7 データの取扱い

7.1 コンピュータが扱うデータ

7.2 数値計算における誤差

7.3 符号

7.4 データの圧縮

7.5 暗号

7.1 コンピュータが扱うデータ

整数： kビット固定、２の補数表現

- $k = 4$のとき、

0111	$2^3 - 1$
$\vdots$	$\vdots$
0000	0
1111	-1
$\vdots$	$\vdots$
1000	-2^3

浮動小数： 仮数 $\times 10^{\text{指数}}$ のような方式

- 32ビット浮動小数表現:

31	30 ⋯ 23	22 ⋯ 0
符	指数	仮数

$\pm 0.\boxed{\text{仮数}} \times 2^{\boxed{\text{指数}}-127}$ を表す。ここで、

- 符号：**0**が正、**1**が負
- 仮数：左のビットから順に重み $1/2, 1/4, \ldots$
- 指数：符号なし整数(**8**ビット)で、$127 = 2^{8-1} - 1$

- 丸め誤差：0.1を10000回加算しても1000にならない

文字,文字列： **ASCII**コード、**UTF-8**コード など

アナログデータ： 音声、画像、動画: 離散化が必要

- 標本化: 時間軸の離散化
- 量子化: 大きさ軸の離散化

圧縮の種類 可逆圧縮(**LZ**,連長圧縮)、非可逆圧縮(**jpeg**)

7.2 数値計算における誤差

桁落ち： 引き算などで、有効数字が減る場合あり

$$
\begin{array}{rr}
 & 123.4567 \\
- & 123.4566 \\
\hline
 & 0.0001
\end{array}
$$

有効数字が6桁減る

情報落ち： 加減算で小さい数字の桁が損失

$$
\begin{array}{rl}
 & 123.4567 \\
+ & \quad\;\; 0.01234567 \\
\hline
 & 123.4444
\end{array}
$$

繰り返し計算すると大きな誤差になるが、小さい順で加えれば誤差はほとんどない

例： 分散 (ここで、m は $x_1, \ldots, x_n$ の平均)

- 桁落ちが誤差を大きくする例

$$\frac{1}{n}\left(\sum_{i=1}^{n} x_i^2\right) - m^2$$

- 桁落ちが誤差に影響しない例

$$\frac{1}{n}\sum_{i=1}^{n} (x_i - m)^2$$

近似式： 精度と計算量はトレードオフ

コラム： $\sum_{n=1}^{\infty} \frac{1}{n} = \infty$ の計算

$\sum_{n=1}^{2^k} \frac{1}{n} \geq 1 + \frac{k}{2}$ であることが知られているが、左辺を単精度（有効数字**8**桁程度）で計算すると k を大きくしても情報落ちのため **15.5** を超えられない

7.3 符号 (7.3.1 符号化)

符号化：アルファベットΓの文字列をアルファベットΣの文字列で表すこと（**1**対**1**の準同型写像$C : \Gamma^* \to \Sigma^*$）

- Cが準同型写像 $\overset{\text{def}}{\Longleftrightarrow}$

  $\forall a_1, \ldots, a_n \in \Gamma \;\; C(a_1 \cdots a_n) = C(a_1) \cdots C(a_n)$

- 準同型写像を定めるには、$\forall a \in \Gamma \;\; C(a)$の値を決めればよい

問**7.8**： $C(a) = 10, C(b) = 1$で定まる準同型写像Cは（たまたま）**1**対**1**であり、逆変換可能

- $C(ab) = C(a)C(b) = 101$
- $C(bba) = C(b)C(b)C(a) = 1110$
- $C^{-1}(101101) = abab$

