

数の表現

整数の表現 (2進法)

- 0と1で整数を表したい
 - 2ビットを使うと 2^2 通りの違う値を表すことができる
 - $2^2 = 4$ 通りのパターンと整数0, 1, 2, 3をどう対応させるか？
- 最も自然な対応  2進法 (2を基数としてその冪で整数を表現)
 - 10進表現 : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, ...
 - 2進表現 : 0, 1, 10, 11, 100, ...

10進法	2進法	2ビット表現
0	0	00
1	1	01
2	10	10
3	11	11

- $1 + 1$ が自然に演算できることを確認する
- 2ビット表現で $01 + 01$ を考える
- 筆算を行ってみる

$$\begin{array}{r} 01 \\ +) 01 \\ \hline 10 \end{array}$$

10進法	2ビット表現
0_{10}	00
1_{10}	01
2_{10}	11
3_{10}	10

違う**符号化**で $1 + 1$ をしてみる

$$\begin{array}{r} 01 \\ +) 01 \\ \hline 10 \end{array}$$

$10 = 3_{10} \neq 1_{10}$

コンピュータ上での整数表現

- コンピュータには扱いやすいビット数がある
- バイト (byte) : 複数のビットがあつまった情報の単位
 - 現在では 1 byte = 8 ビット として良い
- バイト単位での処理  8 の倍数のビット数

型	バイト数	最小値	最大値
int8 (char)	1	-128	127
uint8 (unsigned char)	1	0	255
int16 (short)	2	2^{15}	$2^{15} - 1$
uint16 (unsigned short)	2	0	$2^{16} - 1$
int32 (int)	4	2^{31}	$2^{31} - 1$
uint32 (unsigned int)	4	0	$2^{32} - 1$
int64 (long)	8	2^{63}	$2^{63} - 1$
uint64 (unsigned long)	8	0	$2^{64} - 1$

小数点の表現

- 固定小数点 (fixed-point number)
 - 小数点の位置を固定
 - 例えば上位8ビットは整数部, 下位8ビットは小数部
 - 演算の扱いは整数の場合と同じ
- 浮動小数点 (floating point number)
 - 仮数部と指数という形で表す
 - 1.055×10^{-34} [Js], 2.998×10^8 [m/s], 9.109×10^{-31} [kg], etc.

浮動小数点

- 仮数
 - 符号
 - 仮数の絶対値
- 基数
- 指数

$$(\text{符号})(\text{仮数}) \times (\text{基数})^{(\text{指数})}$$

型	符号ビット数	指数部ビット数	仮数部ビット数	合計ビット数
半精度, float16	1	5	10	16 (2 bytes)
単精度, float32	1	8	23	32 (4 bytes)
倍精度, float64	1	11	52	64 (8 bytes)

小数表現での注意点

- 簡単な計算でも近似になるケースがある
- 例：10進法
 - $1/3 = 0.\underline{3}3333333\dots$
 - $1/5 = 0.2$
- 例：2進法
 - $1/11 = 0.\underline{01}010101\dots\dots$
 - $1/101 = 0.\underline{0011}0011\dots\dots$

循環小数を有限桁で打ち切るため誤差が生じる