Introduction to
the New Information World

第2章

情報の表現
記号･符号化

情報表現の様々な側面(1)

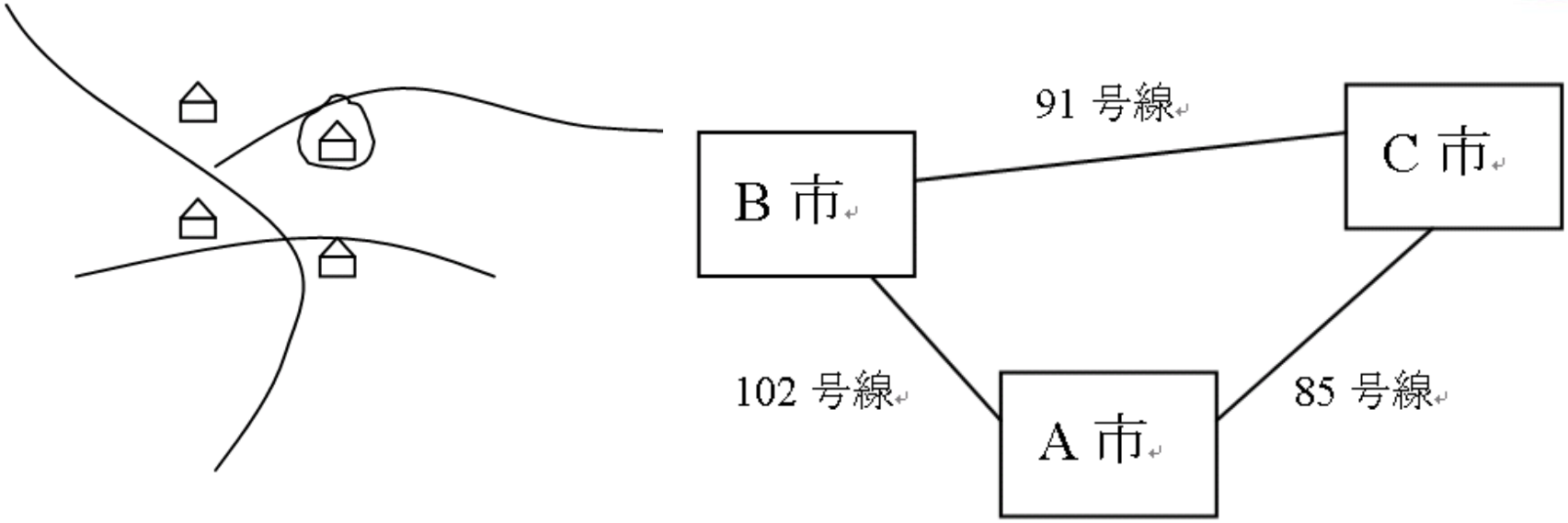
- 情報を表現する言語の違い
  - 自然言語
    - 人間が日常的に使用している言語
  - 人工言語
    - プログラミング言語のように人工的に作られた言語
- 情報の説明の仕方の違い
  - 手続き的表現
    - 時間をおった手順を説明
  - 宣言的表現
    - 対象間の関係や対象の属性を説明

情報表現の様々な側面(2)

↘ 情報の表現のされ方の違い

- 記号表現
  - 数学の記号，数式，方程式，論理式
    - 与えられた記号の集合と解釈するための規則体系
- パターン表現
  - 地図
    - 構成要素間の時空間パターン

パターン表現の例



- 日本語を理解しなくても，実際の世界と地図との対応関係が取れれば，目的地に行くことができる．
- 記号表現の例
  - まっすぐ進んで，次の角を左に曲がり．．．
  - 日本語という規則体系を知らないと，意味をなさない．

情報表現の様々な側面(3)

