情報の構造とデータ処理

水谷 正大

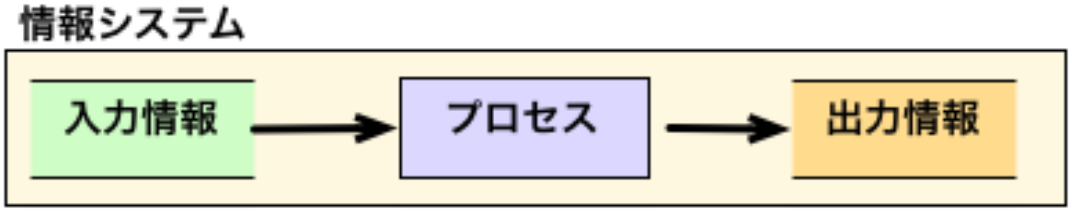
大東文化大学 mizutani@ic.daito.ac.jp

2014

目　次

情報システム（information system）

与えた入力情報（input）に対して特定の目的を達成するための加工処理をするプロセス（process）によって出力情報（output）を取り出すために必要な装置（ハードウエア）やプログラム（ソフトウエア）などの全体（かならずしも機械である必要はない）。



演習：次の情報システムの入力、プロセス、出力は何か？データベースとしてはどうか？

- **インフルエンザウイルス、細菌**
- **手帳、家族、友人**
- **ラーメン屋、コンビニチェーン、学校**

情報は構造をなしている

情報処理するためには、実世界の情報の構造を明らかにして、これを情報システムに適切なデータとして与えるための表現が必要。情報の**意味は情報の構造が与える**。

情報 (information)

情報は**記号** (symbols) で表される。明らかにされるべき**情報の構造** (information structure) とは、**記号の性質や記号間相互の関係**全体である。情報の種類とその構造を知った上で、その情報を自然に表現する方法を考えねばならない。

データ (data)

記号化され情報のこと。情報処理システムの対象となり得るもの

その**部分におけるデータ型**が明らかにされるべき

注意：数字には意味がない。「201411」は象形の連なり？、重さ（ton/kg）？、高さ（m/feet）？、面積（km^2/m^2）？郵便番号？、年月日？**単位は極めて重要**

データ構造 (data structure)

データの基本形（原始形）が情報構造の明確化のための第一歩

プログラム言語で使われる基本データ型

整数型 (integer)、**実数型** (real)、**論理型** (boolean)、**文字型** (character)

基本データ型から複雑なデータ構造を定義する

例：プログラミング言語 Pascal **でデータ構造** person**（人）を定義**

```
const size = 64;  // 定数 size の宣言
type word = array[1..size] of character; //文字列 word
     person = record name: word;  //構造型 person の定義
                     email: word;
                     height: real;
                     gender: char;
                     age: 0..100;
                     student: boolean;
                end;
```

データ構造 person を持つ変数を宣言（Pascal で）

```
var he, she : person; // person 型変数 he と she の宣言

//変数に値を代入
he.name = "taro";  he.email = "taro@hoge.ac.jp";
he.height = 175.3; he.gender = "M"; he.age = 20;
he.student = true;

she.name = "umeko";  she.email = "ume@foo.co.jp";
she.height = 170.0;  she.gender = "F"; she.age = 24;
she.student = false;
```

情報構造 (information structure)

情報がどのような属性から構成されているか（情報構造）が明らかにすることによって、複雑な情報を表現できる。

属性 (property) と属性値 (value)

情報は構造としての型以外に、それを特徴付ける属性 (property) があり、ある属性値を持つ

hasa 関係（属性）

S **は属性** Q **を持つ（**S has a property Q**）** $\Rightarrow$ $S \xrightarrow{\text{hasa}}$ (**属性**,Q)

属性 $(S) = Q$

対象 $S \rightarrow$ (**属性**, **属性値**)　　$\Leftarrow$ **情報構造の基本的記述法**

$\Rightarrow$ **関係とデータベース**

例：太郎の hasa **関係**

太郎 $\rightarrow$(**身長**, **小さい**)
太郎 $\rightarrow$(**職業**, **学生**)
太郎 $\rightarrow$(**メール**,taro@hoge.ac.jp)
太郎 $\rightarrow$(**性別**, **男**)
太郎 $\rightarrow$(**住所**, **東京都**)
太郎 $\rightarrow$(**体重**, **軽い**)

