

2章「データ型」

(数値・文字・エンディアン・文字コード)

中島康彦

§2.0 情報とデータ

情報

- ▶ 次に何がくるかわからない
- ▶ 何がくることにより得られる確かさ=情報

情報の単位

- ▶ 二者択一の事象の1つが起れば, 1ビットの情報
- ▶ 四者択一の事象の1つが起れば, 2ビットの情報
- ▶ 生起確率が等しい Q 個の事象の1つが起こったことにより得られる情報は,
 $\log_2 Q$ ビット
- ▶ 事象 E の生起確率が $P(E)$ である場合, E が起こったことにより得られる情報は,
 $-\log_2 P(E)$ ビット

§2. 0 情報とデータ(続き)

データ

- ▶ 情報を計算機が処理可能な形にしたもの
- ▶ 記号による表現
 - 実世界の対象物をわかりやすく, 少ない量で表現できること.
 - 現実の計算機の上で, 効率良く処理できること.

情報の構造の明確化

- ▶ 「記号の性質および記号間の関係」の明確化に置き換えることができる.

§2. 1 計算機における情報の表し方

情報の最小単位:ビット

- ▶ 1ビット
 - 0または1の2通り
- ▶ 2ビット
 - 00 01 10 11の4通り
- ▶ 1バイト (8ビット)
 - $2^8=256$ 通り

§2. 1 計算機における情報の表し方(続き)

▶ 1ワード (16ビット)

□□□□□□□□□□□□□□□□

2の16乗=6万5536通り

▶ 1ワード (32ビット)

□□□□□□□□□□□□□□□□□□...

2の32乗=42億9496万7296通り

▶ 1ワード (64ビット)

□□□□□□□□□□□□□□□□□□□□□□...

2の64乗=1844京6744兆0737億0955万1616通り

§2. 2 基本データ型

符号無し整数/固定小数点数

unsigned integer/fixed-point number

▶ 8ビット長の場合

00000000	...	0
00000001	...	1
00000010	...	2
00000011	...	3
11111111	...	255

▶ 演算

$00000010(2) + 00000011(3) = 00000101(5)$

▶ オーバフロー/アンダフロー

$11111111(255) + 00000011(3) = 00000010(2)$

$00000010(2) - 00000011(3) = 11111111(255)$

§2.2 基本データ型(続き)

符号付き整数/固定小数点数

signed integer/fixed-point number

▶ 8ビット長の場合

10000000	...	-128
11111111	...	-1
00000000	...	0
00000001	...	+1
01111111	...	+127

▶ 演算

$11111111(-1) + 00000011(3) = 00000010(2)$

▶ オーバフロー/アンダフロー

$01111111(127) + 00000011(3) = 10000010(-126)$

$10000010(-126) - 00000011(3) = 01111111(127)$

§2.2 基本データ型(続き)

実数/浮動小数点数(IEEE754形式の場合)

real/floating-point number

▶ 単精度浮動小数点数 ... 32ビット長

符号1 指数8 仮数23

|-----|-----|
00111111100000000000000000000000

指数=0かつ仮数=0 ... 符号付ゼロ

指数=0かつ仮数≠0 ... 不正規化数

$(-1)^{\text{符号}} \times 2^{(-126)} \times 0.\text{仮数}$

$0 < \text{指数} < 255$... 正規化数

$(-1)^{\text{符号}} \times 2^{(\text{指数}-127)} \times 1.\text{仮数}$

指数=255かつ仮数=0 ... 符号付無限大(∞)

指数=255かつ仮数≠0 ... 非数値

例えば $0 \div 0$ の結果

§2.2 基本データ型(続き)

PI = 3.14159265358979323846

▶ 単精度浮動小数点数は3.141592まで正しい

0/1	00000000	000000000000000000000000	...	+/-0
0/1	01111100	000000000000000000000000	...	+/-0.125
0/1	01111101	000000000000000000000000	...	+/-0.25
0/1	01111110	000000000000000000000000	...	+/-0.5
0/1	01111111	000000000000000000000000	...	+/-1
0/1	10000000	000000000000000000000000	...	+/-2
0/1	10000000	100000000000000000000000	...	+/-3
0/1	10000000	10010010000111111011010	...	+/-3.14159250
0/1	10000000	10010010000111111011011	...	+/-3.14159274
0/1	10000000	10010010000111111011100	...	+/-3.14159297
0/1	10000000	111111111111111111111111	...	+/-3.99999976
0/1	10000001	000000000000000000000000	...	+/-4
0/1	10000010	000000000000000000000000	...	+/-8
0/1	11111111	000000000000000000000000	...	+/-∞
0/1	11111111	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	...	非数値

