第 3 章　情報の伝達と通信

3.1 情報の伝達と情報量

3.1.1 情報の伝達

☆情報の伝達は、**受け手側の状態の変化**に関係。手紙の物理的な移動は本質ではなく、
情報は、その運搬や伝達手段（メディア）とは独立なものである。

☆「情報を受け取った」と言えるのは

{ 自分に影響のある、これまで知らなかった事実を知った。
{ 何らかの判断材料にできる事実を知った。　　といった場合。

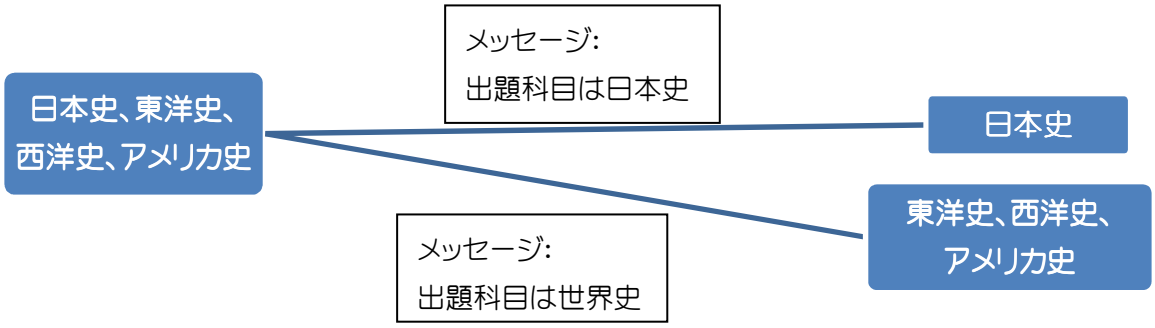
☆送り手から受け手へと伝えられる**メッセージ**の効果を**情報量**として表現。

3.1.2 情報の大きさ－情報量

教科書の例を用いると

a 場合の数の変化

例　歴史の試験･･･日本史、東洋史、西洋史、アメリカ史のどれか 1 つが出題される。



情報量をあらわす

案1　差　　定義:情報量＝事前の場合の数－事後の場合の数

問題点:100→97 と 4→1が同じ価値ということになってしまう　　却下

案2　商　　定義:情報量＝事前の場合の数÷事後の場合の数

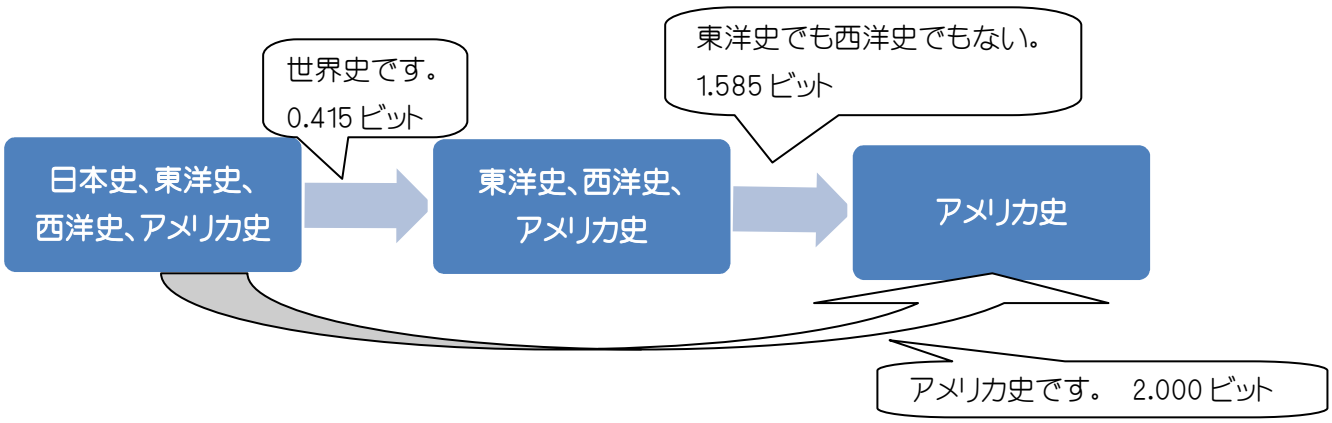
問題点:100÷97＝1.0309、4÷1＝4となり有用度の差を表現できるが、**情報の加法性**の要請を満たさない　　却下