対象群の hasa 関係を集めて表 (table) が得られる

	対象	性別	身長 (cm)	住所	メール
	field↓	↓	↓	↓	↓
record→	梅子	女	158	世田谷区	ume@foo.co.jp
→	太郎	男	172	塩釜市	taro@hoge.ac.jp
→	次郎	男	165	国分寺市	jiro@piyo.or.jp
→	花子	女	170	福岡市	hana@fuga.ac.jp

Table: 「私の友人」という表データ

- 表をファイル（ひとまとまりのデータ）として保存して
- 表のファイル内での 1 件の対象データをレコード
- そのレコードでの項目をフィールド

複数の表データ群を用意して、それらを操作・利用することは大きな有用性がある ⇒ 関係データベース

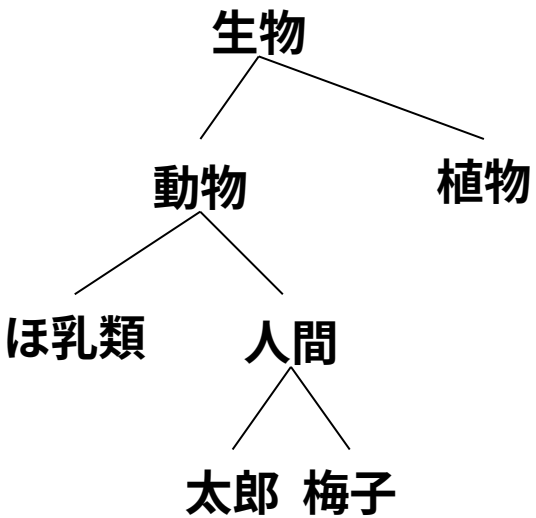
- 関係は表で表されるが、行の順番には意味がない。梅子と花子の行（5 項関係）を入れ替えても構わない。

概念情報の階層性 (hierarchy)

上位概念 (upper concept) **と下位概念** (lower concept) **を** isa **関係で記述（木** (tree) **で表現）**

isa 関係（である関係）

S **は** P **である（**S is a P**）** $\Rightarrow$ $S \xrightarrow{\text{isa}} P$　S **と** P **は** is **関係**



太郎 $\xrightarrow{\text{isa}}$ **人間、人間** $\xrightarrow{\text{isa}}$ **ほ乳類**

演習：第 49 **回学生生活実態調査の概要報告の図表** 7 **について、自分自身についての情報構造を決定せよ。**

データ集合 data set

集合

記号化された情報（データ）において最も簡単な構造は、そのデータが相互に繋がりがないデータの集まりのどれかである場合。そのデータの集まりを**集合** (set) という。このとき、データ a は集合 A に**属する**（または A の**要素である**）といい、$a \in A$ と表す（属さないときは $a \notin A$）。

集合を表す

集合 A の要素 $a_1, a_2, \ldots, a_n$ が列挙可能で有限個のときは、中括弧 $\{\}$ 内に要素をカンマ , で区切って次のように並べる

$$A = \{a_1, a_2, \ldots, a_n\}$$
$$= \{a_{\sigma(1)}, a_{\sigma(2)}, \ldots, a_{\sigma(n)}\}$$

$\sigma : 1, 2, \ldots, n$ の並べ替えとすると

要素が同じなら並ぶ順番は無関係

列挙不可能な場合には、集合要素が満たすべき条件を次のように明記する。

$$A = \{x \mid \text{要素 } x \text{ の満たすべき条件 }\}$$

集合演算 set operation

集合 A と B および全体集合 U が与えられているときの和 (union)**、積** (intersection)**、差** (difference)**、補集合** (complement)**：**

和 $A \cup B = \{x \mid x \in A$ **または** $x \in B\}$

積 $A \cap B = \{x \mid x \in A$ **かつ** $x \in B\}$

差 $A - B = \{x \mid x \in A$ **かつ** $x \notin B\}$

補 $A^c = \{x \mid x \in U - A\}$

演習 1：集合演算をベン図で表せ。

演習 2：全体集合 $U = \{x \mid x$ **は 0 以上 100 未満の整数** $\}$, $A = \{y \mid y$ **は** U **内の 7 の倍数** $\}$、$= \{z \mid z$ **は** U **内の 6 の倍数** $\}$ **としたとき、** $A \cup B, A \cap B, A - B, B - A, A^c$ **を求めよ。**

部分集合

集合 A, B **に対して任意の** A **の要素** $({}^\forall a \in A)$ **が** B **の要素であるとき、** A **は** B **の部分集合** (subset) **といい、** $A \subset B$ **と記す。**