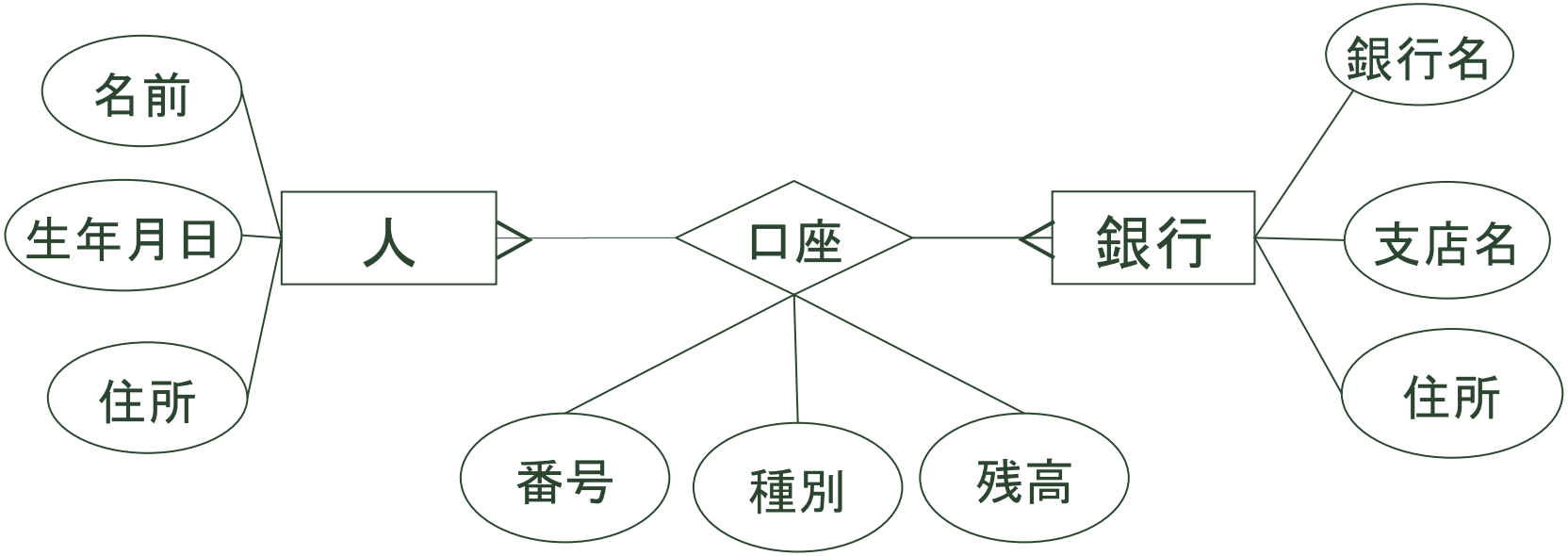
その他にも...

- ディジタル/アナログによる表現の違い
- 情報量からみる側面

モデルとモデル化

- モデル
  - 単純化/抽象化された事物/事象/概念
  - (ジェット旅客機設計の例)
    - 実際の旅客機をテストする前に，小型模型(モデル)を用いた風洞実験を行う.
- モデル化
  - 実際の事物/事象に対応したモデルを構築する過程

