ことが大きな前提となっていることに注意する必要があります。送る側は送りたい文字を符号 (bit パターン) に変換して送り、受ける側は受け取った符号を文字に復元します (図 1.2)。

このような了解を送信の度に取り決めるのは繁雑であるため、情報処理・情報交換用符号が規格として定められています。この符号としては、ASCII(American Standard Code for Information Interchange) コード、ISO(International Organization for Standardization) コード、JIS(Japanese Industrial Standards) コード、シフト JIS(S-JIS) コード、EBCDIC(Extended Binary Coded Decimal Interchange) コード、EUC(Extended Unix Code) コード、Unicode などが用いられます。

パソコンを利用していると「文字化け」に遭遇することがありますが、これは送り手と受け手の間の共通の了解が崩れているために生ずるものです。

コンピュータ内部では 8 bit 単位でデータが扱われることが多く、2 進 8 桁を **Byte**(または B) という単位で表します。$2^8 = 256$ 種類では、英数字を表現するには十分ですが、漢字を含む日本語文字を表現するには足りません。そこで、日本語の表現には通常 2 Byte(16 bit) 用いられます。2 Byte では $2^{16} = 65536$ 種類の中のどれかを表すことができます。

「メモリの容量が 1 Byte」ということは、256 種類のうちのどれかという情報を 1 つだけ記憶できるということです。同様に「1 Byte のデータを送る」ということは、256 種類のうちのどれかという情報を 1 つだけ送るということです。

コンピュータでは、Byte の他に **Word** という単位も用いられます。これは、コンピュータ内部でデータがどういう単位で処理されるかを表すもので、それが何 bit に相当するかは定まっていません。「16 bit コンピュータ」、「32 bit コンピュータ」などの言い方が用いられますが、これは、コンピュータ内部ではそれぞれ 1 Word = 16 bit(2 Byte), 1 Word = 32 bit(4 Byte) の単位で計算などの処理が行われることを示しています。

(練習) 6 bit で表現できるデータの個数は何個か。

[1] 4 章で示すように、**bit** は情報量の単位としても用いられます。記憶容量の bit 値 (2 進桁数) はそこに収容できる最大の情報量 (bit 値) となります。

表 1.1: JIS X 0201 コード表

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	NUL	DLE	SP	0	@	P	‘	p				ー	タ	ミ		
0001	SOH	DC1	!	1	A	Q	a	q			。	ア	チ	ム		
0010	STX	DC2	”	2	B	R	b	r			「	イ	ツ	メ		
0011	ETX	DC3	#	3	C	S	c	s			」	ウ	テ	モ		
0100	EOT	DC4	$	4	D	T	d	t			、	エ	ト	ヤ		
0101	ENQ	NAK	%	5	E	U	e	u			・	オ	ナ	ユ		
0110	ACK	SYN	&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
0111	BEL	ETB	’	7	G	W	g	w			ァ	キ	ヌ	ラ		
1000	BS	CAN	(	8	H	X	h	x			ィ	ク	ネ	リ		
1001	HT	EM	)	9	I	Y	i	y			ゥ	ケ	ノ	ル		
1010	LF	SUB	＊	：	J	Z	j	z			ェ	コ	ハ	レ		
1011	VT	ESC	＋	；	K	[	k	{			ォ	サ	ヒ	ロ		
1100	FF	FS	，	＜	L	¥	l	｜			ャ	シ	フ	ワ		
1101	CR	GS	－	＝	M	]	m	}			ュ	ス	ヘ	ン		
1110	SO	RS	．	＞	N	^	n	￣			ョ	セ	ホ	゛		
1111	SI	US	／	？	O	_	o	DEL			ッ	ソ	マ	゜		

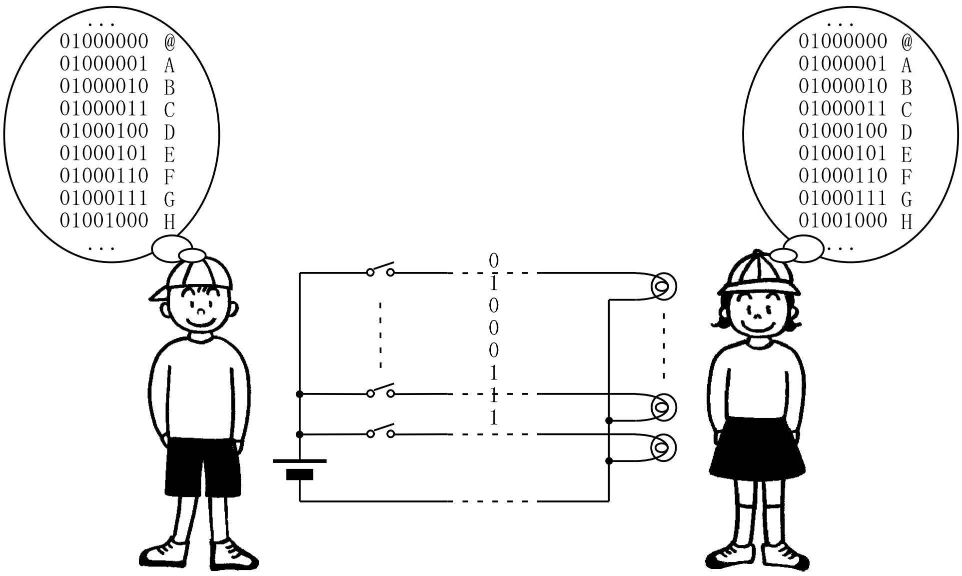


図 1.2: 符号と通信

1.2.2　2進数と16進数

電卓で「43＋17」の計算をする場合を考えましょう。図1.3に示すように、

$\boxed{4}\boxed{3}\boxed{+}\boxed{1}\boxed{7}\boxed{=}$

と操作すると、まず、キー操作で入力した「43」と「17」がそれぞれ2進数の「00101011」と「00010001」に変換され、次に2進の加算演算が行われ「00111100」が得られます。この結果が10進に変換さ

表 1.2: 2 進数と 16 進数

10 進数	2 進数	16 進数	10 進数	2 進数	16 進数	10 進数	2 進数	16 進数
0	00000000	0	16	00010000	10	32	00100000	20
1	00000001	1	17	00010001	11	33	00100001	21
2	00000010	2	18	00010010	12	34	00100010	22
3	00000011	3	19	00010011	13	35	00100011	23
4	00000100	4	20	00010100	14	36	00100100	24
5	00000101	5	21	00010101	15	37	00100101	25
6	00000110	6	22	00010110	16	38	00100110	26
7	00000111	7	23	00010111	17	39	00100111	27
8	00001000	8	24	00011000	18	40	00101000	28
9	00001001	9	25	00011001	19	41	00101001	29
10	00001010	A	26	00011010	1A	42	00101010	2A
11	00001011	B	27	00011011	1B	43	00101011	2B
12	00001100	C	28	00011100	1C	44	00101100	2C
13	00001101	D	29	00011101	1D	45	00101101	2D
14	00001110	E	30	00011110	1E	46	00101110	2E
15	00001111	F	31	00011111	1F	47	00101111	2F

れ「60」となり、ディスプレイに表示されます。

2 進数でデータを表すと桁数が多くなってしまうので、人間が扱うには不便です。そこで、しばしば 16 進数が用いられます。これは、2 進数を 4 桁ごとに区切って、4 bit(2 進 4 桁) を 16 進 1 桁で表す方法です (表 1.2)。表からわかるように、「0〜9，A〜F」の 16 種類の文字を用いて 16 進 1 桁を表します。

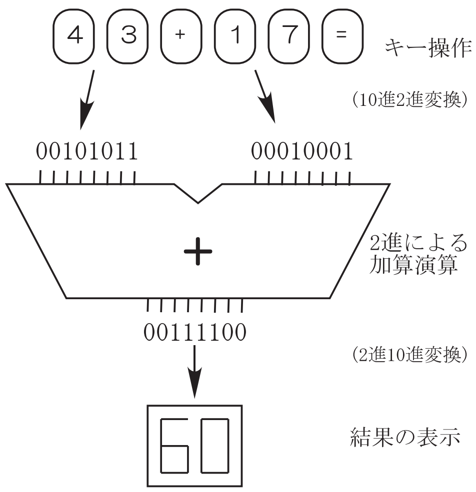


図 1.3: 電卓の操作・演算・表示

例えば、01000001 は、16 進では 41 となります。このように、2 進と 16 進の相互の変換は至って単純です。なお、2 進数の表記であることを明示するために 01000001B または $(01000001)_2$、16 進数の表記であることを明示するために 41H または $(41)_{16}$ のように書きます。

（練習） $(10111010001)_2$ を 16 進数で表しなさい。

（練習） $(11010110010)_2$ を 16 進数で表しなさい。

（練習） $(\mathtt{6B2})_{16}$ を2進数で表しなさい。

（練習） $(\mathtt{9F4})_{16}$ を2進数で表しなさい。

プログラム　ascii.htm

以下のプログラムはASCIIコードの可視文字をコードとともに表示します。

```
<script>
for(c=32;c<=126;c++) document.write(c.toString(16)+":"+String.fromCharCode(c)+"<br>");
</script>
```

1.2.3　接頭語

大きな値や小さな値を表す時には、「1 km」の「k(キロ: 10^3 の意味)」のように接頭語を用います。このような10進の接頭語は国際単位系 (SI) で決められています (表1.4)。

データの量 (bit や Byte) を表す場合に用いる2進の接頭語についても、「1 Ki Byte」の「Ki(キビ:$2^{10} = 1024$ の意味)」などが1998年にIEC(国際電気標準会議) で決められましたが (表1.4)、普及していません。実際には、10進の接頭語に準じた接頭語が「1 K(キロ) Byte」のように慣用的に用いられています。「K(キロ: $2^{10} = 1024$)」については、1000と区別するために大文字を用いることが慣習となっていますが、「M(メガ)」「G(ギガ)」「T(テラ)」の場合には明示的な区別はないので注意が必要です。

「64K Byte」は「64×2^{10}Byte $= 64\times 1024$Byte $= 65536$Byte」で、およそ64k Byte $= 64\times 10^3$Byte になっています。情報機器のカタログを注意深く見ると、

「ディスク容量の表記は1GB=10^9Byteを使用しています」

などの断り書きが見つけることができます。これによれば、1GB $= 2^{30}$ Byte $= 1024^3$ Byte $=$ 1073741824 Byte の表記で「60 GB」のハードディスクは、「64 GB」と表記されることになります。接頭語の解釈の違いから「実際のハードディスクの容量が広告記載の容量よりも少ない」として、訴訟が起こされた例もあります。

このようにハードディスクの容量には10進の接頭語が用いられる一方で、半導体メモリの容量の場合は「256 M Byteのメモリは 256×2^{20} Byte」のように2進の接頭語が用いられるのが慣習となっていて、混乱を来す場面もあります。

（練習） 64 MByteのメモリは正確には何Byteか。

（練習） 各自、自分のパソコンの内蔵メモリとハードディスクの正確なバイト数を計算しなさい。

1.3　数の表現

1.3.1　r 進数から10進数への変換

前節では、データを2進数字の並び (bit パターン) で表しました。ここでは、数の表現に着目します。

表 1.3: 接頭語

国際単位系 (SI)		2 進の接頭語		
記号	意味	慣用記号	IEC60027-2	意味
Y(ヨタ)	10^{24}			
Z(ゼタ)	10^{21}			
E(エクサ)	10^{18}	E(エクサ)	Ei(エクシビ)	$2^{60} = 1024^6 = 10^{18} \times 1.153$
P(ペタ)	10^{15}	P(ペタ)	Pi(ペビ)	$2^{50} = 1024^5 = 10^{15} \times 1.126$
T(テラ)	10^{12}	T(テラ)	Ti(テビ)	$2^{40} = 1024^4 = 10^{12} \times 1.100$
G(ギガ)	10^9	G(ギガ)	Gi(ギビ)	$2^{30} = 1024^3 = 10^9 \times 1.074$
M(メガ)	10^6	M(メガ)	Mi(メビ)	$2^{20} = 1024^2 = 10^6 \times 1.049$
k(キロ)	10^3	K(キロ)	Ki(キビ)	$2^{10} = 1024 = 10^3 \times 1.024$
h(ヘクト)	10^2			
da(デカ)	10			
d(デシ)	10^{-1}			
c(センチ)	10^{-2}			
m(ミリ)	10^{-3}			
μ(マイクロ)	10^{-6}			
n(ナノ)	10^{-9}			
p(ピコ)	10^{-12}			
f(フェムト)	10^{-15}			
a(アト)	10^{-18}			
z(ゼプト)	10^{-21}			
y(ヨクト)	10^{-24}			