＊情報の加法性

•情報を 1 度に受け取った場合（A）
　メッセージ A「アメリカ史を出題」･･･場合の数 4→1
•情報を分割して受け取った場合（B+C）
　メッセージ B「世界史を出題」･･･場合の数 4→3
　メッセージ C「東洋史と西洋史は出題しない」･･･場合の数 3→1
情報量（A）＝情報量（B）＋情報量（C）としたい。

案３　**商の対数**　　定義:**情報量＝$\log_2$(事前の場合の数 ÷ 事後の場合の数)**　　採用

- 単位:**ビット(bit)**
- 性質:**・場合の数が大きく減るほど、値は大きくなる。**
  - ・底2より二者択一の時(場合の数 2→1)の値は 1.0
  - ・情報の加法性を満たす　↓確認



b 確率による定義　　確率に基づく情報量の定義: **情報量＝$-\log_2$(確率)**

- 単位:**ビット(bit)**
- 性質:**確率が低いことを伝える情報量ほど大きい**

Ex)確率 1.0→情報量 0,　確率 0.5→情報量 1,　確率 0.25→情報量 2.0

(犬が人間を噛んだ＜人間が犬を噛んだ)

例　歴史の試験･･･日本史、世界史がどちらか出題される。出題確率は日本史 25%、世界史 75%

メッセージ:「日本史が出題される」「世界史が出題される」

―場合の数の変化はどちらも同じ(2→1)だが･･･

P(日本史)＋P(世界史)＝1、　P(日本史)＝$\frac{1}{4}$　P(世界史)＝$\frac{3}{4}$

メッセージ:「世界史が出題される」の情報量(単位ビット)は　$\log_2 \frac{3}{4}$

3.1.3 平均情報量

平均情報量＝**個々の情報量×メッセージが発せられる確率**

└“メッセージ全体”が持つ情報量、**エントロピー**とも呼ばれる。

メッセージ:「日本史」「東洋史」「西洋史」「アメリカ史」のどれか　の場合、確率はすべて $\frac{1}{4}$ より

平均情報量＝$\left\{\left(-\log_2\frac{1}{4}\right)\times\frac{1}{4}\right\}\times 4 \ =\log_2 4 \ =2$

☆等確率の状況では、平均情報量はここのメッセージの情報量と同じとなる

メッセージ:日本史、世界史がどちらか出題される。出題確率は日本史 25%、世界史 75% の場合

平均情報量＝$\left\{\left(-\log_2\frac{1}{4}\right)\times\frac{1}{4}\right\}+\left\{\left(-\log_2\frac{3}{4}\right)\times\frac{3}{4}\right\}$ ＝2×0.25＋0.415×0.75 ＝0.811

☆一般に2種類のメッセージの場合、片方の確立を**p** とすると他方は**1−p**

平均情報量はp＝0,1で最小（0ビット）となり、p＝0.5で最大（1ビット）

→どのメッセージがくるのか予測が全くつかない場合に、平均情報量は最大になる。

3.1.4 符号化と情報量

・符号化

現実の通信路には固有の伝達速度があるので、通信の効率を考えると、早く伝達するためには符号化によるデータの圧縮が重要となる。 以下、2進符号化を考える。

・データの圧縮

☆ i種類のメッセージm_i を長さl_i の符号で表す。

☆ メッセージm_iが確率p_iであらわれる場合、n個の記号を符号化した長さ（の期待値）は$\sum_i(np_i)\,l_i$
これをnで割って、**平均符号長**＝$\sum_i p_i\,l_i$ （単位ビット）