注意：部分集合の定義から、任意の集合 A **に対して** $A \subset A$

例

- 天気 $=\{$ 晴れ, 曇り, 雨 $\}$ とする。このとき、hasa 関係「東京 $\rightarrow$ (天気, 雨)」は「東京の天気 $=$ 雨 $\in$ 天気」と表される。
- 晴れ $\in$ 天気 であるが、中括弧を用いた $\{$ 晴れ $\}$ は集合となり、$\{$ 晴れ $\} \subset$ 天気 と「天気」の部分集合である。
- 湿度 $=\{$ 乾燥, 快適, 不快 $\}$ とする。天気 $\cap$ 湿度 $=\{\}$ となり、共通要素を持たない。
- $\{\}$ は形式的には要素がない集合を表していると考えられる。$\{\}$ を**空集合** (empty set) といい、記号 ϕ で表す。

演習：任意の集合 A に対して、$A \cup \phi = A$, $A \cap \phi = \phi$, $A - \phi = A$, $\phi - A = \phi$, $\phi^c = U$ であることを確かめよ。空集合 ϕ は四則演算における 0 の役割をしている（$\cup \rightarrow$ 和、$\cap \rightarrow$ 積）。

直積集合 (direct product) と関係 (relation)

- 集合 A と B の要素 $a \in A$ と $b \in B$ から作った (a, b) を 2 項関係 (binary relation)。$(a, b) \neq (b, a)$ であり、$(a, b) = (c, d)$ のときは $a = c, b = d$.
- 集合 $A \times B = \{(x, y) \,|\, x \in A,\, y \in B\}$ を A と B の直積。
  $^\forall a \in A, {}^\forall b \in B$ で作る 2 項関係 $(a, b) \in A \times B$.

例：$A = \{a, b\},\ B = \{x, y, z\}$ のとき、

$$A \times B = \{(a, x), (a, y), (a, z), (b, x), (b, y), (b, z)\},$$
$$B \times A = \{(x, a), (x, b), (y, a), (y, b), (z, a), (z, b)\}.$$

- $A \times B$ の部分集合 R を関係といい、その要素である 2 項関係 $(a, b) \in R$ を aRb と記す（一般に $aRb \neq bRa$）。
- 一般に、$A_1 \times \cdots \times A_n = \{(a_1, \ldots, a_n) \,|\, a_i \in A_i, i = 1, \ldots n\}$ の要素 $(a_1, \ldots, a_n)$ を n 項関係 (n-ary relation) とし、これを集めた部分集合を関係 R と定義する。

関係という集合は一見抽象的にみえるのだが、生活用語の「関係」を表していることがわかる。

関係集合は「関係」を表す

「a は b を好き」を 2 項関係 (a, b) で表わそう。
A $=\{$ 梅子, 花子 $\}$, B $=\{$ 太郎, 一郎, 次郎 $\}$ とする。

- 集合 $R_L = \{$(梅子, 太郎),(花子, 一郎)$\} \subset A \times B$ は女→男の好意関係であり、梅子 R_L太郎, 花子 R_L一郎 と記す。
- 集合
  $R_\ell = \{$(太郎, 梅子),(太郎, 花子),(一郎, 梅子),(次郎, 花子)$\} \subset B \times A$ は男→女の好意関係であり、太郎 R_ℓ梅子,
  太郎 R_ℓ花子, 一郎 R_ℓ梅子, 次郎 R_ℓ花子 と記す。

演習 1：aPb を「街 a から街 b に新幹線で行ける」関係とする。
$A = \{$ 東京, 新潟, 岡山 $\}$, $B = \{$ 甲府, 京都, 宮崎, 秋田 $\}$ としたとき関係集合 P を決定せよ。

演習 2：aWb を「a は b より重い」関係とする。
$A = \{$ 鶏, 馬, 牛 $\}$, $B = \{$ ツバメ, 象, メダカ, 人 $\}$ としたとき関係集合 W を決定せよ。

情報処理を再定義する

再定義 (その 1)

構造を持った入力情報から、それらにどのような演算操作を施すか逐次的に定めて有意味な出力情報を得ること

例：人間 1=(性別、体重、身長)、人間 2=(走力、水泳、重量挙)、人間 3=(名前、住所、メール)、人間 4=（成績、出身大学、勤務先)、人間 5=(優しさ、性格、趣味)

演習 1：企業情報の構造を考え、どのように活用するかを考えよ

再定義 (その 2)