我々は通常「**位取り記数法**」を用いて数を 0〜9 の 10 進数字の並びで表現します。「`29.37`」と書けば、10 の位が 2、1 の位が 9、0.1 の位が 3、0.01 の位が 7 であり、次の数を意味しています。

$$2 \times 10^1 + 9 \times 10^0 + 3 \times 10^{-1} + 7 \times 10^{-2}$$

一般に、$(c_m ... c_1 c_0 . c_{-1} ... c_{-n})_r$ と表す数の値は、次のようになります。

$$c_m \times r^m + ... + c_1 \times r^1 + c_0 \times r^0 + c_{-1} \times r^{-1} + ... + c_{-n} \times r^{-n}, c_i \in \{0, 1, ..., r-1\} \tag{1.2}$$

r は**基数** (radix) と呼ばれ、2 進では 2、10 進では 10、16 進では 16 です。

(1.2) 式を用いて、2 進数や 16 進数で表わされた数を 10 進数に変換することができます。例えば、$(101.01)_2$ が表す数は以下のように計算できます。

$$1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = 4 + 0 + 1 + 0 + .25 = 5.25$$

表**??**を用いる (または頭の中に入れておく) と

$$101.01 = 100 + 1 + .01 = 4 + 1 + .25 = 5.25$$

のように計算できます。

16 進数では、$(\mathtt{B7.F2})_{16}$ が表す数は以下のように計算できます。

$$11 \times 16^1 + 7 \times 16^0 + 15 \times 16^{-1} + 2 \times 16^{-2} = 176 + 7 + .9375 + .0078125 = 183.9453125$$

（練習） $(1101.01101)_2$ を 16 進数および 10 進数で表しなさい。

（練習） $(100.11011)_2$ を 16 進数および 10 進数で表しなさい。

（練習） $(1011101.0001101)_2$ を 16 進数および 10 進数で表しなさい。

表 1.4: 2 進数と 10 進数

10000000	128
01000000	64
00100000	32
00010000	16
00001000	8
00000100	4
00000010	2
00000001	1
00000000.1	0.5
00000000.01	0.25
00000000.001	0.125

（練習） $(\texttt{BED})_{16}$ を 10 進数で表しなさい。

（練習） $(\texttt{FACE})_{16}$ を 10 進数で表しなさい。

（練習） $(\texttt{9D6A})_{16}$ を 10 進数で表しなさい。

（練習） $(\texttt{2B8.A7})_{16}$ を 10 進数で表しなさい。

（練習） $(\texttt{1234})_{8}$ を 10 進数で表しなさい。

（練習） $(\texttt{777})_{8}$ を 10 進数で表しなさい。

1.3.2　10 進数から r 進数への変換

10 進数を r 進数に変換する場合は、整数部と小数部に分けて考えます。

まず整数部について考えます。(1.2) 式から、$(c_m...c_1c_0)_r$ を r で割ると、商が $(c_m...c_1)_r$(つまり元の数を 1 桁右にシフト)、余りが 1 桁目の値 c_0 となります。その商を r で割ると、商が $(c_m...c_2)_r$、余りが 2 桁目の値 c_1 となり、以下同様に求めることができます。

例えば、$(29)_{10}$ を 2 進数に変換する場合、29 を 2 で割った商が 14、余りが 1 で、$c_0 = 1$ となります。その商 14 を 2 で割った商が 7、余りが 0 で、$c_1 = 0$ となり、以下同様に 2 進数に変換することができます。

以上のプロセスを筆算として表すと以下のようになります。

$$
\begin{array}{cc|rl}
2 &) & 29 & \\
\hline
2 &) & 14 & \ldots\ 1 \\
\hline
2 &) & 7 & \ldots\ 0 \\
\hline
2 &) & 3 & \ldots\ 1 \\
\hline
2 &) & 1 & \ldots\ 1 \\
\hline
 & & 0 & \ldots\ 1
\end{array}
$$

従って

$$(29)_{10} = (\texttt{11101})_2$$

となります。

次に小数部について考えます。(1.2) 式から、$(.c_{-1}...c_{-n})_r$ に r を掛けると、整数部が c_{-1}、小数部が $(.c_{-2}...c_{-n})_r$ となります (つまり 1 桁左にシフト)。その小数部に r を掛けると、整数部が c_{-2}、小数部が $(.c_{-3}...c_{-n})_r$ となり、以下同様に求めることができます。

例えば、$(.4375)_{10}$ を 2 進数に変換する場合、.4375 に 2 を掛けた値は、整数部が 0、小数部が.875 で、$c_{-1} = 0$ となります。その小数部に 2 を掛けた値は、整数部が 1、小数部が.75 で、$c_{-2} = 1$ となり、以下同様に 2 進数に変換することができます。

以上のプロセスを筆算として表すと以下のようになります。

$$\begin{array}{ll} .4375 & \times 2 \\ 0.875 & \times 2 \\ 1.75 & \times 2 \\ 1.5 & \times 2 \\ 1.0 & \end{array}$$

従って

$$(.4375)_{10} = (.0111)_2$$

となります。

以上のように、$(29.4375)_{10}$ は $(11101.0111)_2$ のように 2 進数で表すことができます。小数に関しては、実際にはほとんどの場合が無限小数となるので、計算を途中で打ち切らざるを得ず、変換は誤差を伴ったものになります。

（練習） $(0.05)_{10}$ を 2 進数で表しなさい。ただし、小数点以下 8 桁までとしなさい。

（練習） $(252.3161)_{10}$ を 2 進数と 16 進数で表しなさい。

（練習） $(3.14159)_{10}$ を 2 進数と 16 進数で表しなさい。

（練習） $(1997.728)_{10}$ を 2 進数と 16 進数で表しなさい。

プログラム　dec2bin.htm

以下のプログラムは 10 進で数を入力すると 2 進に変換して表示します。

```
<script>
x=prompt("数?","")*1;
document.write("("+x+")D ... ("+bin(x,8)+")B");
function bin(x,n){
    if(x<0)       return bin(256+x,n);
    else if(n>0)  return bin(Math.floor(x/2),n-1)+""+x%2;
    else          return "";
}
</script>
```

1.4　2 進数の演算

2 進数の加減乗除の演算は加算のみで実現できます。コンピュータや電卓の内部の演算回路 (ALU) として加算器だけを用意すれば加減乗除ができ、ハードウェアの軽減につながっています。

1.4.1　2 進数の加減乗除

2 進 1 桁の加算は、以下の 4 つの場合があります。

0+0=0
0+1=1
1+0=1
1+1=0,　上位桁への「桁上がり (**carry**)」1

これを用いた複数桁の加算は以下のようになります (桁上がりに注意)。

$$
\begin{array}{rrcr}
 & \texttt{01001101} & \ldots & (77)_{10} \\
+ & \texttt{01100110} & \ldots & (102)_{10} \\
\hline
 & \texttt{10110011} & \ldots & (179)_{10}
\end{array}
$$

実際、このような加算の演算を実現するために、コンピュータの中には図 1.4 のような加算器が内蔵されています。つまり、8bit の加算を行うためには、1bit の加算器 8 個が桁上がりを考慮して直列に接続されるのです。

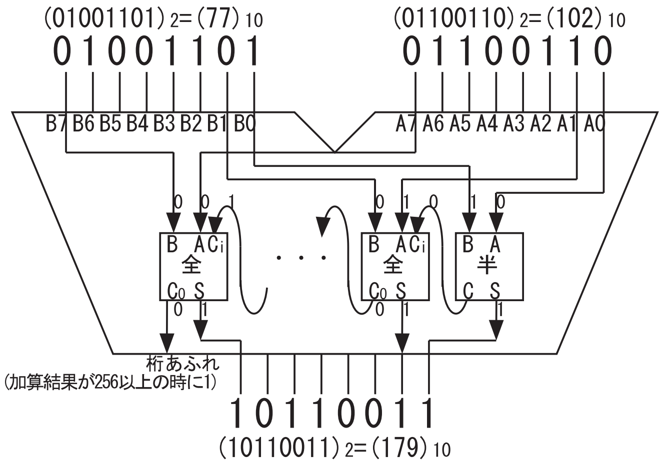


図 1.4: 8bit 加算器の内部

2 進 1 桁の減算は、以下の 4 つの場合があります。

0-0=0
0-1=1,　上位桁から「桁借り (**borrow**)」1
1-0=1
1-1=0

これを用いた複数桁の減算は以下のようになります (桁借りに注意)。

$$
\begin{array}{rrcr}
 & \texttt{00101011} & \ldots & (43)_{10} \\
- & \texttt{00010001} & \ldots & (17)_{10} \\
\hline
 & \texttt{00011010} & \ldots & (26)_{10}
\end{array}
$$

ただし、コンピュータの中では、多くの場合、次節に示すような数の表現方法の工夫により、減算も加算器を用いて行われます。

2 進 1 桁の乗算は、以下の 4 つの場合があります。

$$
\begin{array}{c}
0 \times 0 = 0 \\
0 \times 1 = 0 \\
1 \times 0 = 0 \\
1 \times 1 = 1
\end{array}
$$

これを用いた複数桁の乗算は以下のようになります。

```
    1101   ...  (13)10
×    101   ...   (5)10
--------------------
    1101        (65)10
  1101
---------
 1000001
```

このように、乗算は被乗数を乗数の 1 の箇所まで桁をずらして (シフトして)、それらを加えます。つまり、乗算はシフトと加算で実現できます。

除算も 10 進数の筆算と同様の方法で行うことができます。

```
        10001          17
101) 1010111     5)  87
     101              5
     -------         --
       00111          37
         101          35
     -------         --
          10           2
```

（練習）$(\texttt{01010101})_2 + (\texttt{01100110})_2$ を計算しなさい。ただし、2 進数のままでの計算と、16 進数に直してからの計算の両方を試みなさい。また、結果を 10 進数で表しなさい。

（練習）$(\texttt{1.1011})_2 + (\texttt{1.1101})_2$ を計算しなさい。ただし、2 進数のままでの計算と、16 進数に直してからの計算の両方を試みなさい。また、結果を 10 進数で表しなさい。

（練習）$(\texttt{DD})_{16} + (\texttt{1F})_{16}$ を計算しなさい。ただし、16 進数のままでの計算と、2 進数に直してからの計算の両方を試みなさい。また、結果を 10 進数で表しなさい。

（練習）$(\texttt{25})_{10} + (\texttt{37})_{10}$ の計算しなさい。ただし、被演算数を一旦 2 進数に直してから計算し、その結果を 10 進数に直しなさい。