§2.2 基本データ型(続き)

▶ 倍精度浮動小数点数 ... 64ビット長

符号1 指数11 仮数52

$(-1)^{\text{符号}} \times 2^{\text{指数}-1023} \times 1.\text{仮数}$

▶ 拡張倍精度浮動小数点数 ... 80/128ビット長

符号1 指数15 仮数64

符号1 指数15 仮数112

$(-1)^{\text{符号}} \times 2^{\text{指数}-16383} \times 1.\text{仮数}$

実数/浮動小数点数(VAX形式)

実数/浮動小数点数(IBM形式)

§2.2 基本データ型(続き)

論理 logical number

▶ 1ビット長の場合

0 ... 偽(FALSE)

1 ... 真(TRUE)

▶ 8ビット長の場合

00000000 ... 偽(FALSE)

その他 ... 真(TRUE)

▶ 各ビットを独立論理値とした場合のビット毎演算

否定: .NOT.00001111 = 11110000

論理和: 00111100.OR.00001111 = 00111111

論理積: 00111100.AND.00001111 = 00001100

排他論理和:00111100.XOR.00001111 = 00110011

§2.2 基本データ型(続き)

文字 character

▶ null文字

00000000 ... ''

▶ 1バイト文字(半角文字)

00110010 ... '2'

▶ 2バイト文字(全角文字)

10100011,10110010 ... '2'

▶ 文字の演算(比較)が可能

00110010('2')<00110011('3')

10100100,10100010('あ')<10100100,11110011('ん')

§2.2 基本データ型(続き)

文字列 `string`

- ▶ 文字の並び+終端 (`null`) 文字
00110010,00110011,00000000 ... "23"
- ▶ 空の文字列
00000000 ... ""
- ▶ 文字列の演算(比較)が可能
"あい"<"あん"

§2.3 二種類のエンディアン

データの桁位置の高低

- ▶ 下位ビット ... 桁位置が低いビット
最下位ビット=Least Significant Bit
- ▶ 上位ビット ... 桁位置が高いビット
最上位ビット=Most Significant Bit

主記憶アドレスの高低

- ▶ 低アドレス ... 数値として小(0x00010000)
 - ▶ 高アドレス ... 数値として大(0xfffff000)
-

§2.3 二種類のエンディアン(続き)

▶ ビッグエンディアン ... IBM汎用機, SPARCなど

16進数 0x41424344 ... 文字 ABCD

桁の高低 ビット番号	MSB 0				LSB 31
レジスタの値	01000001	01000010	01000011	01000100	
バイト番号	0	1	2	3	

↓バイト番号順に格納

ビット番号	0	7	0	7	0	7	0	7
主記憶の値	01000001	01000010	01000011	01000100				
アドレス	0	1	2	3				
	↑先頭アドレス							

↓アドレス順に出力

ファイル	ABCD
------	------

§2.3 二種類のエンディアン(続き)

▶ リトルエンディアン ... Pentiumなど

16進数 0x41424344 ... 文字 ABCD

一桁の高低 ビット番号	MSB 31				LSB 0
レジスタの値	01000001	01000010	01000011	01000100	
バイト番号	3	2	1	0	

↓バイト番号順に格納

ビット番号	7	0	7	0	7	0	7	0
主記憶の値	01000001	01000010	01000011	01000100				
アドレス	3	2	1	0				
			先頭アドレス↑					

↓アドレス順に出力

ファイル	DCBA
------	------

§2.3 二種類のエンディアン(続き)

- ▶ エンディアンの異なる計算機間では、主記憶データに互換性がない。
C言語のread/writeなどにより、主記憶データを直接ファイルに格納したもの。
構造体をファイルに格納する場合は、個々の要素のエンディアンを考慮しなければならない。
ネットワーク間でデータを共有する場合にはエンディアンに関する取り決めが必要。
移植性の高いソフトウェアを開発する際も同様。