☆圧縮の問題とは、復号可能な符号化の中で平均符号長が短くなるものを探すこと

メッセージ:日本史、世界史がどちらか出題される。出題確率は日本史 25%、世界史 75% の場合

符号化すると、「日本史」→ 0、「世界史」→1 となり、平均符号長は 1

2年分の出題事情を符号化する

出題（1年目＋2年目）	確率（分母16）	符号	符号長
日本史＋日本史	分子1	111	3
日本史＋世界史	分子3	110	3
世界史＋日本史	分子3	10	2
世界史＋世界史	分子9	0	1

平均符号長＝$\frac{1}{2}\left(3\times\frac{1}{16}+3\times\frac{3}{16}+2\times\frac{3}{16}+1\times\frac{9}{16}\right)\fallingdotseq 0.844$

	平均符号長
1年分	1
2年分	0.844
3年分	0.823
4年分	0.818
平均情報量	0.811

☆情報源符号化定理

情報理論では、メッセージ集合の平均情報量＝平均符号長の下限 である。

この下限は、その情報量に等しい長さで符号化することで実現できる。

3.2 情報通信

3.2.1 実際の通信

3.2.2 プロトコル　(a)プロトコルの必要性　(b)アプリケーションとプロトコル

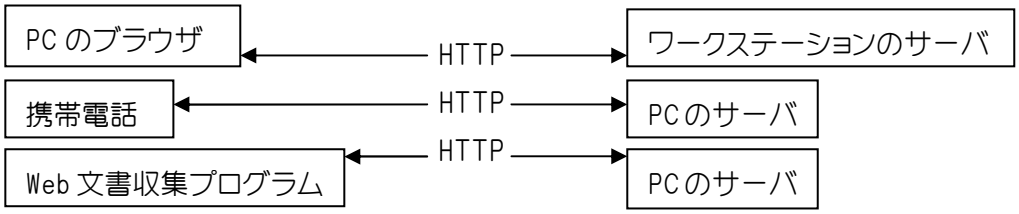
☆**プロトコル**…通信の際の決め事　　Ex)電話「もしもし」、トランシーバ「どうぞ」

☆コンピュータ同士の通信…人間の場合より厳密

Ex)　ウェブの閲覧では **HTTP** ,メールでは **SMTP** というプロトコルが使われる。

☆
- 情報を要求する側―**クライアント**
- 要求に応じて適切な情報を送り出す側―**サーバ**

☆プロトコルを正しく使えば機器によらず通信可能



3.2.3 通信の秘密と相手の認証　(a) 共通鍵暗号と公開鍵暗号

☆
- 暗号技術…盗聴を防ぐ
- ディジタル署名…認証や否認防止

☆
- **平文**…元のデータ　　**暗号文**…暗号化したデータ
- **暗号化**…平文から暗号文を作成すること　　**復号**…暗号文を平文にもどすこと
- **鍵**…暗号化や復号の際に用いられるデータ

☆
- **共通鍵暗号**…**暗号化と復号に同じ鍵を使う方式**、鍵の保存の仕方が重要
  Ex)初代「標準暗号」DES,AES
- **公開鍵暗号**…**暗号化と復号で違う鍵を使う方式、公開鍵と秘密鍵を用いる。**機密性は高まるが、暗号化にかかる時間が共通鍵暗号より長くかかってしまう。
  →ハイブリッド暗号(平文を共通鍵暗号で暗号化し、その鍵を公開鍵暗号で送る)の利用も。