1.4.2 負の数の表現

正の数だけを扱えばよい場合は、4bit で 0〜15 の 16 個の数を表すことができます。

2 進数字列	値
0000	0
0001	1
0010	2
...	
1111	15

負の数を表現する方法には、符号 bit を用いる方法もありますが、コンピュータでは、多くの場合**2 の補数**により負の数を表現します。

ある定められた数に対していくつ足りないかを表す数を補数といいます。r 進数の場合は $r-1$ の補数と r の補数があります。

はじめに 10 進数の補数である「9 の補数」と「10 の補数」について考えてみます。「9 の補数」は各桁の数値を 9 から引いたもので、この演算は桁借りなしで各桁ごとに行うことができます。「10 の補数」は「9 の補数」に 1 を加えたものです。例えば、10 進 3 桁で数を表す場合、$(\texttt{17})_{10}$ の「9 の補数」は次のようになります。

$$\begin{array}{rr} & 999 \\ - & 17 \\ \hline & 982 \end{array}$$

「10 の補数」はこれに 1 を加えた 983 です。これは、10 進 3 桁から桁あふれを起こした 1000 から 17 を引いた $1000 - 17 = 983$ という値です。このように -17 という数を符号を用いずに 983 で表すのが 10 の補数の表現方法です。最上位桁が 0〜4 の場合を正、5〜9 の場合を負とすれば、以下のように、10 進 3 桁で -500〜499 の 1000 個の数を表すことができます。

10 の補数表現	値
000	0
001	1
…	
499	499
500	-500
501	-499
…	
957	-43
…	
974	-26
…	
983	-17
…	
998	-2
999	-1

「10 の補数」を用いる利点は「減算を加算で行うことができる」という点にあります。

例えば、「$43 - 17$」は「$43 + (-17)$」であり、以下のように -17 の 10 の補数表現 983 と加算を行い、桁あふれを捨て、26 が得られます。

$$
\begin{aligned}
43 - 17 &= 43 + (1000 - 17) - 1000 \\
&= 43 + 983 - 1000 \qquad \leftarrow \text{10 の補数と加算し} \\
&= 1026 - 1000 \qquad \leftarrow \text{桁あふれを捨てる} \\
&= 26
\end{aligned}
$$

「$17 - 43$」は「$17 + (-43)$」であり、以下のように -43 の 10 の補数表現 957 と加算を行い、桁あふれを捨て、-26(の補数表現) が得られます。

$$
\begin{aligned}
17 - 43 &= 17 + (1000 - 43) - 1000 \\
&= 17 + 957 - 1000 \qquad \leftarrow \text{10 の補数と加算し} \\
&= 974 - 1000 \\
&\quad (-26 \text{ の補数表現})
\end{aligned}
$$

次に 2 進数の補数である「1 の補数」と「2 の補数」について考えてみます。「1 の補数」は各桁の数を 1 から引いたものですが、これは各桁を反転したものに他なりません。「2 の補数」は「1 の補数」に 1 を加えたものです。例えば、2 進 8 桁で数を表わす場合、$(00010001)_2$ の「1 の補数」は次のようになります。

$$
\begin{array}{r}
11111111 \\
-00010001 \\
\hline
11101110
\end{array}
$$

「2 の補数」はこれに 1 を加えた (11101111) です。これは 2 進 8 桁から桁あふれを起こした $(100000000)_2$ から $(00010001)_2$ を引いた $(100000000)_2 - (00010001)_2 = (11101111)_2$ という値で

す。このように $(-00010001)_2$ という数を符号を用いずに $(11101111)_2$ で表すのが 2 の補数の表現方法です。最上位桁が 0 の場合が正、1 の場合が負とすれば、以下のように、2 進 8 桁で -128〜127 の 256 個の数を表すことができます。

2 の補数表現	値
00000000	0
00000001	1
…	
01111111	127
10000000	-128
…	
11111111	-1

「2 の補数」を用いる利点は、「減算を加算で行うことができる」という点にあります。

例えば、「$43 - 17$」の減算は、コンピュータや電卓の内部では次のような 2 進の演算として行われます。17 を 2 進数にした 00010001 の bit を反転させ 1 を加えると 11101111 となり、これは -17 の「2 の補数表現」になっています。43 を 2 進数にした 00101011 にこれを加え、8 桁からあふれた 9 桁目は捨ててしまいます。

$$
\begin{array}{rr}
 & \texttt{00101011} \\
+ & \texttt{11101111} \\
\hline
 & \texttt{00011010}
\end{array}
\qquad
\begin{array}{rr}
 & 43 \\
+ & (-17) \\
\hline
 & 26
\end{array}
$$

（練習）-43 を 2 の補数で表しなさい。また、$17 - 43$ を 2 進数で計算しなさい。

（練習）-21 を 2 の補数で表しなさい。また、$64 - 21$ を 2 進数で計算しなさい。

プログラム add.htm

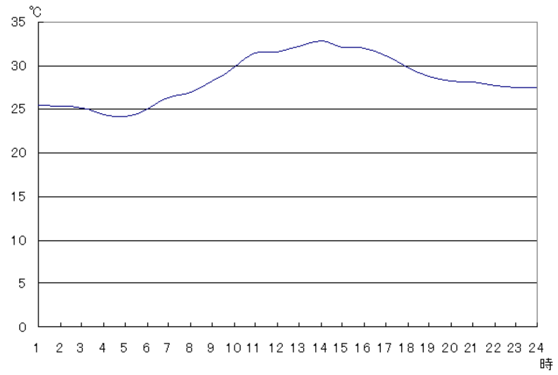
以下のプログラムは、10 進で入力した 2 つの数を 2 進に変換し、加算される様子を表示します。負の数を入力すると 2 の補数の演算の様子がわかります。

```
<script>
x=prompt("x=","")*1;
y=prompt("y=","")*1;
with(document){
    write("<table><tr><td align=right>");
    write("("+x+")D ("+bin(x,8)+")B<br>");
    write("("+y+")D ("+bin(y,8)+")B<br>");
    write("-------------------<br>");
    write("("+(x+y)+")D ("+bin(x+y,8)+")B<br>");
    write("</td></tr></table>");
}
function bin(x,n){
    if(x<0)      return bin(256+x,n);
    else if(n>0) return bin(Math.floor(x/2),n-1)+""+x%2;
    else         return "";
}
</script>
```

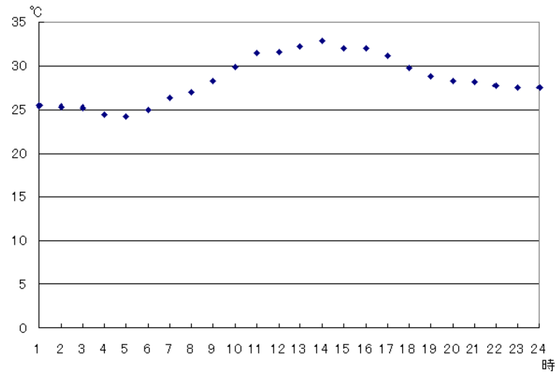
第2章　アナログとデジタル

2.1　標本化と量子化

図 2.1(a) はある日の気温の変化を表したものです。これは、時々刻々値が**連続的**に変化する**アナログ**情報です。
私達が気温を測定する場合、一定時間毎に有限精度で観測し、記録します。私達がごく普通に行うこのことがアナログ信号のデジタル化に他なりません。

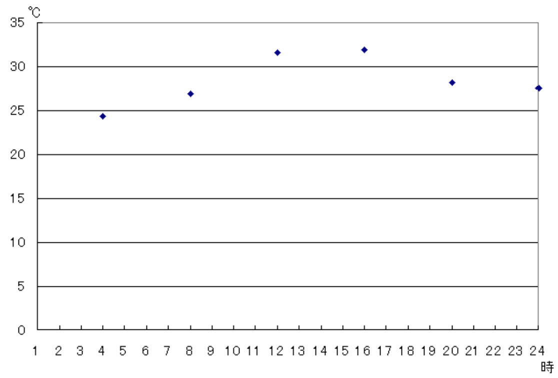


(a) 気温の変化



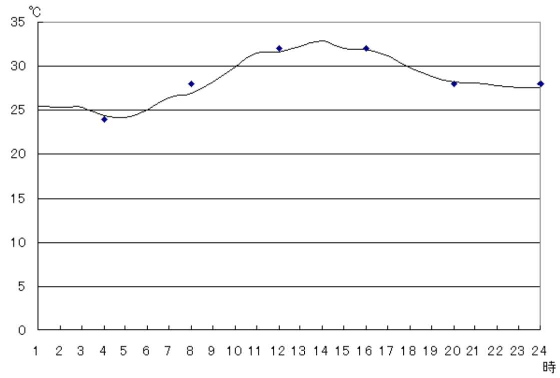
時	1	2	3	4	5	6
℃	25.4	25.3	25.2	24.4	24.2	25.0
時	7	8	9	10	11	12
℃	26.3	26.9	28.2	29.9	31.4	31.5
時	13	14	15	16	17	18
℃	32.2	32.8	32.0	31.9	31.1	29.8
時	19	20	21	22	23	24
℃	28.8	28.2	28.1	27.7	27.5	27.5

(b) サンプリング周期 1 時間 量子化レベル 0.1 ℃



時	4	8	12	16	20	24
℃	24.4	26.9	31.5	31.9	28.2	27.5

(c) サンプリング周期 4 時間 量子化レベル 0.1 ℃



時	4	8	12	16	20	24
℃	24	26	32	32	28	28

(d) サンプリング周期 4 時間 量子化レベル 4 ℃

図 2.1: 気温の標本化と量子化

図 2.1(b) はこの気温を 1 時間毎に 0.1 ℃の精度で測定したもので、これは**離散的**な**デジタル**情報です。

一定時間毎にデータを採取することを**標本化 (サンプリング)** と呼び、標本化の時間間隔を**サンプリング周期**) といいます。サンプリング周期の逆数 (1 秒間にサンプリングする回数) を**サンプリング周波数**といいます。同図 (b) のサンプリング周期は 1 時間、(c)(d) のサンプリング周期は 4 時間です。細かな変動も記録したい場合は、サンプリング周期を短くします。この場合、変動の「最も高い周波数成分の 2 倍の周波数でサンプリングすればよい」ことが知られています (**サンプリング定理**)。

また、測定データの値の精度に関しては、同図 (b)(c) は 0.1 ℃刻み、(d) は 2 ℃刻みに測定しており、これを**量子化**と呼びます。

今、データの範囲を −30 ℃〜49.9 ℃とします。0.1 ℃刻みでデータを記録する場合は、800 の値のいずれであるかを記録しなければならないので 10 ビット必要です。それを 1 時間毎に測定するとすれば、1 日分のデータは 10 ビット× 24 回=240 ビットとなります (図 2.1(b))。

気温の精度が 2 ℃刻みで良いならば、−30 ℃〜48 ℃の 40 の値のいずれであるかを記録すればよいので、6 ビットで済みます。それを 4 時間毎に測定するとすれば、1 日分のデータは 6 ビット× 6 回=36 ビットとなります (図 2.1(d))。

図の (c) や (d) を見ると、気温の細かな時間変化の情報が捨てられていることがわかります。実は捨てられるだけでなく、気温の細かな時間変化が精度に悪影響を及ぼすことから (たまたま細かな時間変動の峰や谷を標本化してしまう)、標本化の前に細かな時間変動を除去する平滑化 (フィルタリング) を行う必要があります (これは音や画像でも同様)[1]。

サンプリング周期や量子化レベルをどの程度にするかは、測定対象や目的によって異なります。

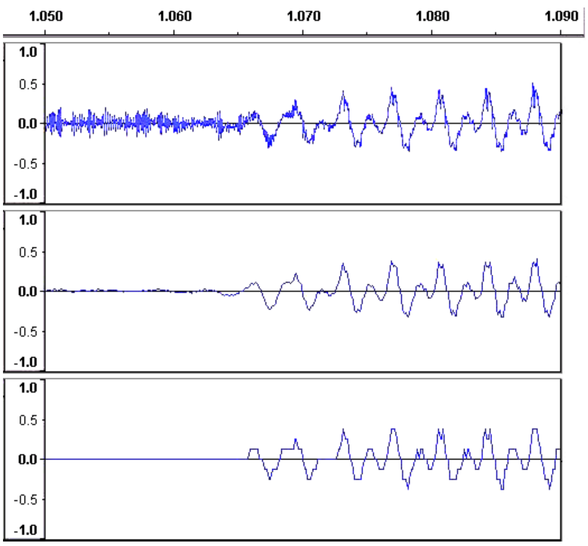


図 2.2: 音声波形の一部 (1.05 秒〜1.09 秒)