§2.3 文字コード

EBCDIC ... IBM汎用機

	0	1	2	3	4	5	6	7	8	9	A	B	C	(下位4ビット) D E F		
0	NUL	SOH	STX	ETX	PF	HT	LC	DEL	GE	RLF	SMM	VT	FF	CR	SO	SI
1	DLE	DC1	DC3	TM	RES	NL	BS	IL	CAN	EM	CC	CU1	IF'S	IGS	IRS	IUS
2	DS	SOS	FS		BYP	LF	ETB	ESC			SM	CU2		ENQ	ACK	BEL
3			SYN		PN	RS	UC	EOT				CU3	DC4	NAK		SUB
4	SP										¢	.	<	(+	
5	&										!	\$	*)	;	
6		/										,	%	_	>	?
7											:	#	@	'	=	"
8		a	b	c	d	e	f	g	h	i						
9		j	k	l	m	n	o	p	q	r						
A			s	t	u	v	w	x	y	z						
B																
C		A	B	C	D	E	F	G	H	I						
D		J	K	L	M	N	O	P	Q	R						
E			S	T	U	V	W	X	Y	Z						
F	0	1	2	3	4	5	6	7	8	9	LVM					EO

§2.4 文字コード(続き)

EBCDIK ... 片仮名に対応

	(下位4ビット)															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	PF	HT	LC	DEL	GE	RLF	SMM	VT	FF	CR	SO	SI
1	DLE	DC1	DC3	TM	RES	NL	BS	IL	CAN	EM	CC	CU1	IFS	IGS	IRS	IUS
2	DS	SOS	FS		BYP	LF	ETB	ESC			SM	CU2		ENQ	ACK	BEL
3			SYN		PN	RS	UC	EOT				CU3	DC4	NAK		SUB
4	SP	。	「	」	、	・	ヲ	ア	イ	ウ	[.	<	(+	
5	&	エ	オ	ヤ	ユ	ヨ	ッ		ー]	¥	*)	;	
6		/									^	,	%	_	>	?
7											:	#	@	'	=	"
8		ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ		サ	シ	ス	セ
9		ソ	タ	チ	ツ	テ	ト	ナ	ニ	ヌ	ネ	ノ		ハ	ヒ	フ
A			ヘ	ホ	マ	ミ	ム	メ	モ	ヤ	ユ		ヨ	ラ	リ	ル
B											レ	ロ	ワ	ン	ゝ	ゞ
C	{	A	B	C	D	E	F	G	H	I						
D	}	J	K	L	M	N	O	P	Q	R						
E	\$		S	T	U	V	W	X	Y	Z						
F	0	1	2	3	4	5	6	7	8	9	LVM					EO

§2.4 文字コード(続き)

ASCII

(American national Standard Code for Information Interchange)

	(下位4ビット)															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	NL	VT	NP	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SP	!	"	#	\$	%	&	'	()	*	+	,		.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}		DEL

§2.4 文字コード(続き)

JIS X0201 ローマ字

	(下位4ビット)															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	NL	VT	NP	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SP	!	"	#	\$	%	&	'	()	*	+	,	.	/	
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[¥]	^	_
6	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	—	DEL

§2.4 文字コード(続き)

制御コード

	(下位4ビット)															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	NL	VT	NP	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SP															
7																DEL

- NUL : NULL
- STX : Start of TeXt
- EOT : End Of Transmission
- ACK : ACKnowledge
- BS : BackSpace
- NL : New Line
- NP : New Page
- SO : Shift Out
- DLE : Data Link Escape
- NAK : Negative Acknowledge
- ETB : End of Transmission Block
- EM : End of Medium
- ESC : ESCape
- GS : Group Separator
- US : Unit Separator
- DEL : DElete
- SOH : Start Of Heading
- ETX : End of TeXt
- ENQ : ENQuiry
- BEL : BELl
- HT : Horizontal Tab
- VT : Vertical Tab
- CR : Carriage Return
- SI : Shift In
- DCn : Device Control n
- SYN : SYNchronous idle
- CAN : CANcel
- SUB : SUBstitute
- FS : File Separator
- RS : Record Separator
- SP : SPace

§2.4 文字コード(続き)

JIS X0201 片仮名

▶ いわゆる半角片仮名, 使用は避けるべき

(下位4ビット)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		。	「	」	、	・	ヲ	ア	イ	ウ	エ	オ	ヤ	ユ	ヨ	ッ
B	ー	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス	セ	ソ
C	タ	チ	ツ	テ	ト	ナ	ニ	ヌ	ネ	ノ	ハ	ヒ	フ	ヘ	ホ	マ
D	ミ	ム	メ	モ	ヤ	ユ	ヨ	ラ	リ	ル	レ	ロ	ワ	ン	、	。

§2.4 文字コード(続き)