そもそも、入力情報からどのようにして構造をもった各種属性を抽出・列挙を得るかを定め、収集した各種情報を組織的に構造化データ化とすること

演習 2：ビッグデータの活用とは何か

しかし対象から情報や意味を得ることは難しい

[対象] テキスト：
『ノッポの太郎君と梅子さんが居る 40 人のクラスで席順を話合いました。その結果、背の高い人が後ろに座ることになりました。一番前の席に決まった次郎君の右隣には梅子さんが座ることになりました。男性の過半数が後方に座っています。』

演習 1：テキストから次の情報を取り出すプロセスを書き出せ

- 太郎君が座っている場所
- 太郎君は次郎君の背の高さを比較する
- 梅子さんの背の高さ
- クラスの女性と男性の人数比較
- 記載内容をいつ、誰が書いたか

演習 2：同様な『文章』自分で書いてみて、どのような情報が取り出せるのか、取り出せないかを詳しく報告せよ

情報とそのデータの関係

いままでの議論でわかったこと（注意すべきこと）：

対象としているモノの情報　≠　データ自身

目的に応じたデータ設計：情報を有する対象から

- 目的：どのような情報を必要としているのか？
- 蓄積：どんな属性や関係性に着目したデータを格納するか？
- 方法：データから有意味な情報をどうやって取り出すか？
- システム構築：情報収集から情報の取り出しまでの自動化

情報・データと意味

人はデータから情報を取り出し、その意味を理解することができる。この仕組みはきわめて複雑で、人の認知活動に関係しており依然不明なことが多い。

演習：地図と現実世界との対応関係を理解できるのは人だけであることを理解せよ。

測定（現象や行動）によって得られるデータは対象全体が有する対象間の結びつきや複雑な関係性を表している。⇐ 人はそこに意味を見いだす

言語学における格文法 case grammar による意味論

ある現象・行動に対して語要素間の係わり方を動詞と深層格（意味・役割）とを中心とした格構造を明らかにして、言語（テキスト）を把握する方法（1968 年 Charles Fillmore、フレーム意味論、人工知能論へ発展）

意味論をデータ構造化による意味ネットワークとして援用

自然言語処理の主な応用

- 機械翻訳
- 情報検索
- 情報抽出（要約を含む）

形態素解析	語の構造（文のアトム的分解）
統語解析	文文法
意味解析	語・文の意味
談話解析	文と文との関係

Table: 自然言語理解の情報処理段階

情報処理の方向性

処理目的：定型から**非定型**へ、量的計算から**質的判断**へ

1. 構造化（定型）データ（関係データベース管理システム RDBMS）を使った（ルーチンとしての）情報処理から、
2. 大規模に収集した現実世界を反映している（はずの）非構造化（非定型）データに基づいて、発見的に（あからじめ意図していなかった）非自明で有用な情報を取り出す（データマイニング）へ
3. 正しく唯一の解がない現実の問題（論理や経験から解決を見いだせない課題）を解決するための活用へ

計算方法：単独計算から**並列計算** (parallel computing) さらに**並行計算** (Concurrent Computing) へ、そして**分散計算** (distributed computing) や**グリッド計算** (grid) へ

資源利用：自前のシステムから**クラウドコンピューティング** (cloud computing) へ

データ表現の姿

構造化データ (structured data)

同一形式・同一構造（定型）を持つように記述・記号化されたデータ群。狭義には、hasa 関係を定義した関係データベース（関係集合）やオブジェクトデータベースのデータ。

非構造化データ (unstructured data)

一貫した構造定義を有しない非定型データ群。狭義には、人同士のコミュニケーションで用いられるような自由記述文書・書籍、画像・映像、音声など。非構造化データではデータ内容をその用途に応じたメタデータ**によってタグ付けする。**

メタデータ (metadata)