符号： $C(\Gamma) = \{C(a) \mid a \in \Gamma\}$

- 前ページの問では、$\{10, 1\}$ が符号

接頭符号： 接頭条件（$X\Sigma^+ \cap X = \emptyset$）を満たす符号 X

ここで、$X\Sigma^+ = \{ww' \mid w \in X, w' \in \Sigma^+\}$

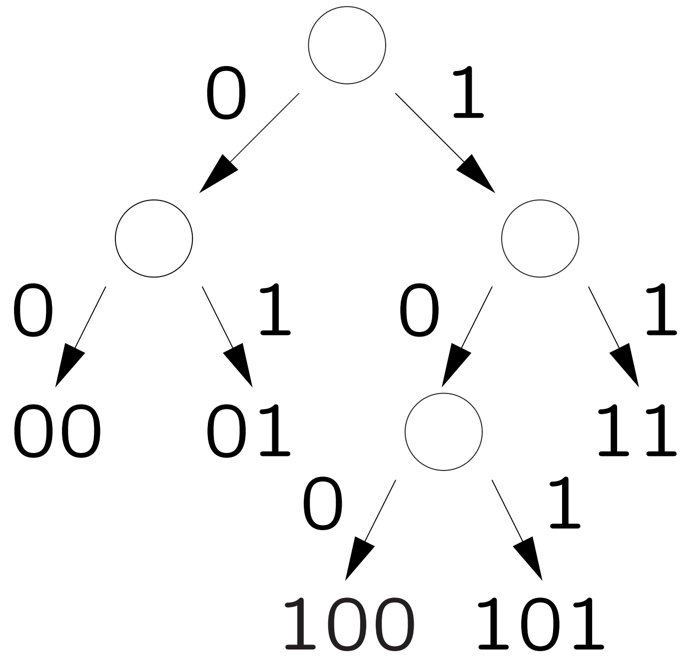
- 符号 $\{10, 1\}$ は接頭符号でない

問7.9： 接頭条件を満たす文字列の集合は符号であるか?

解： **YES.** X が接頭条件を満たすが符号でないと仮定

- $\exists x_1 \cdots x_n, y_1 \cdots y_m \in X \;\; x_1 \cdots x_n = y_1 \cdots y_m$ かつ ある i で $x_i \neq y_i$ (i を最小のものとする)
- x_i と y_i は接頭の関係であり、接頭条件を満たすことに矛盾する

- 木を利用した接頭符号の生成



符号$\{00, 01, 100, 101.11\}$

- 木を利用すると復号も容易

7.3 符号 (7.3.2 誤検出、訂正符号)

パリティ：１の個数が偶数になるよう、パリティビットを付け加える

$$0110111\mathbf{1}, \quad 0110011\mathbf{0}$$

ここで、青字がパリティビット

- １が奇数個のとき誤りを検知
- 偶数個の誤りは検知できない

ハミング符号： どの符号間も**3**以上のハミング距離をもつ符号

- ハミング距離: 異なるビットの数（以下の例では**3**）

  $$\begin{array}{l} 0001110 \\ \;\updownarrow\;\;\;\;\;\;\updownarrow\updownarrow \\ 0101101 \end{array}$$

- 1ビットの誤りなら訂正可能

  ハミング距離1の符号がひとつ定まるはず

6.4 データ圧縮　（6.4.1 ハフマン符号）

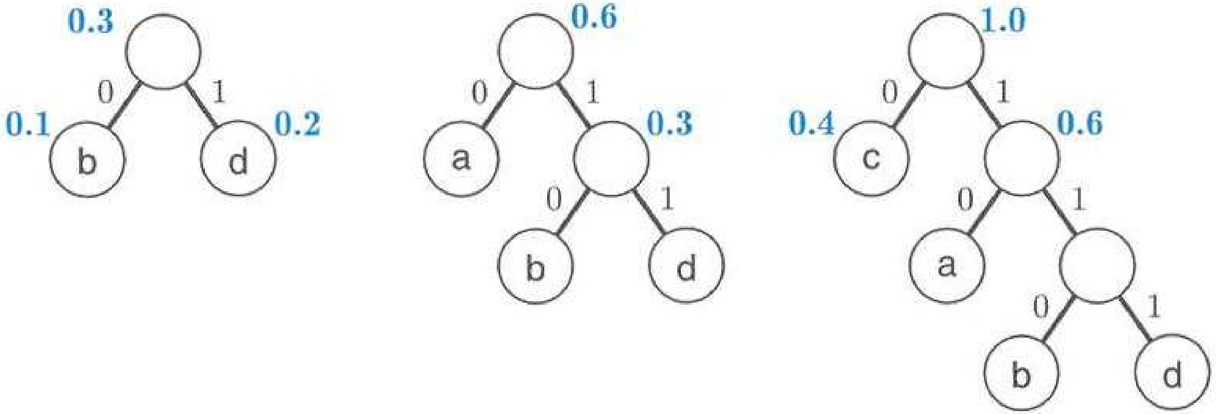
ハフマン符号： 圧縮したい文字列における(各文字の)生起確率の高い記号を短い符号に割り当てる方法

例： モールス符号

長さ	1	3	3	5	5	5	$\cdots$
符号	$\cdot$	$\cdot\cdot$	—	$\cdot\cdot\cdot$	—$\cdot$	$\cdot$—	
記号	e	i	t	s	c	n	