JIS C6226-1978(旧JIS)...日本規格協会

「第1水準」2965字, 「第2水準」3384字, 「非漢字」453字

◆□■△▲▽▼※〒→←↑↓≡

0123456789 ABCDEFGHIJKLMN

ああいいうええおおかがぎくくげげこごさざしじすずせそそただちぢつづてとどなにぬねの
 アアイウウエエオオカガキクケケココサザシスズセゼソタダチツツツテトドナニヌネノ
 ΑΒΓΔΕΖΗΘΙ ΚΑΜΝΞΟΠΡΣΤΤΦΧΨΩ αβγδεζηθικλμνξ
 АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ

亜啞娃阿哀愛挨始逢葵茜穂悪握渥旭葦蔞粹庠翰扱宛妲妲蛇饒綉紉或粟裕安庵按暗案闇鞍杏以伊位依偉
 院陰隱韻吋右宇烏羽迂兩卯鵝窺丑確白濁嘔唄鬱蔚縵婁既蒲瓜閨噴云運雲苙餌馭宮嬰影映曳采永泳洩瑛盈
 押旺橫歐歐王翁禿鶯鷗黃岡冲萩億屋憶臆桶牡乙掩卸恩温穩音下化佻何伽伽佳加可嘉夏嫁家寡科暇果架歌
 魁晦械海灰界皆餘芥蟹開階貝凱劬外咳害崖慨概滉碍益銜該鎧骸涅馨蛙垣柝蠟鈎劃嚇各廓擴格核穀獲確
 粥刈苻瓦乾侃冠寒刊勣勣卷喚堪姦完官寬干幹恩感憤憾換敢柑桓棺款歎汗漢澗濯環甘監看竿管簡緩缶翰肝
 機婦毅氣汽畿祈季稀紀徽規記貴起勳輝飢騎鬼龜偽儀妓宜戲技擬欺儀疑祇義蟻誼議掬鞠吉吃喫桔橘詰砧
 供俠儔兇競共凶協匡卿叫喬境峽強滄怯恐恭挾教矯況狂狹矯胸脅輿鄉鏡響響驚仰凝蕘曉業局曲極玉桐秆
 掘窟咎靴懲鑿熊隈桑乘綵桑鞞動君黨訓群郡卦袞祁係傾刑兄啓圭挂型契形徑患慶慧憩揭携敬景桂溪哇稽
 檢權牽犬獻研硯緝具肩見謙賢軒運鏗險黠驗諭元原嚴幻弦滅源玄現絃絃言諺限乎個古呼固姑孤己庫弧戶故

§2.4 文字コード(続き)

ISO 10646

JIS X0221-1995(UCS)...日本規格協会

- ▶ UCS-4 : 1文字を4バイトで表現
上位2バイトが0の範囲を
BMP(Basic Multilingual Plane)と呼ぶ.
- ▶ UCS-2 : BMPの上位2バイトを省いた2バイトコード
これがいわゆるUnicode