2.2 音のデジタル化

我々の耳はおよそ 20〜20000Hz の音を聴きとることができます。CD に記録されている音は、この範囲の音を忠実に再現できるように、44100Hz でサンプリングされています。

[1] このフィルタをアンチエイリアシングフィルタといいます。

一方、電話などの通信で音声を明瞭に相手に伝えるには300〜3000Hz の帯域で十分で、8000Hz でサンプリングされます。この場合は、事前にフィルタで 4000Hz に帯域制限した後、8000Hz でサンプリングされます。

いずれの場合もサンプリング周波数の 2 分の 1 以上の高い周波数の音 (細かな時間変動) が標本値の精度 (歪み) に悪影響を及ぼさないように、事前に 20000Hz または 4000Hz で帯域制限した後で標本化します。

表 2.1 に示す音を試聴すると、サンプリング周波数 (帯域制限) や量子化ビット数による音の違いがわかります。量子化ビット数に関して、16 ビットの場合 (B) と 8 ビットの場合 (C) で大きな違いが感じられないかもしれません。これは、8 ビットの音を録音する際に、音声波形の最大値が 255 を超えない程度に録音レベルを上げるよう心掛けたためです。実際の音 (音声や音楽) の大きさは、小さい音から大きい音まで広範囲に及びます。このため、大きな音でも歪まないよう録音レベルを下げると、通常の音は数ビットでしか量子化されず、量子化歪みが目立つ結果となります。

図 2.2 に (A)CD 品質の音、(B) 帯域制限した音、(D) 量子化を粗くした音のそれぞれの音声波形の一部 (1.05 秒〜1.09 秒) を示します。波形からも、帯域制限を行ったために細かな時間変動が除去されていること (A → B)、量子化を粗くすると階段状の波形となること (B → C) がわかります。パソコンやその他の録音機器で録音する際には、適切な録音レベル (歪まない程度に録音レベルを上げる) よう注意するとよいでしょう。

(A) の CD 品質の音をあきらめ、(C) の電話品質で我慢したとすれば、ファイルサイズは 10 分の 1 以下になっていて、メモリ容量の節約になります。ビットレートに着目すれば、同じ回線料金で 10 人が同時に会話できることになります。目的に応じた品質を心掛けるのがよいでしょう。

(練習) 自分の声を録音し、CD 品質と電話品質で保存し、音の違いを確認しなさい。また、ファイルサイズを調べなさい。

表 2.1: 音声の標本化と量子化 (5 秒間の音声)

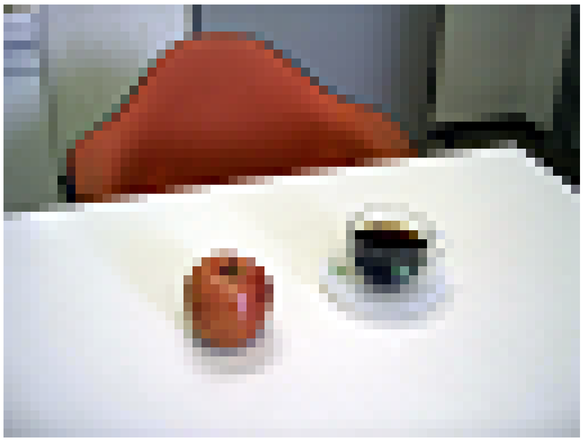
サンプリング周波数	量子化ビット数	ファイルサイズ	ビットレート	品質
44.1kHz	16 bit	441134 byte	706 kbps	CD 品質 (サンプリングや量子化の影響がわからない)
8kHz	16 bit	80052 byte	128 kbsp	帯域制限された音
8kHz	8 bit	40048 byte	64 kbps	電話品質
8kHz	4 bit	(約 20000 byte)	32 kbps	量子化雑音が目立つ

2.3 画像のデジタル化

気温も音も、時間とともに変化する情報を時間軸で標本化しました。ここでは、写真画像のデジタル化をとりあげます。時間軸の標本化ではなく、写真画像の縦方向と横方向の空間の標本化となります。

図 2.3 は同じ対象空間を解像度 (量子化密度) と量子化ビット数を変えたものです。640×480 ドット 24 ビット ($2^{24} = 16777216 = 1670$ 万色) のデジタルカメラの画像 (a) に対し、(b) はサンプリング密度を粗くしたもの (64×48 ドット)、(c) は量子化レベルを粗くしたもの (各ドット 8 ビット $2^8 = 256$ 色)、(d) はサンプリング密度も量子化レベルも粗くしたものです。

(a) 640 × 480 ドット 24 ビット (1670 万色)
ファイルサイズは約 640 × 480 × 3=921600 バイト



(b) 64 × 48 ドット 24 ビット (1670 万色)
ファイルサイズは約 64 × 48 × 3=9216 バイト

(c) 640 × 480 ドット 8 ビット (256 色)
ファイルサイズは約 640 × 480 × 1=307200 バイト

(d) 64 × 48 ドット 8 ビット (256 色)
ファイルサイズは約 64 × 48 × 1=3072 バイト

図 2.3: 画像の標本化と量子化

最近のデジタルカメラは性能が向上しており、目的に応じた解像度で保存するよう心掛けるとよいでしょう。

(練習) デジタルカメラで撮影した画像の解像度と量子化ビット数を変えて保存し、品質の違いを確認しなさい。また、ファイルサイズを調べなさい。

第3章　確率

3.1　標本空間と事象

サイコロ投げのように、結果が確率的であるような行為を**試行** (trial) といいます。

試行により現れうる全ての結果を要素とする集合を**標本空間** (sample space) といいます。個々の試行の結果は標本空間の要素となります。

サイコロ投げの標本空間は次のようになります。

$$S = \{1\text{ の目}, 2\text{ の目}, 3\text{ の目}, 4\text{ の目}, 5\text{ の目}, 6\text{ の目}\}$$

事象 (event) は標本空間の ある部分集合です。集合 A のいずれかの要素が生起した場合に、「事象 A が生起した」といいます。サイコロ投げで「奇数の目」を事象 A、「偶数の目」を事象 B とすると、3 の目が出たならば「事象 A が生起した」ということになります。

$$A = \{1\text{ の目}, 3\text{ の目}, 5\text{ の目}\}$$

$$B = \{2\text{ の目}, 4\text{ の目}, 6\text{ の目}\}$$

標本空間も、それ自身が標本空間の部分集合なので、事象となります。これを**全事象**といいます。試行を行った場合、全事象は必ず生起することになります。

標本空間の各要素もそれぞれが事象となります (要素事象)。この場合、ひとつの要素事象 (例えば「3 の目」) が生起した時に同時に他の要素事象が生起することはありません。このような場合、**互いに排反** (exclusive) な事象といいます。上記の例で、事象 A(奇数の目) と事象 B(偶数の目) も互いに排反となります。事象 C を「3 より大きい目」とすると A と C、B と C は互いに排反ではありません。

$$C = \{4\text{ の目}, 5\text{ の目}, 6\text{ の目}\}$$

赤と白の 2 つのサイコロを同時に投げる試行を考えてみましょう。この時の標本空間は、以下のように 36 個の要素からなる集合として表すことができます。

$$S = \{(\text{赤 1 白 1}), (\text{赤 1 白 2}), (\text{赤 1 白 3}), ..., (\text{赤 6 白 6})\}$$

また、まったく同じ行為について、2 つのサイコロの目の和に着目すれば、以下のような 11 個の要素からなる標本空間を考えることもできます。

$$S = \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$$

（練習）トランプ 52 枚の中から 1 枚を引くという試行を考えると、次のように標本空間を表すことができます。

$$S = \{\clubsuit A, \clubsuit 2, \clubsuit 3, ..., \spadesuit Q, \spadesuit K\}$$

この時、事象 A をエースのカード、事象 C をクラブのカードとすると、A と C は互いに排反かどうかを調べなさい。

3.2　公理による確率の定義

標本空間を S とし、標本空間の中に事象 A を定義します。この時、以下の 3 つの公理を満たす $P(\ )$ を確率として定義します。

(公理 1) どの事象の確率も負ではない。

$$P(A) \geq 0 \tag{3.1}$$

(公理 2) 全事象が生起する確率は 1 である。

$$P(S) = 1 \tag{3.2}$$

(公理 3) $A_1, A_2, A_3, ...$ が互いに排反な事象であれば

$$P(A_1 \cup A_2 \cup A_3 \cup ...) = P(A_1) + P(A_2) + P(A_3) + ... \tag{3.3}$$

これらの公理は、私達が直観的に持っている確率の概念、つまり「等しく起こりやすい」、「相対頻度」、「確信の度合」などの概念と矛盾するものではありません。

サイコロの例では、正確に作られたサイコロでも、そうでないサイコロでも、$P(1\text{ の目}) \geq 0$、$P(1〜6\text{ のいずれかの目}) = 1$、$P(1\text{ または }2\text{ の目}) = P(1\text{ の目}) + P(2\text{ の目})$ などは理にかなったものです。

（練習） 2 つのサイコロを投げた時、目の和に着目し、以下の標本空間を考えます。

$$S = \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$$

また、これらの要素事象の生起確率が以下のようであったとします。

表 3.1: 2 つのサイコロの目の和の生起確率

目の和	確率
2	$\frac{1}{36}$
3	$\frac{2}{36}$
4	$\frac{3}{36}$
5	$\frac{4}{36}$
6	$\frac{5}{36}$
7	$\frac{6}{36}$
8	$\frac{5}{36}$
9	$\frac{4}{36}$
10	$\frac{3}{36}$
11	$\frac{2}{36}$
12	$\frac{1}{36}$

公理 3 に注意し、「目の和が 7 または 11」の確率を求めなさい。次に、「目の和が 2 でも 3 でも 12 でもない」確率を求めなさい。

プログラム　dice.htm

以下のプログラムはサイコロ投げのシミュレーションプログラムです。初めに何回投げるかを指定し (変数 `n` に入る)、その回数だけ 1〜6 のいずれかを各 $\frac{1}{6}$ の確率でランダムに表示します。`Math.random()` で 0〜1 の一様乱数を生成しています。if 文の箇所で 1〜6 の目の出る確率を調整できます (現在はどの目も $\frac{1}{6}$)。

```
<script>
n=prompt("何回投げますか?","")*1;
for(i=0;i<n;i++){
    r=Math.random();
    if(r<1/6) num=1;
    else if(r<2/6) num=2;
    else if(r<3/6) num=3;
    else if(r<4/6) num=4;
    else if(r<5/6) num=5;
    else  num=6;
    document.write("サイコロの目は"+num+"<br>");
}
</script>
```

3.3　結合確率

2 つの事象 A と B の和 $A \cup B$ の生起確率 $P(A \cup B)$ は次のようになります (加法定理)。

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) \tag{3.4}$$

これは、要素事象についてベン図を用いて考えると明らかです。ここで、$P(A \cap B)$ を**結合確率** (joint probability) といいます。$P(A, B)$ と書くこともあります。A と B が互いに排反ならば $A \cap B = \phi$ なので、

$$P(A \cup B) = P(A) + P(B) \tag{3.5}$$

となります。

（練習） 前節の練習に引き続き、2 つのサイコロを投げた時の目の和を要素事象とする標本空間を考えます。ここでは事象 A, B が次のようであるとします (図 3.1)。

$$A = \{4, 5, 6, 7\}$$

$$B = \{7, 8, 9, 10, 11\}$$

この時、$P(A \cup B)$ を求めなさい。

3.4　条件付き確率と統計的独立

条件付き確率 (conditional probability) を次のように定義する。

$$P(A|B) \stackrel{\text{def}}{=} \frac{P(A \cap B)}{P(B)} \tag{3.6}$$

例えば、良く作られたサイコロがあったとします。

$$P(1\text{ の目}) = P(2\text{ の目}) = ... = P(6\text{ の目}) = \frac{1}{6}$$

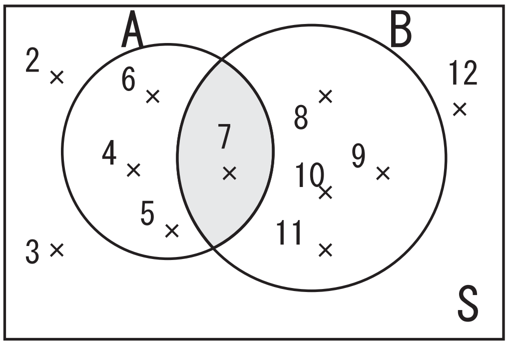


図 3.1: 2 つの目の和を要素事象とする標本空間

このサイコロを投げた時の出る目を要素事象とする標本空間 S は次のようになります。

$$S = \{1\text{ の目}, 2\text{ の目}, 3\text{ の目}, 4\text{ の目}, 5\text{ の目}, 6\text{ の目}\}$$