3.4 インターネット

3.4.1 ネットワークの集合体と通信

☆**インターネット**…ネットワークの集合体、グループごとに管理、TCP/IP プロトコル使用

☆**ルータ**…ネットワーク間の通信を中継する

☆様々なプロトコル…役割毎に分割

3.4.2 階層プロトコル

すべての場合に備えたプロトコルを準備するのは不可能

→プロトコルを階層的に準備し、組み合わせて使用する。

・**TCP/IP 階層プロトコル**…共通の通信手順は同じプロトコルを使用、異なる部分だけ取り換え可能

TCP/IP モデル	主なプロトコル	主な役割
アプリケーション層	HTTP(www)　SMTP(メール)	アプリケーション間の通信
トランスポート層	TCP　UDP	1対1 の通信
インターネット層	IP	ネットワーク間通信
ネットワークインターフェース層	イーサネット　無線 LAN	ネットワーク内通信

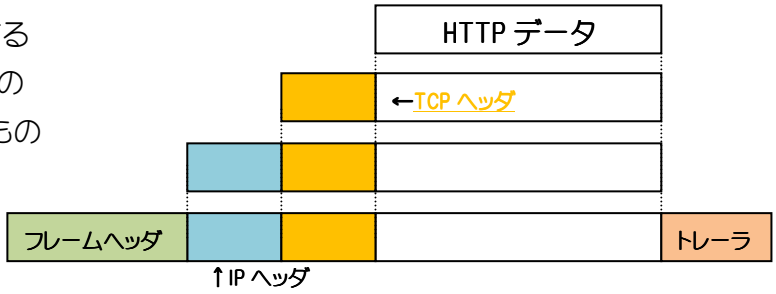
・**カプセル化**

☆階層毎に制御用のデータを付加する

- **ヘッダ**…先頭に付加されたもの
- **トレーラ**…末尾に付加されたもの

☆役割

- データの宛先
- 誤り訂正
- 順序の統制　など



3.4.3 IP アドレスとポート番号

(・TCP…相手との双方向な通信路を確保し、データの分割や分割されたデータ(**セグメント**)の順序通りの組み立てを行うプロトコル。セグメントの受信確認応答も行う。)

・**IP**…TCP によって分割され送りやすくなったデータを、宛先のコンピュータまで届けるプロトコル

・**IP アドレス**…インターネット内の住所

☆**32ビット**の数値、8ビット毎に記録　　Ex)172.16.38.100

☆インターネットに接続した機器は必ず一意の IP アドレスを持つ

・**ポート番号**…**16 ビット**の数値、コンピュータ内のアプリケーションを識別

(手紙に例えると、IP アドレス＝住所…東京都～区～番地、 ポート番号＝宛名…～様)

3.4.7 IP アドレスとホスト名の位置づけーDNS

・**ホスト名**…単なる数値であるため人間には扱いにくい IP アドレスに代わり、ユーザが目にする部分ではコンピュータの識別方法としてホスト名が使われる。　　Ex)www.u-tokyo.ac.jp

・**DNS**(Domain Name System)…ホスト名と IP アドレスを関連付ける仕組み　　TLD という。↑

・ホスト名は木構造であり、ホスト名に対応する IP アドレスを調べる場合、ルートサーバ(現在全世界で 13 種類稼働)を起点に**反復問合せ**を行い、問い合わせ結果はしばらく記憶される。

第 4 章 データの扱い

4.1 データモデル

4.1.1 データとデータモデル

☆**データ…**コンピュータの処理対象となる符号化された情報

☆**データモデル…**データを体系的に扱うためのモデル

4.3 代表的なデータモデルと演算

4.3.2 ネットワークモデル(「ウェブまで」)

☆ **グラフ…**ノードをエッジで結んだデータ(2 章参照)