銀行，口座，人をモデル化した例

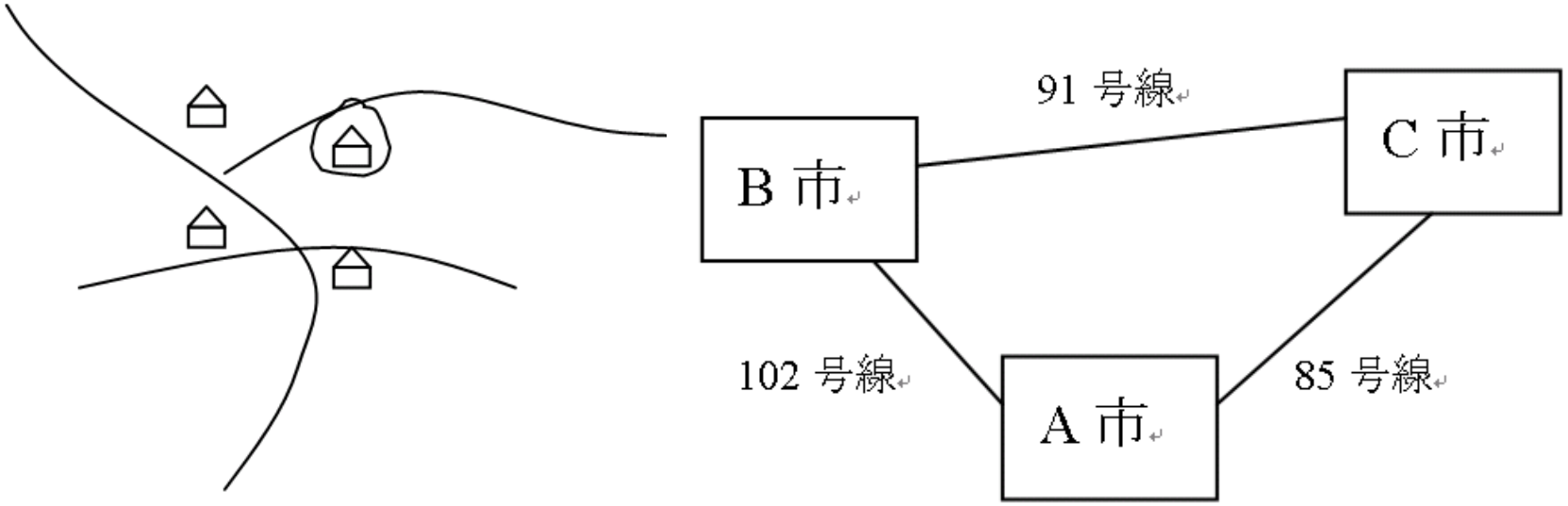


（実体関連図）

モデルの表現形式(1)

- 目的によって，構築するモデルの表現形式が異なる
  - (ジェット旅客機の例)
    - 風洞実験が目的:実機と同様の材料でモデル化
    - デザインが目的:加工/修正のしやすい材料でモデル化

モデルの表現形式(2)



↘ 目的別の表現形式

- 徒歩で移動する場合(左の図)
- 車で移動する場合(右の図)

本日の出席表(1)

- 地図の例以外で，同一対象を表現している記号的表現とパターン的表現の具体例を見つけ，それぞれの利点．欠点を，表現の目的に照らして考察せよ．
  - 表現している対象：
  - 記号的表現について：
    - 例
    - 表現の目的
    - 利点欠点
  - パターン的表現について：
    - ．．．．

モデルの表現形式の例(1)

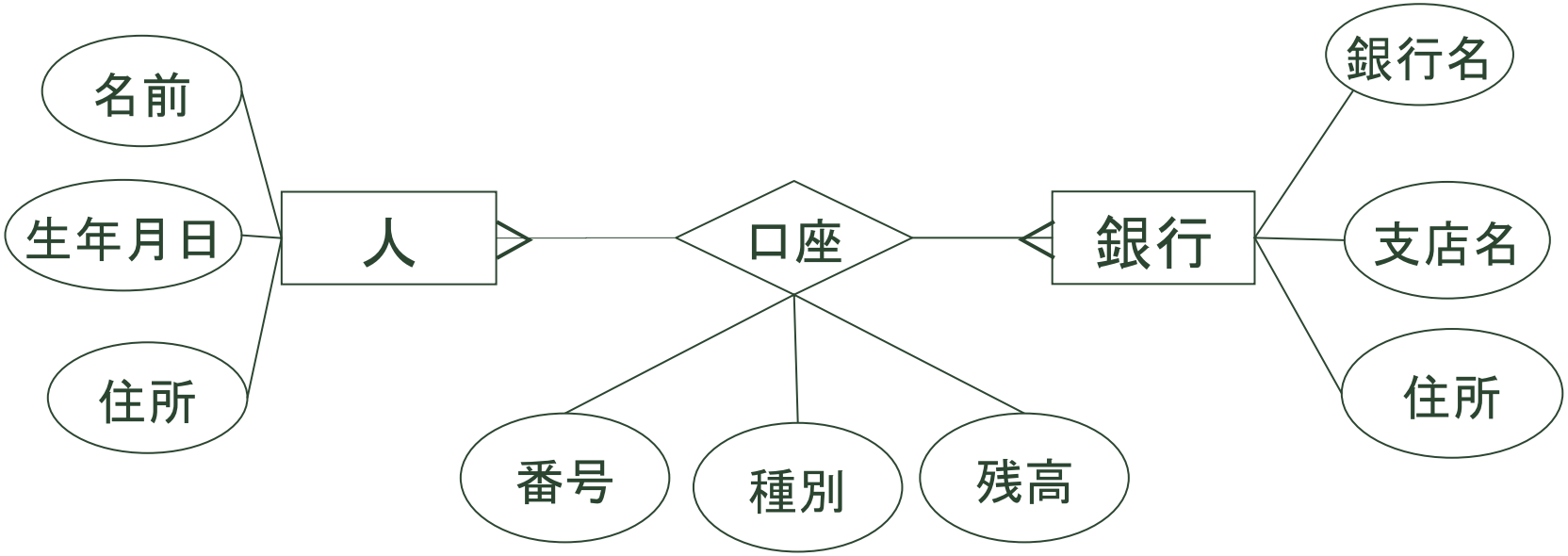
- 表(table)
  - こみいった事柄を整理できる
  - 歴史年表/賃借対照表/成績表など
  - 計算機上の表計算ソフトの利用も一般的
- 図
  - 何らかの目的で描いた2次元図形
  - 人間の思考･推論を支援/拡張する
  - 設計図/地図など
  - 広義には絵画/スケッチなども含める