ここで、2 の目を事象 A とし、偶数の目を事象 B とします。

$$A = \{2\text{ の目}\}$$

$$B = \{2\text{ の目}, 4\text{ の目}, 6\text{ の目}\}$$

2 の目が出る確率は $P(A) = \frac{1}{6}$ ですが、もしもその目が偶数であることを知らされたとしたら、2 の目が出る確率は次のようになります。

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{\frac{1}{6}}{\frac{1}{6} + \frac{1}{6} + \frac{1}{6}} = \frac{1}{3}$$

次に、

$$P(A|B) = P(A) \tag{3.7}$$

ならば、A の生起確率が B の生起によらないので、この時 A と B は**統計的に独立**といいます。上式は書き換えると次のようになります (乗法定理)。

$$P(A|B)P(B) = P(A \cap B) = P(A)P(B) \tag{3.8}$$

上の例では $A = \{2\text{ の目}\}$ と $B = \{\text{偶数の目}\}$ は統計的に独立ではありません。

赤と白のサイコロを同時に投げる試行を考え、赤のサイコロの目が1〜6となる事象を $A_1, A_2, ..., A_6$、白のサイコロの目が 1〜6 となる事象を $B_1, B_2, ..., B_6$ とした場合、赤のサイコロの目が白のサイコロの目に影響することはないので、

$$P(A_i|B_j) = P(A_i)$$

で、A_i と B_j は統計的に独立と考えられます。

（練習） 前節の練習に引き続き、2 つのサイコロを投げた時の目の和を要素事象とする標本空間を考えます。前の例と同様に

$$A = \{4, 5, 6, 7\}$$

$$B = \{7, 8, 9, 10, 11\}$$

とします。この時、$P(A|B)$ を求めなさい。

3.5　ベイズの定理

条件付き確率を次のように定義します。

$$P(A|B) \stackrel{\text{def}}{=} \frac{P(A \cap B)}{P(B)}$$

$$P(B|A) \stackrel{\text{def}}{=} \frac{P(B \cap A)}{P(A)}$$

これらから、以下の式が導かれます。

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{3.9}$$

これを**ベイズの定理** (Bayes theorem) といいます。

ある事象 B と、B が起きる原因と考えられる n 個の排反事象 $A_1, A_2, ..., A_n$ を考えます。このとき、$P(B|A_i)$ は「A_i が原因で B が起きる確率」を表します。また、原因に関する確率 $P(A_i)$ を事前確率 (*a priori* probability)、$P(A_i|B)$ すなわち「B が起きたことを知って、それが原因 A_i から起きたと考えられる確率」を事後確率 (*a posteriori* probability) ということがあります。この時、$P(A_i|B)$ は以下のように表すことができます。

$$P(A_i|B) = \frac{P(A_i)P(B|A_i)}{P(B)} = \frac{P(A_i)P(B|A_i)}{\sum_i P(A_i \cap B)} = \frac{P(A_i)P(B|A_i)}{\sum_i P(A_i)P(B|A_i)} \tag{3.10}$$

興味深い例を紹介します。ある病気が流行していて、感染者の割合は0.1%であったとします (事前確率)。

$$P(\text{感染}) = 0.001, P(\text{非感染}) = 0.999$$

この病気に感染しているかどうかを調べる簡単な検査があり、感染していれば 0.99 の確率で陽性と判定されるが、感染していなくとも 0.02 の確率で陽性と判定されるものとします。

$$P(\text{陽性}\mid\text{感染}) = 0.99, P(\text{陽性}\mid\text{非感染}) = 0.02$$

では、ある人がその検査を受けた結果が陽性であったならば、その病気に感染している確率 (事後確率) はいくらかという問題です。

$$\begin{aligned} P(\text{感染}\mid\text{陽性}) &= \frac{P(\text{陽性}\mid\text{感染})P(\text{感染})}{P(\text{陽性})} = \frac{P(\text{陽性}\mid\text{感染})P(\text{感染})}{P(\text{陽性}\mid\text{感染})P(\text{感染}) + P(\text{陽性}\mid\text{非感染})P(\text{非感染})} \\ &= \frac{0.99 \times 0.001}{0.99 \times 0.001 + 0.02 \times 0.999} = 0.0472 \end{aligned}$$

つまり、この例では、陽性と判定されても、感染している確率は5%未満であることがわかります。

（練習）冬の天気について考えます。青森が晴の確率が 0.1、雪の確率が 0.9 とします (事前確率)。また、青森が晴の時には弘前が晴の確率が 0.8、雪の確率が 0.2、青森が雪の時には弘前が晴の確率が 0.3、雪の確率が 0.7 とします。弘前が晴であった時に青森が晴の確率 (事後確率) を求めなさい。

（練習）ある TV には M 社の IC が 80%、T 社の IC が 20%使われています (事前確率)。M 社の IC が原因で TV が故障する確率が 0.01、T 社の IC が原因で TV が故障する確率が 0.02 であるとします。この TV が故障した時に、M 社の IC の破損が原因である確率と T 社の IC の破損が原因である確率 (事後確率) を求めなさい。

（練習）送信者が 0 か 1 を送信し、受信者が 0 か 1 かを受け取る通信の問題を考えます。送信者が 0 を送る確率が 0.75、1 を送る確率が 0.25 とします (事前確率)。

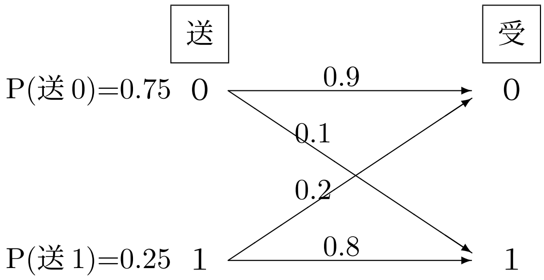
$$P(\text{送}\,0) = 0.75, P(\text{送}\,1) = 0.25$$

送信者が送った信号に雑音が加わり、受信者に届きます。結果、送信者が 0 を送った場合は 0.9 の確率で受信者に 0 として伝わり、送信者が 1 を送った場合は 0.8 の確率で受信者に 1 として伝わるものとします。

$$P(\text{受}\,0|\,\text{送}\,0) = 0.9, P(\text{受}\,1|\,\text{送}\,0) = 0.1$$

$$P(\text{受}\,1|\,\text{送}\,1) = 0.8, P(\text{受}\,0|\,\text{送}\,1) = 0.2$$

受信者が 1 を受け取る確率を求めなさい。また、受信者が 1 を受け取った時に、送信者が 1 を送った確率 (事後確率) を求めなさい。



3.6　確率変数と平均と分散

標本空間 S の中で定義される数として、変数 X を考えます。この変数 X がある具体的な値 x をとる確率 $P(x)$ がわかっているとします。このような変数 X を**確率変数** (random variable) といいます。確率変数 X の**平均** (mean)μ および**分散** (variance)σ^2 は次のように定義されます。

$$\mu \stackrel{\text{def}}{=} E[X] \tag{3.11}$$

$$\sigma^2 \stackrel{\text{def}}{=} E[(X-\mu)^2] \tag{3.12}$$

ただし、$E[X]$ は**期待値** (expectation) の演算を表し、

$$E[X] = \sum_x xP(x) \tag{3.13}$$

を意味します。

1〜6 の目が出る確率がいずれも $\frac{1}{6}$ のサイコロを投げた時の目を確率変数と考えます。この時、平均と分散は次のようになります。

$$\begin{aligned}\mu &= E[X] = \sum_x xP(x) \\ &= 1\cdot\frac{1}{6}+2\cdot\frac{1}{6}+3\cdot\frac{1}{6}+4\cdot\frac{1}{6}+5\cdot\frac{1}{6}+6\cdot\frac{1}{6} = 3.5\end{aligned}$$

$$\begin{aligned}\sigma^2 &= E\left[(X-\mu)^2\right] = \sum_x (x-\mu)^2 P(x) \\ &= (1-3.5)^2\cdot\frac{1}{6}+(2-3.5)^2\cdot\frac{1}{6}+(3-3.5)^2\cdot\frac{1}{6}+(4-3.5)^2\cdot\frac{1}{6}+(5-3.5)^2\cdot\frac{1}{6}+(6-3.5)^2\cdot\frac{1}{6} \\ &= 2.92\end{aligned}$$

（練習） 2 個のサイコロを投げた時、目の和を確率変数と考え、その確率は表 3.1 のようであったとします。平均と分散を計算しなさい。

（練習） 1000 本のくじの中に、1 等 1,000,000 円が 1 本、2 等 50,000 円が 10 本、3 等 10,000 円が 100 本が含まれ、他はすべて空くじであるとします。この中から 1 本引く人の期待値 (金額) を求めなさい。

プログラム　kuji.htm

以下のプログラムは、上の練習問題にある「くじ引き」プログラムです。期待値を求めているわけではなく、単に与えられた確率で当たりくじが出るというものです。

```
<script>
r=Math.random();
if(r<1/1000) document.write("1 等賞 1,000,000 円");
else if(r<1/1000+10/1000) document.write("2 等賞 50,000 円");
else if(r<1/1000+10/1000+100/1000) document.write("3 等賞 10,000 円");
else  document.write("ハズレ");
</script>
```

3.7　標本平均と標本分散

今までは予め確率が与えられた場合について考えてきましたが、今度はこれらが未知の場合について考えてみましょう。ある細工が施されたサイコロがあったとして、その性質 (平均と分散) を知りたいというような場合です。

初めに、サイコロを n 回投げた時の**標本平均** (sample mean)$\bar{X}$ を考えてみましょう。このような確率変数の関数を一般に**統計量**といいます。

$$\bar{X} \stackrel{\text{def}}{=} \frac{X_1 + X_2 + ... + X_n}{n} \tag{3.14}$$

実際にサイコロを投げることによって得られる実現値は

$$\bar{x} \stackrel{\text{def}}{=} \frac{x_1 + x_2 + ... + x_n}{n} \tag{3.15}$$

です。この実現値を平均 μ の**推定値** (estimate) とします。この標本平均の期待値は次のようになります[1]。

$$E[\bar{X}] = \mu \tag{3.16}$$

n 回投げた時の標本平均の期待値は、1 回投げた時の平均 (期待値) に等しいことがわかります。これは推定値として用いる時に好ましい性質のひとつです。このような推定を**不偏推定** (unbiased estimate) といいます。

次に、この標本平均の分散を調べると次のようになります[2]。

$$E\left[(\bar{X} - E[\bar{X}])^2\right] = \frac{\sigma^2}{n} \tag{3.17}$$

---

[1]

$$E[\bar{X}] = E\left[\frac{X_1 + X_2 + ... + X_n}{n}\right] = \frac{E[X_1] + E[X_2] + ... + E[X_n]}{n} = \mu$$