ネットワークモデル…グラフのようにつながり方をあらわすモデル一般

路…順にたどっていけるエッジの列

オイラー路…すべてのエッジを重複なくたどる路　　Ex)ケーニヒスベルグの橋

☆**ウェブ**の各ページをノードと考えると、リンクはノードからノードを指す有向エッジ

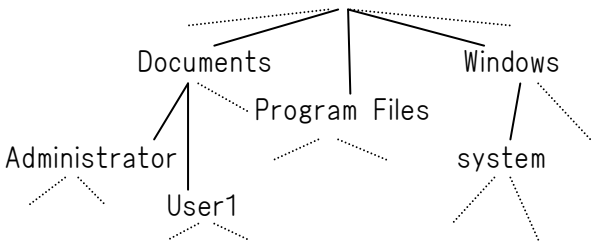
…ノードが有向エッジで結ばれたものもグラフという。

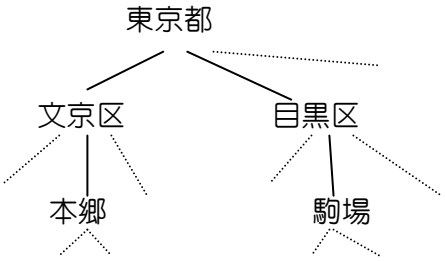
→グラフの構造に基づき、各ページの重要度を決めることが可能に(サーチエンジンで利用)

4.3.3 階層モデル(「住所の階層性」まで)

☆**階層モデル(木構造)…**分類の際にしばしば使われる枝分かれの構造

Ex)コンピュータのファイルシステム、住所の階層構造、コンピュータのドメイン名





第 5 章 計算の方法

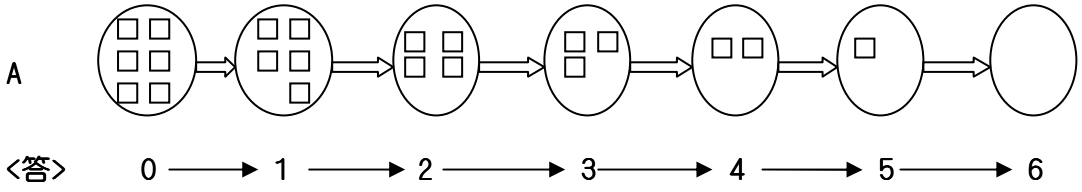
5.1 計算とその記述方法

5.1.2 計算の方法

計数(counting,ある集合 A の要素数を求める計算)を例にとる。

(a) **取り出し型…**要素を1つ1つ、指折り数えていくやり

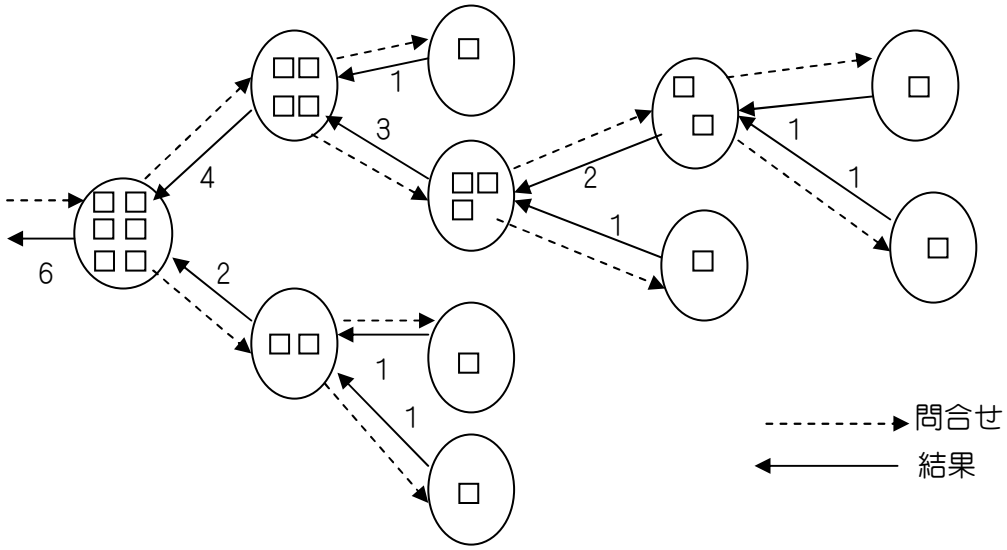
- 集合に対して用意されている処理
  - •空(要素が 1 つもない)かどうかを判定する
  - •要素を 1 つとりだす(集合の要素数は 1 だけ減る)
- 計算
  - •まず<答>を 0 にする
  - •A が 0 ではないあいだ、要素を 1 つ取り出す→<答>を 1 つ増やす、という処理を繰り返す