モデルの表現形式の例(2)

グラフ

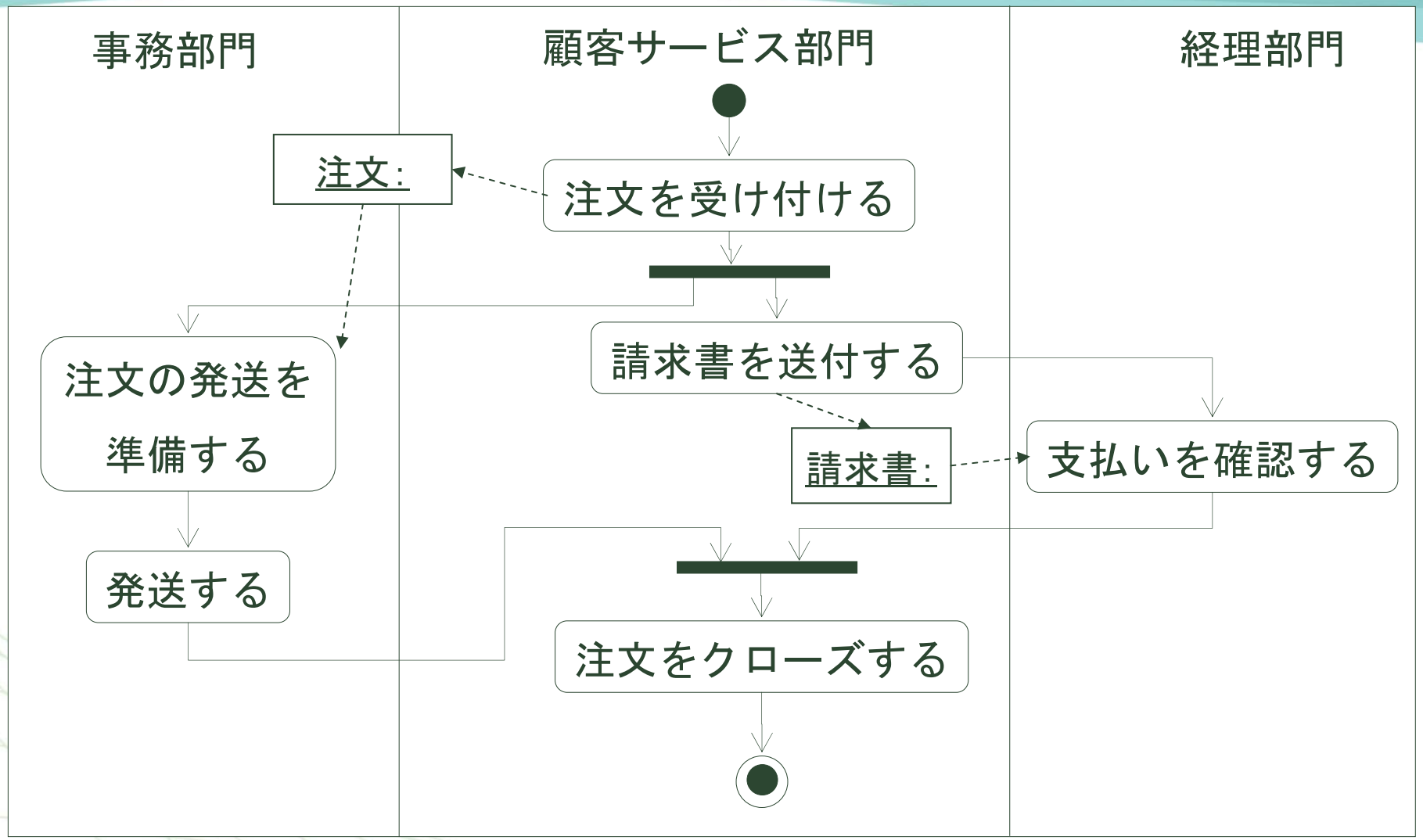
- ノード(node)とエッジ(edge)から構成される
- ラベル付きグラフ
  - ラベル付きのエッジで構成されるグラフ
- 有向エッジ，弧
  - 方向を持つエッジ
- 道路ネットワーク/組織図/pert図/意味ネットワークなど様々な領域で幅広く用いられる