[2]

$$\begin{aligned} E\left[(\bar{X} - E[\bar{X}])^2\right] &= E\left[(\bar{X} - \mu)^2\right] = \frac{1}{n^2} E\left[((X_1 - \mu) + (X_2 - \mu) + ... + (X_n - \mu))^2\right] \\ &= \frac{1}{n^2}\left\{\sum_i (X_i - \mu)^2 + \sum_{i \neq j} E[(X_i - \mu)(X_j - \mu)]\right\} = \frac{\sigma^2}{n} \end{aligned}$$

n 回投げた時の標本平均の分散は、1 回投げた時の分散の $\frac{1}{n}$ になり、$n \to \infty$ の時に標本平均は μ に収束します[3]。

次に**標本分散** (sample variance) について考えてみましょう。

$$S^2 \stackrel{\mathrm{def}}{=} \frac{1}{n-1}\sum_i (X_i - \bar{X})^2 \tag{3.18}$$

実際にサイコロを投げることによって得られる実現値は

$$s^2 \stackrel{\mathrm{def}}{=} \frac{1}{n-1}\sum_i (x_i - \bar{x})^2 \tag{3.19}$$

です。この標本分散の期待値は次のようになります[4]。

$$E\left[S^2\right] = \sigma^2 \tag{3.20}$$

n 回投げた時の標本分散の期待値は、1 回投げた時の分散に等しく、不偏推定となっています。なお、標本分散を

$$s^2 \stackrel{\mathrm{def}}{=} \frac{1}{n}\sum_i (x_i - \bar{x})^2 \tag{3.21}$$

で計算することもあるが、この場合は不偏推定とはなりません。

以上のような標本平均や標本分散の持つ性質は、確率の分布によらず成り立つことを強調しておきます。

[3] 任意の $\varepsilon > 0$ に対して下式が成立するというもので、**大数の法則**と呼ばれています。なお、このような収束は確率収束と呼ばれます。

$$\lim_{n\to\infty} P(|\bar{X} - \mu| > \varepsilon) = 0$$

[4]

$$\begin{aligned}
E\left[S^2\right] &= \frac{1}{n-1}\sum_i E\left[\left((X_i-\mu)-(\bar{X}-\mu)\right)^2\right] \\
&= \frac{1}{n-1}\sum_i E\left[(X_i-\mu)^2 + (\bar{X}-\mu)^2 - 2(X_i-\mu)(\bar{X}-\mu)\right] \\
&= \frac{1}{n-1}\sum_i \left(\sigma^2 + \frac{\sigma^2}{n} - 2E\left[(X_i-\mu)(\bar{X}-\mu)\right]\right) \\
&= \frac{1}{n-1}\left(n\sigma^2 + \sigma^2 - \frac{2}{n}\sum_{i,j} E\left[(X_i-\mu)(X_j-\mu)\right]\right) \\
&= \frac{1}{n-1}\left(n\sigma^2 - \sigma^2\right) = \sigma^2
\end{aligned}$$

プログラム diceave.awk

以下のプログラムはサイコロ投げのシミュレーションプログラムです。初めに何回投げるかを指定し (変数 `n` に入る)、その回数だけ 1〜6 のいずれかを各 $\frac{1}{6}$ の確率でランダムに生成し、目の数の標本平均を求めています。

```
<script>
n=prompt("何回投げますか? ","");
sum=0;
x=new Array();
for(i=0;i<n;i++){
    r=Math.random();
    if(r<1/6) num=1;
    else if(r<2/6) num=2;
    else if(r<3/6) num=3;
    else if(r<4/6) num=4;
    else if(r<5/6) num=5;
    else  num=6;
    document.write(num+", ");
    x[i]=num;
    sum+=num;
}
ave=sum/n;
sum2=0;
for(i=0;i<n;i++) sum2+=Math.pow(x[i]-ave,2);
v=sum2/(n-1);
document.write("<br>m="+ave);
document.write("<br>s<sup>2</sup>="+v);
document.write("<br>"+"s="+Math.sqrt(v));
</script>
```

第4章　情報量とエントロピー

(対数計算)

$$
\begin{aligned}
&\log_a b = c \Leftrightarrow b = a^c (\text{対数の定義})\\
&\log_a b = \frac{\log_b}{\log_a}\\
&\log_a (xy) = \log_a x + \log_a y\\
&\log_a \frac{x}{y} = \log_a x - \log_a y\\
&\log_a x^y = y \log_a x\\
&-\log_a \frac{1}{x} = \log_a x
\end{aligned}
$$

$$\log_2 1 = 0,\ \log_2 2 = 1,\ \log_2 3 = 1.58496,\ \log_2 4 = 2,\ \log_2 5 = 2.32193,$$

$$\log_2 6 = 2.58496,\ \log_2 7 = 2.80735,\ \log_2 8 = 3,\ \log_2 11 = 3.45943$$

4.1　自己情報量と相互情報量

この章では、情報源から発せられる情報の「量」をどのように表すかを示します。まずは、ある事象の生起がもたらす自己情報量と相互情報量について学びます。

4.1.1　自己情報量

めったに起きない出来事が起きたと聞いた時に受け取る情報の量は大きく、逆に、当然のように起きる出来事が起きたと聞いた時に受け取る情報の量は小さいと考えられます。つまり、情報量は、ある事象が生起したことを知った時の「驚きの度合」と考えることができます。

事象 A が生起する確率を $P(A)$ とした時、Shannon はその**自己情報量**を以下のように定義しました。

$$I(A) \stackrel{\text{def}}{=} -\log_2 P(A) \tag{4.1}$$

情報量の単位は **bit** で、確率 $\frac{1}{2}$ で生起する事象が実際に生起したことがもたらす情報量が 1 bit となります。

良く作られたコイン ($P(\text{表}) = P(\text{裏}) = \frac{1}{2}$) を投げた時に、表が出たとしたら $I(\text{表}) = -\log_2 \frac{1}{2} = 1\text{bit}$ の情報をもたらします。裏が出た時も、同じく 1bit の情報をもたらします。$P(\text{表}) = \frac{1}{4}$ のイカサマコインを投げた時に、表が出たとしたら $I(\text{表}) = -\log_2 \frac{1}{4} = 2\text{bit}$ の情報をもたらします。

では $P(\text{表}) = 0.999$ のイカサマコインの場合はどうでしょうか。$I(\text{表}) = -\log_2 0.999 = 0.0014\text{bit}$ で、この場合はほとんど表が出ることがわかっているので、情報量 (驚きの度合) は小さな値とな

ります。逆に裏が出たとしたら、$I(\text{裏}) = -\log_2 0.001 = 9.97\text{bit}$ で、情報量 (驚きの度合) は大きな値となります。

今度は良く作られたサイコロ ($P(1\text{ の目}) = P(2\text{ の目}) = ... = P(6\text{ の目}) = \frac{1}{6}$) について考えてみましょう。このサイコロを投げた時に 2 の目が出たとしたら、そのことがもたらす情報量は $I(2\text{ の目}) = -\log_2 \frac{1}{6} = 2.58\text{bit}$ となります。

（練習） 次のようなイカサマサイコロを投げた時に、3 の目が出たとしたら、そのことがもたらす情報量はいくらか。また、2 の目が出た場合はどうか。

$$P(1\text{ の目}) = \frac{1}{8}, P(2\text{ の目}) = \frac{1}{2}, P(3\text{ の目}) = \frac{1}{16}, P(4\text{ の目}) = \frac{1}{16}, P(5\text{ の目}) = \frac{1}{8}, P(6\text{ の目}) = \frac{1}{8}$$

4.1.2　相互情報量

自己情報量の自然な拡張として**条件付自己情報量**を次のように定義します。

$$I(a_i|b_j) \stackrel{\text{def}}{=} -\log_2 P(a_i|b_j) \tag{4.2}$$

これを用いて、「事象 b_j の生起を知ることが事象 a_i の生起にもたらす情報量」を**相互情報量**として次のように定義します。

$$I(a_i; b_j) \stackrel{\text{def}}{=} I(a_i) - I(a_i|b_j) \tag{4.3}$$

サイコロの例で考えましょう。

良く作られたサイコロを投げて、2 の目が出たとして、そのことをすぐに知ったとすれば、$I(2\text{ の目}) = -\log_2 \frac{1}{6} = 2.58\text{bit}$ の情報量を得ることになります。ところが、その結果を知る前に、誰かが「偶数の目が出た」と教えてくれたとします。すると、この時点で、$P(2\text{ の目} \mid \text{偶数}) = \frac{1}{3}$ となるので、その後に 2 の目が出たことを知らされてもそのことがもたらす情報量は $I(2\text{ の目} \mid \text{偶数}) = -\log_2 \frac{1}{3} = 1.58\text{bit}$ となります。「偶数の目が出た」という知らせが「2 の目が出る」事象の生起にもたらす相互情報量は

$$I(2\text{ の目}; \text{偶数の目}) = I(2\text{ の目}) - I(2\text{ の目} \mid \text{偶数の目}) = 2.58 - 1.58 = 1[\text{bit}]$$

となります。

上の例からわかるように、情報を受け取ることにより不確実さが減少し、情報の量とは「不確実さの減少量」とみることができます。

自己情報量と相互情報量の関係を図 4.1 に示します。

3 章で取り上げた、病気の感染と検査の関係の例を考えましょう。ある病気が流行していて、$P(\text{感染}) = 0.001, P(\text{非感染}) = 0.999$ とします。この病気に感染していることがわかった時に得る情報量は $I(\text{感染}) = -\log_2 P(\text{感染}) = -\log_2 0.001 = 9.97\text{bit}$ です。感染について調べる検査方法があり、その精度は $P(\text{陽性} \mid \text{感染}) = 0.99, P(\text{陽性} \mid \text{非感染}) = 0.02$ とします。

この検査を受けた結果が陽性であったならば、その病気に感染している確率はベイズの定理を用いて

$$P(\text{感染} \mid \text{陽性}) = \frac{P(\text{陽性} \mid \text{感染})P(\text{感染})}{P(\text{陽性} \mid \text{感染})P(\text{感染}) + P(\text{陽性} \mid \text{非感染})P(\text{非感染})} = 0.0472$$

と計算できました。このことから、検査の結果が陽性であることを知った時点で、感染していることがわかった時に得る情報量は

$$I(\text{感染} \mid \text{陽性}) = -\log_2 0.0472 = 4.41[\text{bit}]$$

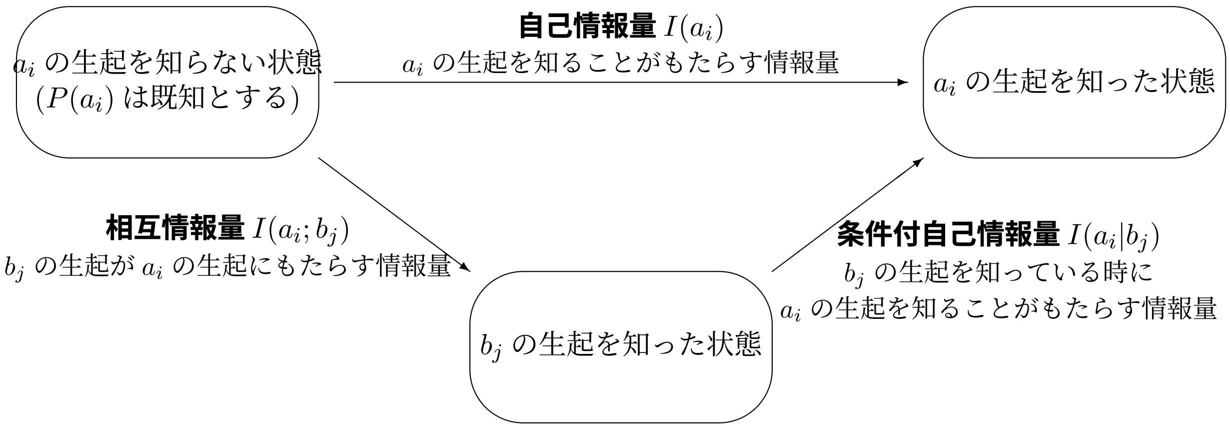


図 4.1: 自己情報量と相互情報量

となります。従って、陽性という検査結果が、感染という事実に関して与える相互情報量は

$$I(\text{感染}; \text{陽性}) = I(\text{感染}) - I(\text{感染} \mid \text{陽性}) = 9.97 - 4.41 = 5.56[\text{bit}]$$