(b) **分割型…**自分の手に余る仕事を下請けに出していくやり方

- 集合に対して用意されている処理
  - •空か、要素が 1 つだけであるかを判定する
  - •空でない 2 つの集合に分割する
- 計算
  - •A が空なら<答>は 0、要素数が 1 なら<答>は1
  - •そうでなければ、A を B と C に分割(B,C ともに空集合でない)

<答>＝B の要素数＋C の要素数



5.1.2 計算の記述

☆ **変数**…いろいろに変化
変数名…以下の例では　<残り日数>
代入…変数に値を設定する操作　　以下代入の操作は <変数名>←式 と表示
逐次処理…計算処理の基本、書かれている順番に１つずつ順序よく処理
条件付き処理…操作の切り替え　以下

```
if 条件
 then “条件が成立した場合に行う処理”
 else “条件が成立しない場合に行う処理”
endif
```

とあらわす。
字下げ…対応する処理を縦に並ぶように書き、「まとまりの構造」をわかりやすく表す

例 八十八夜問題…立春(2 月 4 日)から数えて 88 日目は何月何日か？
(2 月 4 日の 87 日後を求める)

•<残り日数>＝4+87→2 月 91 日
91＞28(2 月の日数)
•<残り日数>＝91 －2月の日数→3月63日
63＞31(3月の日数)
•<残り日数>＝63 －3月の日数→4月32日
32＞30(4月の日数)
•<残り日数>＝32 －4月の日数→5月2日

2＜31(5月の日数)なので計算終了

```
<残り日数>←4+87
if <残り日数>＞28(2 月の日数)
 then <残り日数>←<残り日数>－2 月の日数
  if <残り日数>＞31(3 月の日数)
   then <残り日数>←<残り日数>-3 月の日数
    if <残り日数>＞30(4 月の日数)
     then <残り日数>←<残り日数>-4 月の日数
      if <残り日数>＞31(5 月の日数)
       then <6 月以降の処理>
       else “5 月”<残り日数>”日”と表示
      endif
     else “4 月”<残り日数>”日”と表示
    endif
   else “3 月”<残り日数>”日”と表示
  endif
 else “2 月”<残り日数>”日”と表示
endif
```

☆ **反復処理**…ある条件が成立している限り、指定された処理を繰り返して実行する

```
while 条件 do
  繰り返し実行する処理
done
```

配列…添え字をつけられる名前およびその値

配列 *daymonth*

添え字値→	1	2	3	4	……	11	12
要素値→	31	28	31	30	……	30	31

例: $daymonth_2$＝28 ••• 配列名＝*daymonth*、添え字＝2、値＝28

これらを用いて、”m(n月の日数)”という表現が何度も現れていることに着目し、
“(6 月以降の処理)”という部分を明確化させる。

改良版表記

```
<残り日数> ← 4＋87
m ← 2
while <残り日数> ＞ daymonth m do
    <残り日数> ← <残り日数> － daymonth m
    m ＝m＋1
done
```

“<残り日数>と月の日数の比較を繰り返し行う” 手順を、“反復処理”と“配列”を使ってすっきりと表記できた。

授業より補足　modulo の世界•••公開鍵暗号のあたり。。。

•modulo n＝「nを法として」•••nを法とする世界＝0～n-1までの整数しか存在しない
→計算の結果n以上になるときは、その数をnで割ったあまりの数を用いる。

Ex)modulo 133　　11^2＝121、 12^2＝(144÷133 の余りをとる)＝11

•公開鍵暗号の中で最も広く使われている RSA 方式
•••2 つの素数p、qを想定し、n＝p×qを計算、nを法とする

→この時任意の k^x ＝kになる （ただし 0≦k≦n－1、**x＝(p－1とq－1の公倍数)＋1**)