グラフ表現の例(1)



（実体関連図）

グラフ表現の例(2)



（アクティビティ図）

情報表現で考慮すべきこと(1)

- 情報表現の受け手側
  - 情報表現を適切に理解･解釈しなければならない
- 情報表現のデザイン側
  - 目的に応じて適切な表現手段を選択しなければならない

これが難しい．．．

情報表現で考慮すべきこと(2)

- **表現の対象に関して**
  - 表現の対象となる事物/事象を明確にする必要がある
    - 表現の対象：物理的実体，抽象的アイディア，思考の方法，思考の結果など
- **表現の目的に関して**
  - 表現されている/する目的を理解する
    - 表現の目的:他者への伝達，依頼，自身のアイディアの整理，効率的な問題解決など
- **表現の方法に関して**
  - 表現に関わるコストや目的に照らして，よりよい方法を選択する必要がある

記号表現

- **記号表現**
  - 事物/事象,心的概念を抽象化したもの
- **記号表現の実際の形式**
  - 図記号(ピクトグラム), 数の表現など
- **記号が表す2側面**
  - 意味されるもの(シニフィエ)
  - 意味するもの(シニフィアン)

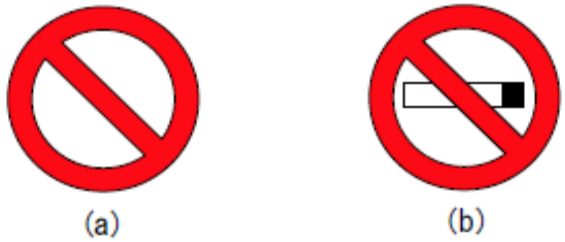
サービスエリアの図記号

- 抽象化された図形によるデザイン
  - 瞬時に表示内容を認識できる
- 記号表現とパターン表現の混在
  - パターン表現は常に具体的/直接的であればいいわけではない
  - サービスエリアの写真？？？

図記号の修辞法

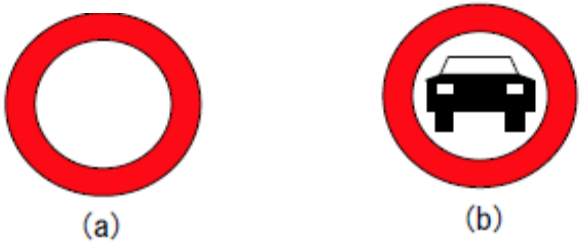
- 提喩に相当する表現方法
  - ある事物を表現するのに，それと意味的包含関係にある事物を代わりに用いる比喩
    - ナイフ，フォークの図でサービスエリアを表現する
    - 花見．．．この花は桜を表現している
- コンピュータのGUIにおけるアイコン
  - ゴミ箱アイコンは「ゴミを捨てる」という行為の隠喩として表現される
    - 隠喩の例：「雪の肌」「ばらのほほえみ」
- レトリック（修辞法の話題が出たので，脱線）
  - 換喩：赤頭巾ちゃん

交通標識の図表現(1)



- (a)車両通行禁止の標識(日本)
- (b)禁煙の標識(日本)
- 禁止や否定を表すために用いられる図記号(日本)

交通標識の図表現(2)



- (a)すべての車両通行禁止(欧州)
- (b)二輪車以外の車両通行禁止(欧州)
- 記号の恣意性
  - 記号表現と命題の対応付けは恣意的である
    - ○,×による表現が常に肯定，否定(禁止)に対応づけられるわけではない
  - 情報表現のデザイナーは受け手側の解釈の枠組みに注意を払う必要がある

本日の出席表(2)