ハフマン符号の構成： 生起確率の小さい順に木を作成

記号 x	出現確率 p_x	ハフマン符号
a	0.3	10
b	0.2	110
c	0.4	0
d	0.1	111



符号長の期待値: $0.3\times2+0.2\times3+0.4\times1+0.1\times3 = 1.9$

問**7.18**： $abracatabra$ に対するハフマン符号を求めよ

6.4 データ圧縮　（6.4.2 情報量）

情報量： データがもつ本質的な情報の量：圧縮可能な最小のビット数

定理7.2： どんなnビットデータでも$n-1$ビット以下に可逆に圧縮するアルゴリズムは存在しない

証明 nビットデータは2^n種類あるが、$n-1$ビットデータの種類は真に小さいため

シャノンの情報量 基本的な考え方

- 事象の確率が高ければ、その事象の情報量は少ない
- 確率pの事象Pの情報量を$f(p)$と書く
- 独立事象P_1, P_2について、$f(p_1 \times p_2) = f(p_1) + f(p_2)$

情報量の定義： $f(p) = -\log_2 p$

定理7.3： 独立事象の確率$p_1, \ldots, p_m$について、その情報を表す符号を平均して

$$\sum_{i=1}^{m} (-p_i) \log_2 p_i$$

ビットより縮めることは出来ない

問7.19： 13ページの生起確率における情報量を求め、

ハフマン符号の平均長1.9と比較せよ

$$-p_a \log^2 p_a + \cdots + -p_d \log^2 p_d$$

$$= 1.847$$

6.5 暗号　（6.5.1 暗号の基本）

送信者： $\overset{\text{ひらぶん}}{\text{平文}} \xrightarrow{\text{暗号化}}$ 暗号

受信者： 暗号 $\xrightarrow{\text{復号}}$ 平文

- 暗号関数＝アルゴリズム ＋ 鍵
- 復号関数＝アルゴリズム ＋ 鍵

共通鍵暗号系： 暗号関数と復号関数で共通の秘密鍵を用いる

公開鍵暗号系： 一方の鍵を秘密とし、もう一方の（異なる）鍵を公開とする

Diffie-Hellman鍵交換： 秘密鍵を共有するためのプロトコルのひとつ

前提：（秘密でない）数xと素数pを**A**と**B**で共有

a: **A**の秘密の数（ナンス）、b: **B**の秘密の数（ナンス）

A	通信	**B**
$A := x^a \bmod p$	$\xrightarrow{A}$	
	$\xleftarrow{B}$	$B := x^b \bmod p$
$K_A := B^a \bmod p$		$K_B := A^a \bmod p$

ここで、鍵は、$K_A = K_B = x^{ab} \bmod p$。通信の盗聴者は、$a, b$を知らないため鍵を求めることは困難

6.5 暗号　（6.5.2 公開鍵暗号）

暗号 受信者の公開鍵で暗号化...受信者の秘密鍵でのみ復号可能

署名 送信者の秘密鍵で暗号化...送信者の公開鍵でのみ復号可能

RSA暗号系：　Rivest-Shamir-Adleman 提案の公開鍵暗号系

基礎事実：p, qを素数とするとき

$$\forall x, k \in \mathbb{N},\ \ x \equiv_{pq} x^{1+k(p-1)(q-1)}$$

ここで、$\equiv_{pq}$はmod pqのもとでの等価性を表す

基礎事実の証明は教科書**6.5.3**を見よ

鍵生成： 公開鍵(e,n)、秘密鍵(d,n)の生成

- 大きな素数p,qを選ぶ（$n=pq$とする）
- $de \equiv_{(p-1)(q-1)} 1$を満たす$d,e \in \mathbb{N}_+$を定める
- 基礎事実より$x \equiv_n x^{de} (= (x^d)^e = (x^e)^d)$
- $f(x) = x^e \bmod n$, $g(x) = x^d \bmod n$とさだめれば、

$$g(f(x)) = f(g(x)) = x \ (0 \le x < n)$$

p,qが知られるとeからdが計算できるので、鍵を生成したらp,qを忘れることが必要。基礎事実の証明は教科書**6.5.3**を見よ

素数の候補を探す：　フェルマーテストを使って素数候補を決める

pが素数ならば$2^{p-1} \equiv_p 1$

この他、いろいろな（決定的、あるいは、確率的）素数判定法がある

鍵の対を探す：拡張ユークリッド法

$mx + ny = \mathrm{gcd}(m, n)$を満たす$x, y$を計算する手法