§2.5 日本語コード

シフトJISコード ... Microsoft, アスキー

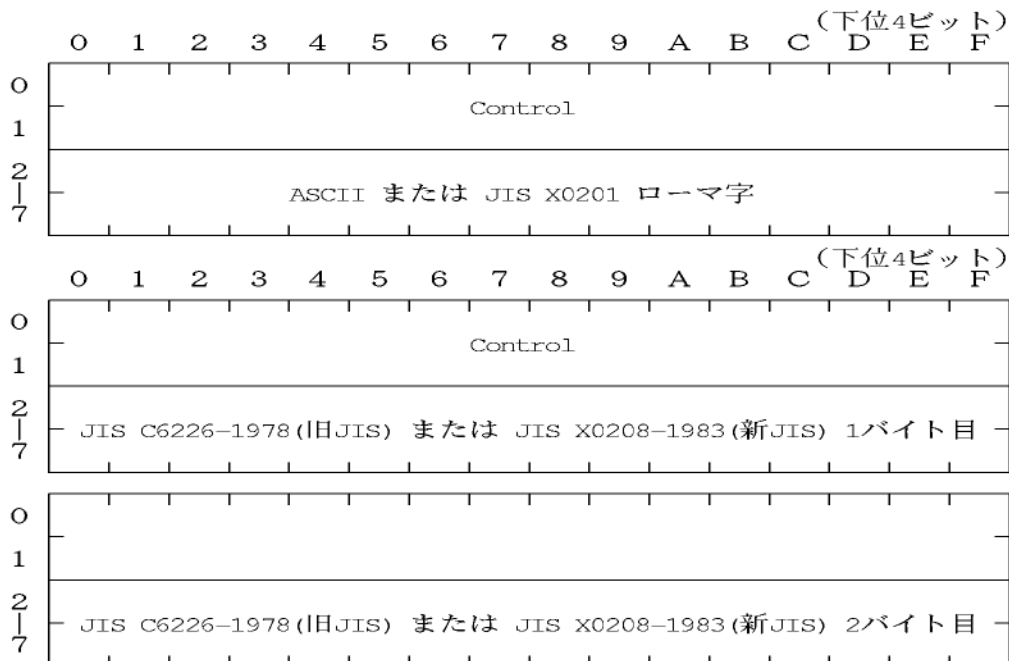
- ▶ 特別な遷移コードがない代わりに字数が少ない

	(下位4ビット)															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	Control															
1																
2	ASCII または JIS X0201 ローマ字															
7																
8	シフトJIS 1バイト目															
9																
A		。	「	」	、	・	ヲ	ァ	ィ	ゥ	ェ	ォ	ャ	ュ	ョ	ッ
B	ー	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス	セ	ソ
C	タ	チ	ツ	テ	ト	ナ	ニ	ヌ	ネ	ノ	ハ	ヒ	フ	ヘ	ホ	マ
D	ミ	ム	メ	モ	ヤ	ユ	ヨ	ラ	リ	ル	レ	ロ	ワ	ン	ゝ	ゞ
E	シフトJIS 1バイト目															
F																
0																
3																
4	シフトJIS 2バイト目															
F																

§2.5 日本語コード(続き)

ISO-2022-JP(いわゆるJISコード)

- ▶ 7ビットの文字列で漢字を表現できる



§2.5 日本語コード(続き)

- ▶ 2バイト文字への遷移コード

JIS C6226-1978(旧JIS)へは, 1B2440 ESC\$@

JIS X0208-1983(新JIS)へは, 1B2442 ESC\$B

- ▶ 1バイト文字への遷移コード

JIS X0201-1976 ローマ字へは, 1B284A ESC (J

ASCIIへは, 1B2842 ESC (B

ISO-2022-JPにはJIS X0201 片仮名が無い。

- ▶ JIS7方式 ... 20~5Fに割り当てる。

1B2849 ESC (Iを片仮名開始とする方法

0E/0F(ShiftOut/In)を片仮名開始/終了とする方法

- ▶ JIS8方式 ... A1~DFに割り当てる。

§2.5 日本語コード(続き)

NEC漢字コード

▶ 独自の遷移コード 1B4B/1B48(KanjiIn/Out)

	(下位4ビット)															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	Control															
1	Control															
2	ASCII															
7	ASCII															
8	独自野線・記号															
9	独自野線・記号															
A		。	「	」	、	・	ヲ	ア	イ	ウ	エ	オ	ヤ	ユ	ヨ	ッ
B	ー	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス	セ	ソ
C	タ	チ	ツ	テ	ト	ナ	ニ	ヌ	ネ	ノ	ハ	ヒ	フ	ヘ	ホ	マ
D	ミ	ム	メ	モ	ヤ	ユ	ヨ	ラ	リ	ル	レ	ロ	ワ	ン	。	。
E	独自野線・記号															
F	独自野線・記号															

	(下位4ビット)															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2	JIS C6226-1978 (旧 J I S) 1バイト目 2バイト目															
7	JIS C6226-1978 (旧 J I S) 1バイト目 2バイト目															
8																
F																

§2.5 日本語コード(続き)

EUC(Extended Unix Code)

▶ 重複する場合, シフトJISと区別できない

	(下位4ビット)															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	Control															
1	Control															
2	ASCII または JIS X0201 ローマ字															
7	ASCII または JIS X0201 ローマ字															
8																
9																
A	JIS C6226-1978(旧JIS) または JIS X0208-1983(新JIS)															
	JIS X0201 片仮名 … 8Eを前置した2バイト表現															
F	JIS X0212-1990(補助漢字)… 8Fを前置した3バイト表現															

§2. 5 日本語コード(続き)

UTF-8

- ▶ UCS-2に基づく1~3バイトの文字コード
- ▶ 0000~007Fの場合
下1バイトを使用
- ▶ 0080~07FFの場合(00000xxxxxyyyyyy)
110xxxxを1バイト目, 10yyyyyを2バイト目とする
- ▶ 0800~FFFFの場合(xxxxyyyyyyzzzzzz)
1110xxxxを1バイト目, 10yyyyyを2バイト目, 10zzzzzzを3バイト目とする

今日はここまで