- 命題(スローガン)をピクトグラム(図記号)でデザインしてみる．ただし，以下の事項に留意すること．
  - デザインと命題の対応
  - 簡潔さ
  - 視認のしやすさ
- 「次回の情報の授業は情報教育棟の大演習室でやります」
- (動物園で)「動物に餌をやらないでください．」
- (トンカツ屋さんで)「キャベツのおかわり自由」

日本語文字コード

- 文字と計算機上の符号(数値)を対応づけるための枠組み
- 現在，JIS,シフトJIS,EUCなどの異なった日本語文字コードが混在している
- 解釈の枠組みが異なれば記号の意味が異なってしまう例
  - プログラムが想定するコード体系と異なると，「文字化け」が起こる
- コード体系の標準化･統一化には困難も多い

数の表現の歴史

- インドでの発見
  - 数字のゼロ
  - 位取り表記法による算術演算
- アラビア数字表記法
  - 0から9の10種類の記号を用いる
  - 各記号が各桁に対応している
- ローマ数字表記法
  - Ⅰ,Ⅱ,Ⅲ, Ⅹ, Cなどの記号を用いる

位取りに基づいた計算

- **アラビア数字での1963＋41の計算**
  - 表記上の各桁を計算していけばよい
- **ローマ数字でのMCMLXⅢ＋XLⅠの計算**
  - 位取りに基づいた計算をすることが出来ない

情報表現間のトレードオフ

- アラビア数字表記
  - 筆算や位取りの観点からは優れている
- ローマ数字表記，漢数字表記
  - 表記された数字改ざん防止に優れている
    - アラビア数字表記「2030」は「1203000」のように改ざんされやすい
    - 漢数字表記「弐千参拾」は改ざんされにくい
- 情報表現のデザイナーは情報表現間のトレードオフを考慮する必要がある

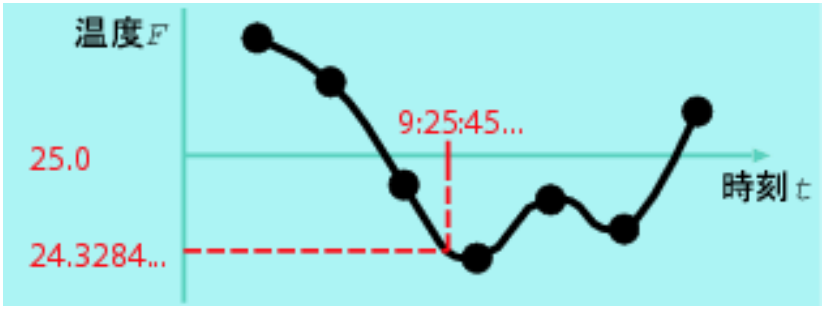
コンピュータでの数の表現

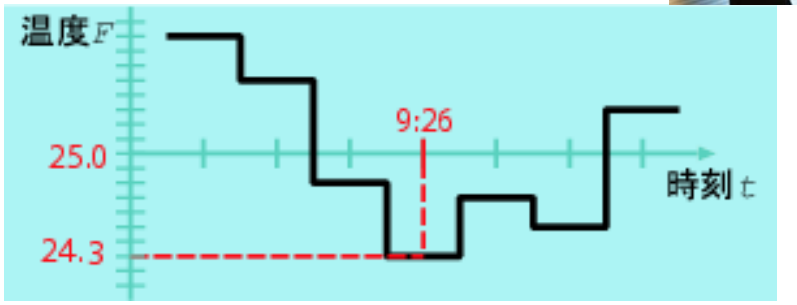
- 「0」と「1」の2種類の記号を用いたビット列で表現される
- 表現できる数値はコンピュータに依る
  - 表現できる正の整数
    - 16ビットのシステム：0～65,535（$2^{16}-1$）までを表現できる
    - 32ビットのシステム：0～4,294,967,295（$2^{32}-1$）を表現できる

アナログ表現とデジタル表現

- アナログ表現
  - ある情報を連続量(アナログ量)として表すこと
  - 無限の精度を必要とするため，データの複製は元のデータの近似にしかならない
- デジタル表現
  - ある情報を離散的に表すこと(デジタル量)
    - ある情報に対して一定の間隔の尺度を導入し，その尺度の値に近似して表現する
  - 複製時にデータが劣化しにくい
  - 情報コンテンツの著作権保護への問題をもたらす