となります。これは、「感染」に関する不確実さの減少量と考えることができます。

（練習） 3 章の練習で取り上げた、冬の天気の問題を考えましょう。前と同様に

$$P(\text{青森晴}) = 0.1, P(\text{青森雪}) = 0.9$$

$$P(\text{弘前晴} \mid \text{青森晴}) = 0.8, P(\text{弘前雪} \mid \text{青森晴}) = 0.2$$

$$P(\text{弘前晴} \mid \text{青森雪}) = 0.3, P(\text{弘前雪} \mid \text{青森雪}) = 0.7$$

とします。「青森晴」を知る自己情報量はいくらか。「弘前晴」であったときに、「青森晴」を知る条件付自己情報量はいくらか。「弘前晴」が「青森晴」についてもたらす相互情報量はいくらか。

（練習） 3 章の練習で取り上げた、送信者が 0 か 1 を送信し、受信者が 0 か 1 かを受け取る通信の問題を考えましょう。前と同様に

$$P(\text{送 } 0) = 0.75, P(\text{送 } 1) = 0.25$$

$$P(\text{受 } 0 | \text{ 送 } 0) = 0.9, P(\text{受 } 1 | \text{ 送 } 0) = 0.1$$

$$P(\text{受 } 1 | \text{ 送 } 1) = 0.8, P(\text{受 } 0 | \text{ 送 } 1) = 0.2$$

とします。1 が送信されたことを知る自己情報量はいくらか。1 を受信した時点で、1 が送信されたことを知る条件付自己情報量はいくらか。1 を受信したことが 1 を送信したことについてもたらす相互情報量はいくらか。

4.2 平均情報量 (エントロピー) と平均相互情報量

4.2.1 平均情報量 (エントロピー)

これまで見てきたように、自己情報量は「個々の事象について」その情報がもたらす驚きの量を表すものでした。この期待値、すなわち「事象の生起がもたらす平均の情報量」が**平均情報量 (エントロピー)** です。

$$H(X) \stackrel{\text{def}}{=} \sum_i P(A_i) I(A_i) = -\sum_i P(A_i) \log_2 P(A_i)$$

良く作られたコイン ($P(表) = P(裏) = \frac{1}{2}$) を投げた時に、表が出たとしたら $I(表) = -\log_2 \frac{1}{2} = 1$bit、裏が出た時も同じく $I(裏) = 1$bit の情報をもたらします。その平均は $H(X) = P(表)I(表) + P(裏)I(裏) = 1$bit となります。

$P(表) = \frac{1}{4}, P(裏) = \frac{3}{4}$ のイカサマコインを投げた時には、各々の事象は $I(表) = -\log_2 \frac{1}{4} = 2$bit、$I(裏) = -\log_2 \frac{3}{4} = 0.415$bit の情報をもたらします。その平均は $H(X) = P(表)I(表) + P(裏)I(裏) = \frac{1}{4} \times 2 + \frac{3}{4} \times 0.415 = 0.811$bit となります。このコインは、1 回の試行につき平均 0.811bit の情報をもたらすことになります。

$P(表) = p$ とすると $P(裏) = 1 - p$ で、p が変化した時に H がどのように変化するかを示したのが次式です。

$$H = -p \log_2 p - (1-p) \log_2 (1-p)$$

H の値は $p = 0.5$ の時に最大の値 1 bit となります (下のプログラム参照)。$p = 0.5$ とは、「表か裏かまったく予想がつかない」という状況を表しています。一方、$p = 0.01$ の場合は「99%裏が出る」ということなので、たまに表が出て大きな情報量をもたらすとしても、平均すれば 1 回の試行がもたらす情報量は小さいものになります。

今度は良く作られたサイコロ ($P(1 の目) = P(2 の目) = ... = P(6 の目) = \frac{1}{6}$) について考えてみましょう。各々の目についての自己情報量は、$I(1 の目) = I(2 の目) = ... = I(6 の目) = 2.585$bit で、その平均は 2.585 bit となります。このサイコロを投げた時に出た目が何かを知ることは、1 回の試行について平均 2.585 bit の情報量を得ることになります。

(練習) 次のようなイカサマサイコロを投げた時に、出た目が何かを知ることは平均どれだけの情報量を得ることになるか。

$$P(1 の目) = \frac{1}{8}, P(2 の目) = \frac{1}{2}, P(3 の目) = \frac{1}{16}, P(4 の目) = \frac{1}{16}, P(5 の目) = \frac{1}{8}, P(6 の目) = \frac{1}{8}$$

(練習) 前出の病気の感染の例 ($P(感染) = 0.001, P(非感染) = 0.999$) で、感染しているかどうかを知ることはどれだけの情報を得ることになるか。

(練習) 前出の天気の例 ($P(青森晴) = 0.1, P(青森雪) = 0.9$) で、青森の天気を知ることはどれだけの情報を得ることになるか。

(練習) 前出の通信の例 ($P(送 0) = 0.75, P(送 1) = 0.25$) で、送った信号が何かを知ることはどれだけの情報を得ることになるか。

4.2.2 平均相互情報量

相互情報量は「ある b_j がある a_i の生起にもたらす情報量」でした。この期待値、すなわち「Y に関する事象の生起が X に関する事象の生起にもたらす平均の相互情報量」が**平均相互情報量**です。

$$I(X;Y) \stackrel{\text{def}}{=} \sum_{i,j} P(a_i, b_j) I(a_i; b_j) \tag{4.4}$$

ただし、X は $\{a_i\}$ の生起に関する確率変数、Y は $\{b_j\}$ の生起に関する確率変数とします。平均相互情報量 $I(X;Y)$ は、「Y に関して知ることが X の生起に関してもたらす情報量」で、前項の平均情報量、および条件付自己情報量を拡張した条件付平均情報量の関係として表すことができます。

条件付平均情報量 (条件付エントロピー) は次のように定義されます。

$$H(X|Y) \stackrel{\text{def}}{=} -\sum_{i,j} P(a_i, b_j) \log_2 P(a_i|b_j) \tag{4.5}$$

これを用いて、**平均相互情報量**は次のように表すことができます。

$$I(X;Y) = H(X) - H(X|Y) \tag{4.6}$$

サイコロの例で考えましょう。

良く作られたサイコロを投げて、出た目をすぐに知ったとすれば、$H(X) = -\sum \frac{1}{6} \log_2 \frac{1}{6} = 2.58\text{bit}$ の情報量を得ます。ところが、その結果を知る前に、誰かが「偶数か奇数か」を教えてくれたとしましょう。すると、この時点で、$P(1\text{の目} \mid \text{奇数}) = P(3\text{の目} \mid \text{奇数}) = P(5\text{の目} \mid \text{奇数}) = \frac{1}{3}, P(2\text{の目} \mid \text{偶数}) = P(4\text{の目} \mid \text{偶数}) = P(6\text{の目} \mid \text{偶数}) = \frac{1}{3}$ となるので、その後に出た目を知らされてもそのことがもたらす情報量は $H(X|Y) = -\sum P(a_i, b_j) \log_2 \frac{1}{3} = 1.58\text{bit}$ となります。偶数か奇数かの知らせ (Y) が出た目 (X) にもたらす平均相互情報量は

$$I(X;Y) = H(X) - H(X|Y) = 2.58 - 1.58 = 1[\text{bit}]$$

となります。

平均情報量と平均相互情報量の関係を図 4.2 に示します。

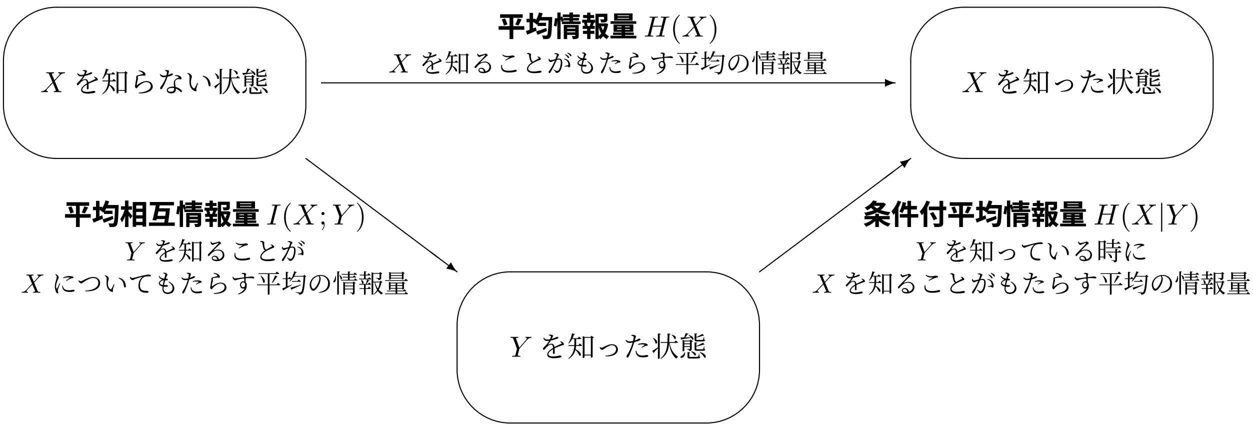


図 4.2: 平均情報量と平均相互情報量

平均相互情報量は

$$I(X;Y) = I(Y;X) \geq 0$$

という性質を持っています。このことを前出の天気の例にあてはめてみると、弘前の天気が青森の天気にもたらす情報量と青森の天気が弘前の天気にもたらす情報量が等しいということを意味しています。また、相互情報量が負にならないということは、弘前の天気を知ったことで青森の天気がより不確実になることはないことを意味しています。

??章で扱った例と同様、情報源アルファベットを $A = \{\texttt{\_}, \texttt{A}, \texttt{B}, ..., \texttt{Z}\}$ とする無記憶情報源を考えます。各々の情報源記号の生起確率は表 3.1 のようにわかっているものとします。この情報源から生成される記号列の 1 文字当たりの平均情報量は次のようになります。

$$\begin{aligned} H(X) &= -P(\texttt{\_}) \log_2 P(\texttt{\_}) - P(\texttt{A}) \log_2 P(\texttt{A}) - ... - P(\texttt{Z}) \log_2 P(Z) \\ &= -0.174 \log_2 0.174 - 0.067 \log_2 0.067 - ... - 0.001 \log_2 0.001 = 4.07[\text{bit}] \end{aligned}$$

（練習）次のようなイカサマサイコロを投げた時に、偶数か奇数かを知ることは出た目についてどれだけの情報量を与えるか。

$$P(1\text{ の目}) = \frac{1}{8}, P(2\text{ の目}) = \frac{1}{2}, P(3\text{ の目}) = \frac{1}{16}, P(4\text{ の目}) = \frac{1}{16}, P(5\text{ の目}) = \frac{1}{8}, P(6\text{ の目}) = \frac{1}{8}$$

（練習）前出の病気の感染の例で、検査結果がわかっている時に、感染しているかどうかを知ることの平均情報量はいくらか。また、検査は感染しているかどうかについてどれだけの情報を与えるか。

$$P(\text{感染}) = 0.001, P(\text{非感染}) = 0.999$$

$$P(\text{陽性} \mid \text{感染}) = 0.99, P(\text{陰性} \mid \text{感染}) = 0.01$$
$$P(\text{陽性} \mid \text{非感染}) = 0.02, P(\text{陰性} \mid \text{非感染}) = 0.98$$