アナログ表現とディジタル表現の実際





- (左の図)気温のアナログ表現
- (右の図)気温のディジタル表現
- アナログ量をディジタル量に変換する際には，情報を離散化する間隔を選択し，表現する必要がある
  - 量子化，標本化

量子化

- 連続量の情報をある間隔ごとの離散量として表現する作業
- 情報の用途によって間隔の詳細度を決める
  - コンピュータディスプレイ装置
    - 赤(R)緑(G)青(B)を混色したRGB形式を用いている
    - 各々256種類の異なる色で表現するとすると. . .
    - 256×256×256=16,777,216色を表示できる
  - 音楽CD
    - 量子化のために16ビットを用いるとすると. . .
    - 音の振幅を65536(2の16乗)個の段階に分割できる

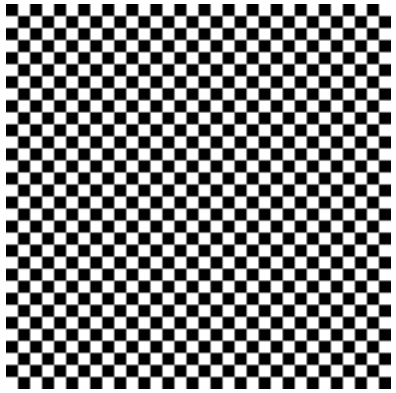
標本化

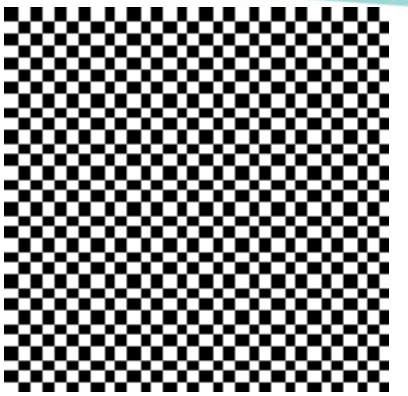
- 情報をある間隔(頻度)ごとに抽出すること
- 標本空間
  - 対象の情報が定義される時間や領域
- 標本化の間隔はデータの利用目的により変わる
  - 高い精度が求められるのか，荒くてもサイズが小さい方が良いのか．

標本化定理(シャノン)

- 情報の精度から必要な標本化の頻度を示す
- 標本化の対象となるアナログ量Fが周波数の異なる複数の周期関数の重ね合わせで表現できる事を基本にする
- 周期関数(周期T,周波数$\omega = 1 / T$)の周波数がW以下であるとすると，$1 / 2W$間隔で標本化すれば，元のアナログ関数Fを復元できる

エイリアシング





- 対象にナイキスト周波数より高い周波数の周期関数が含まれている場合に，誤った関数が復元される現象をエイリアシングと呼ぶ
  - (左の図)オリジナル画像
  - (右の図)エイリアシングが生じた例

標本化の実際

- **音楽CDの標本化**
  - 人間の鑑賞が目的なので，聴覚で知覚できない高い周波数まで記録する必要はない
  - 実際
    - 標本化の基準：44.1kHz
    - 1/44100=0.0000227秒間隔で音の情報を標本化
- **適切な細かい標本間隔を用いれば，アナログ量を欠損なくディジタル量に処理できる**

周期関数への分解

- フーリエ解析
  - 与えられた信号を個々の異なる周波数成分の波に分解
  - テキストp.26,図2.9
- 音声や画像などの情報表現と圧縮に用いる
  - (図)画像の低周波成分から高周波成分へと足し合わせていったもの．復元に必要な情報量が分かり，データ量を圧縮できる

デジタル符号化(1)

- 2進符号
  - 10進数を2進数に変換したもの
- ハミング距離
  - 2つの符号間で対応する桁の記号が異なる個数
    - (0000)と(0001)では1
    - (0011)と(0100)では3
  - 2進符号では数値の差とハミング距離が一致しない

デジタル符号化(2)

グレイ符号(テキストp.28.表2.2)

- 値が隣接する符号間のハミング距離を常に1とした符号
- 2進符号から作る事ができる
  - 最上位桁は2進符号と一致
  - 最上位桁以外では，対応する2進符号の桁とその左の桁が一致すれば0，異なっていれば1
- パタン生成や機械の制御コード，遺伝子の変異を模した計算などにも用いられる

ディジタル符号の圧縮

- ディジタル符号化された情報は圧縮できる利点を持つ
- 可逆圧縮
  - 圧縮したものから元の情報を完全に復元できる方法
- 非可逆圧縮
  - 元の情報には復元できない方法
  - 人間の知覚では差異が分からない程度の復元が可能ならば様々に応用が可能

ハフマン符号化

- 出現確率の大きな記号には短いビット列を割り当て，全体を少ない情報で表現する方法
  - (例)CABDABABAAを0,1のビット列に符号化
    - A:00,B:01,C:10,D:11と符号化すると20ビット
      - 10 00 01 11 00 01 00 01 00 00
    - A:0,B:10,C:110,D:111と符号化すると17ビット
      - 110 0 10 111 0 10 0 10 0 0

出現頻度によって，使用するビット数を変える

符号は可変長となるが，先頭から調べれば一意に解釈できる

ランレングス圧縮

- ビット列を値とその繰り返し回数で表す
  - (例)32ビットのビット列00001100000000111110001111110100 0
  - 0と1の繰り返しを数えると
    - 4回(0), 2回(1), 7回(0), 5回(1), 3回(0), 6回(1), 1回(0), 1回(1), 3回(0)となる
  - 繰り返し回数を並べる
    - 4 2 7 5 3 6 1 1 3
      →100 010 111 101 011 110 001 001 011
  - 27ビットに圧縮する事ができる
  - ファクシミリ通信などで使われている
    - 背景の白の部分が多いので, 大幅な圧縮が可能

JPEG圧縮

- 画像データの圧縮方法(非可逆圧縮)
- 要求される精度の周波数成分までを符号化する
  - 人間が必要としない高周波成分に対する情報を切り落とすことでデータ量を圧縮

符号の誤り検出･訂正

- **A，Bという情報を相手に伝えたい場合にノイズによりビットが1つ反転しうると仮定すると…**
  - A:0,B:1として符号化
    - 受け手側では誤りを検出する事ができない
  - A:00,B:11として符号化
    - 受信する可能性のある符号は4通り
    - 誤りを検出できる：01,10は誤りが生じたと分かる
  - A:000,B:111として符号化
    - 受信する可能性のある符号は8通り
    - 誤り検出と訂正ができる
      - 001,010,100はAを伝送しようとして誤ったもの
      - 110,101,011はBを伝送しようとして誤ったもの

ハミング距離と誤り検出･訂正

- 符号の1つのビットが反転すると，反転前後の符号のハミング距離は1となる
- 2つの符号に関して，
  - 誤り検出に必要な符号間のハミング距離：2以上
  - 誤り訂正に必要な符号間のハミング距離：3以上
- 一般にt個までの誤りに関して
  - 誤り検出に必要な符号間のハミング距離：t+1以上
  - 誤り訂正に必要な符号間のハミング距離：t+2以上

パリティ

- パリティ検査
  - 冗長なビットを追加することで符号の誤り検出や訂正を行う
  - nビットの符号(x1−x2−x3−‥‥−xn)に付加ビットyを追加して1となるビットが偶数になるようにする
    - $x1+x2+x3+\cdots+xn+y \equiv 0 \pmod 2$
- パリティ:元の符号で1となるビットの個数が偶数個か奇数個かを表す
- 単一パリティ検査符号:ビットを1つ追加してパリティを検査できるようにした符号

ハミング符号(1)

- 任意の符号間のハミング距離を3以上とすることで，誤り訂正を可能にする
  - (ハミング符号の例)
  - 4ビット符号(x1,x2,x3,x4)に3つのビットを付加した符号(x1,x2,x3,x4,y1,y2,y3)をつくる
  - y1,y2,y3は以下のようにする
    - x1+x2+x3+y1≡0(mod2)
    - x1+x2+x4+y2≡0(mod2)
    - x1+x3+x4+y3≡0(mod2)

ハミング符号(2)

- 受信した符号に対して以下を調べる
  - $z1 \equiv x1+x2+x3+y1 \pmod 2$
  - $z2 \equiv x1+x2+x4+y2 \pmod 2$
  - $z3 \equiv x1+x3+x4+y3 \pmod 2$
- 誤りがどこで起こったかをしる事ができる
- 符号間のハミング距離を3以上とすることで1つの誤りに対する訂正までを可能にする