（練習）前出の天気の例で、弘前の天気を知っている時に、青森天気を知ることの平均情報量はいくらか。また、弘前の天気を知ることは青森の天気に関してどれだけの情報を与えるか。

$$P(\text{青森晴}) = 0.1, P(\text{青森雪}) = 0.9$$

$$P(\text{弘前晴} \mid \text{青森晴}) = 0.8, P(\text{弘前雪} \mid \text{青森晴}) = 0.2$$
$$P(\text{弘前晴} \mid \text{青森雪}) = 0.3, P(\text{弘前雪} \mid \text{青森雪}) = 0.7$$

（練習）前出の通信の例で、受信信号を知っている時に、送信信号を知ることの平均情報量はいくらか。また、受信信号は送信信号に関してどれだけの情報を与えるか。

$$P(\text{送 }0) = 0.75, P(\text{送 }1) = 0.25$$

$$P(\text{受 }0| \text{ 送 }0) = 0.9, P(\text{受 }1| \text{ 送 }0) = 0.1$$
$$P(\text{受 }1| \text{ 送 }1) = 0.8, P(\text{受 }0| \text{ 送 }1) = 0.2$$

第5章　通信のモデルと符号化

5.1　通信のモデル

通信は、ある場所から他の場所へ情報を伝送することです。図 5.1 はシャノンによる**通信のモデル**です。送信側では、**情報源**から出力された**通報** (記号列) は、符号器で**符号列**に変換され、**通信路**に送られます。符号列は通信路を介して受信側に伝わりますが、**雑音**が加わるために、正確に受信側に伝わるとは限りません。受信側で受け取った符号列は復号器で通報 (記号列) に復元されます。

なるべく効率良く、しかもなるべく誤りが少なくなるように通報を伝えることが、通信に共通して求められる課題です。通信路の部分をディスクなどの記憶メディアと考えれば、この通信のモデルは記憶装置の書き込みと読み出しのモデルと見ることもできます。

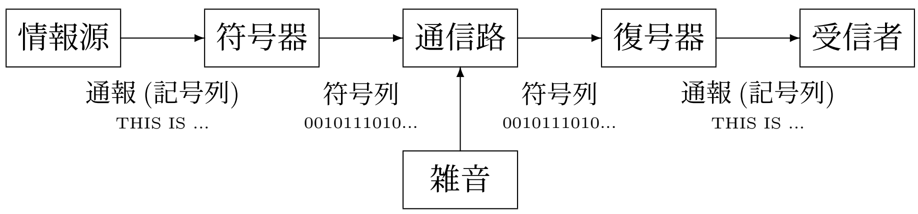


図 5.1: シャノンによる通信のモデル

シャノンは、英文で書かれた情報を「効率良く」「信頼性を保ちながら」通信する問題を通じて、情報の本質はその確率統計的側面にあることを示しました。「効率良く」という面に関しては情報源符号化問題、「信頼性を保ちながら」という面に関しては通信路符号化問題として扱われています。

5.2　情報源と符号化

サイコロを投げては、その結果を符号化して誰かに伝えることを考えてみましょう。この場合は、情報源は**情報源アルファベット**

$$A = \{1, 2, 3, 4, 5, 6\}$$

のいずれかの要素を、それぞれに対応する生起確率で (例えば各々$\frac{1}{6}$) 次々と出力すると考えることができます。

この通報を

$$B = \{0, 1\}$$

という**符号語アルファベット**からなる下表の符号 (2 元符号) で符号化するものとします。6 種類の記号のどれかは 3bit で表すことができます。

情報源記号	符号語
1	000
2	001
3	010
4	011
5	100
6	101

この場合、「3 6 2 3 ...」のような通報 (サイコロの目の数の並び) は「010101001010...」のように符号化され、受信者に送られます。これを受け取った受信者は、同じく上の表を見ながら、「3 6 2 3 ...」というサイコロの目の数の並びを復元することができます。サイコロの場合は、毎回出る目は独立で、このような情報源を**無記憶情報源**といいます。

別の例を考えましょう。情報源からは "THIS_IS..." のような英文文字列が通報として出力されるものとします。ここで、簡便のために、情報源アルファベットを

$$A = \{\_, \mathtt{A}, \mathtt{B}, ..., \mathtt{Z}\}$$

とします。ただし、'_' はスペースを表します。

表 5.1 はこれを符号化する 3 つの例を示したものです。符号 1(JIS コード) では、"THIS_IS..." という文字の並びは "0101010001001000010010010101010011..." と符号化されます。

情報源アルファベットの記号の数は 27 なので、そのどれであるかは 5bit で表すことができます ($2^5 = 32$)。これが符号 2 です。

符号 1 や符号 2 は符号語の長さ (符号語長) は固定で、このような符号を**固定長符号**といいます。符号 1 や符号 2 を用いた場合、受信者は受信した符号の並びを 8 または 5 ずつに区切って復号すれば、通報を復元することができます。

これに対し、符号 3 は符号語長が固定ではありません。このような符号を**可変長符号**といいます。符号化や復号の手順は複雑になりますが、データをより短い符号語に変換して効率良く伝えることが期待できます。

英文にどのような文字が多く現れるかを調査した結果によれば、'_' の頻度が最も高く、続いて 'E','T', 'A', ... とされています (表 5.1)。とすれば、頻度の高い情報源記号により短い符号語を割り当てることにより、平均的に短い符号にすることができると期待されます。符号 3 はこのような符号になっています[1]。

下表は、符号 1〜符号 3 で "THIS_IS" を符号化した場合の符号語列を示したものです。

	"THIS_IS" の符号語列	平均符号語長
符号 1(JIS コード)	01010100010010000100100101010011001000000100100101010011	8
符号 2	10100010000100110011000000100110011	5
符号 3	001011111100110100011001101	3.86

この例からは、効率良く通信や記憶を行うには符号 3 が有効と考えられます。この考え方で万事 OK と思われるかもしれませんが、統計情報 (各文字の生起確率) をどのように求めるかが極めて重要になってきます。文学作品から求めるのか、会話文から求めるのか、あるいはどの言語について求めるのか、そして「通信する文は果たしてその性質を反映したものかどうか」で、効果の程度は変わってきます。場合によっては逆効果 (長い符号語) となってしまうこともあります。

[1] このような符号をどのように構成するかについては、次節のハフマン符号を参照。

表 5.1: 符号の例

情報源記号	符号語 1(JIS コード)	符号語 2	符号語 3	確率
_	00100000	00000	000	.174
A	01000001	00001	0100	.067
B	01000010	00010	100111	.012
C	01000011	00011	11101	.023
D	01000100	00100	01101	.031
E	01000101	00101	101	.108
F	01000110	00110	11100	.024
G	01000111	00111	011000	.016
H	01001000	01000	1111	.043
I	01001001	01001	1100	.052
J	01001010	01010	0110011011	.001
K	01001011	01011	01100111	.003
L	01001100	01100	10010	.028
M	01001101	01101	001100	.021
N	01001110	01110	0111	.059
O	01001111	01111	0101	.066
P	01010000	10000	001111	.016
Q	01010001	10001	0110011010	.001
R	01010010	10010	1000	.056
S	01010011	10011	1101	.050
T	01010100	10100	0010	.087
U	01010101	10101	001101	.020
V	01010110	10110	0110010	.008
W	01010111	10111	100110	.013
X	01011000	11000	0110011001	.001
Y	01011001	11001	001110	.016
Z	01011010	11010	0110011000	.001

さて、この例のような自然言語は、サイコロの場合と異なり無記憶情報源ではないと考えられます。つまり、'T' や 'H' が独立に情報源から出力されるのではなく、'T' の次には 'H' や 'O' が出力される確率が高いと考えられます。このような情報源を**マルコフ情報源**といいます。情報源記号の生起確率が直前 1 文字のみに依存する情報源を**単純マルコフ情報源** (または 1 重マルコフ情報源)、直前 2 文字に依存する情報源を **2 重マルコフ情報源**といい、その性質を表す確率は $P(\cdot|\cdot)$ や $P(\cdot|\cdot\cdot)$ のような条件付確率で表されます。情報源の性質を良く表すモデルを用いることにより、更に効率の良い符号化が期待できます。

表 5.1 の確率を用いた単純マルコフ情報源の出力は次のようなものです。

```
PTEIEHSS_S_DFP_YDDD_PEIY_CM_OOE__G_AFOENIINSTANR__TRANNSSGTEEAMEBN_I...
```

（練習）上の表の符号語列を受け取ったものと考え、これを符号語ごとに区切って復号してみなさい。どの符号も、**一意復合可能** (uniquely decodable)、**瞬時復合可能** (instantaneously decodable) という重要な性質をもっています。

5.3 符号化法と符号語長

前節では、符号を適切に構成することにより1文字当たりの符号語長を短くできることを例で示しました。この符号語長をどこまで短くできるかについて、Shannonは**情報源符号化定理**を示しました。それは「1記号当たりの平均符号語長は4章に示した平均情報量$H(X)$にほぼ等しい長さで符号化できる」というものです(詳細略)。

情報源アルファベット$A = \{_, \mathtt{A}, \mathtt{B}, ..., \mathtt{Z}\}$が表5.1の確率で生起する場合、この情報源から生成される記号列の1文字当たりの平均情報量は$H(X) = 4.07$です(4章)。前節の“THIS_IS”を符号化する例では、符号3を用いると1文字当たり3.86bitで符号化することができました。これは都合の良い例を出したまでのことで、平均すると、この情報源から生成される記号の平均情報量4.07bitよりも効率良く符号化することはできません。しかし、実際のテキストは意味不明な文字の並びではないので、独立生起情報源よりももっと適切なモデル化ができるはずです。モデルを見直すことにより、もっと効率のよい符号化が期待できるのです。

Shannonの符号化定理は「... の長さで符号化できる」という可能性(あるいは目標値)を示したもので、そのような符号をどのように構成するかは別に考えなくてはいけません。

ハフマン(Huffman)が考案した最短符号の作り方(**ハフマン符号化法**)はその代表的なもので、その手順は以下のとおりです[2]。

(1) 情報源記号$a_1, a_2, ..., a_n$を確率の大きいものから順に並べる。

(2) 生起確率の最も小さい2個の情報源記号の一方に0、もう一方に1を割り当てる。これらをまとめて新しい情報源記号とみなし、その生起確率は2つの記号の生起確率の和とする。

(3) 情報源記号の数が1になるまで(1)と(2)の手続きを繰り返す。

以上のように割り当てた0と1を並べたものを符号語とします。

前出の情報源アルファベット$A = \{_, \mathtt{A}, \mathtt{B}, ..., \mathtt{Z}\}$の例にこの手順を適用すると次のようになります。

- 生起確率の最も小さい2個の情報源記号はJ，Q，X，Zですが、そのうちのJとQを取り上げ、Jに0、Qに1を割り当てる。そしてJとQの記号を削除し、代わりに「JQ」という新たな情報源記号を加える($P(\mathtt{JQ}) = 0.002$)。
- 生起確率の最も小さい2個の情報源記号を探すとXとZで、Xに0、Zに1を割り当てる。そしてXとZの記号を削除し、代わりに「XZ」という新たな情報源記号を加える($P(\mathtt{XZ}) = 0.002$)。
- 生起確率の最も小さい2個の情報源記号を探すとXZとJQで、XZに0、JQに1を割り当てる。そしてXZとJQの記号を削除し、代わりに「XZJQ」という新たな情報源記号を加える($P(\mathtt{XZJQ}) = 0.004$)。
- ...

このように、符号語はその末尾から徐々に決まっていき、最終的に表3.1の符号3が構成されます。

英文の平均情報量に関するある研究によれば、1文字当たりの平均情報量は1.3 bitと報告されています。これは、符号化方法を工夫すれば、英文テキストを6分の1程度に圧縮してファイルに格納したり通信することができることを意味しています。

[2]ファイル圧縮で用いられている圧縮プログラムに吉崎栄泰氏が開発したLHAがあるが、LHAの「H」や圧縮ファイルの拡張子LZHの「H」は、ハフマン符号化法を用いていることに由来し、他の符号化法と組み合わせることにより高い圧縮率を実現しています。