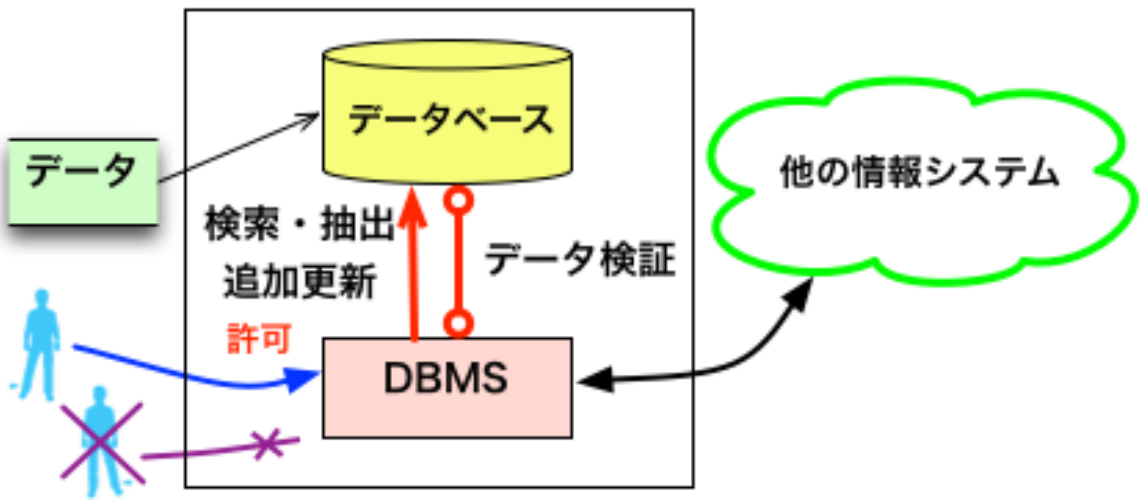
データが付随して持つデータ自身についての高位の（メタ：抽象度の高い）データ（データについてのデータ）。スキーマ (schema)**を統一して、メタデータの相互運用性を実現できる。書籍の書誌情報、写真データ、音楽情報など日常的にも使われている。**

データベース (database)

特定の目的のために収集したデータを検索・抽出や追加更新などの操作ができるようにしたもの。情報処理のプラットホーム

データベース管理システム (Database Management System)

データベースを管理（アクセスや更新・蓄積できる人も管理）した上で、データが正しい形式になっていることを保証しながらデータベースをコンピュータで利用するための情報システム。



他の情報システムは DBMS のサービスを利用しながら必要な処理を行う（情報資源の共有化）。

演習：次の情報システムのデータベースを説明しなさい

- **警察、病院**
- **銀行**
- **楽天、**Apple, Amazon
- Google

データマイニング (data mining)

適切な構造化データでは、目的とするデータ要素へのアクセスが容易である。一方、非構造化データではそれ自身が有している情報そのものをメタデータの組み合わせとして取り出すことは容易ではない。

例 1：テキスト『ここではきものをぬいでください。にわにはにわとりがいます。』

例 2：全文 Web 検索サービス、PageRank 技術

データマイニング

統計学、言語理論、パターン認識、人工知能などの知見・技法を活用して、収集した（一見無意味に見える）データから**非自明で有意味な情報を取り出す**（掘り起こす）こと。大量のデータを**収集し、保存**しておくこと、**大規模データ群のさまざまな組み合わせ**に対して網羅的に**複雑な情報処理**を可能とする**コンピュータパワー**の達成よって実現。

演習：ビッグデータのデータマイニング事例を挙げよ

関係データベース管理システム RDBMS の役割

関係データベースの各データは、表計算で扱う表 (table) **と同じように見える。しかし、** (**行**, **列**) **の位置指定はなく根本的に異なる（ただし、説明では「表」を使う！）。**

RDBMS(Relational Database Management System) **では、「表」の集まりに対して、どの値がどこにあり、複数の表がどう関連しているかを管理するために、背後で膨大な計算処理を行っている。**

SQL(Structured Query Language)

RDBMS **はデータベースに対して様々な操作を行い、データの更新や新しくデータを生成するための操作言語（問い合わせ言語）**

代表的 RDBMS

- **市販：**Microsoft Access(**小規模**)/SQL Server, Oracle Database, IBM DB2/Infomix **など**
- **オープンソース：**MySQL, PostgresSQL **など**
- **クラウドサービス**: Amazon RDB, Google Cloud SQL

関係データベースにおけるデータ

関係データベースにおけるデータ形式は、列 (column) **と行** (row) **からなる表** (table) **で表される。ただし、例では説明のために** record_i **と順番をふったが、行の並び順には意味がない（**n**-項関係を要素とする集合であるため)。つまり、(行目, 列目) による場所指定（配列）の概念はない。**

	field_1	field_2	field_3	field_4	field_5
record_1	**梅子**	**女**	158	**世田谷区**	ume@foo.co.jp
record_2	**太郎**	**男**	172	**塩釜市**	taro@hoge.ac.jp
record_3	**次郎**	**男**	165	**国分寺市**	jiro@piyo.or.jp
record_4	**花子**	**女**	170	**福岡市**	hana@fuga.ac.jp
⋮	⋮	⋮	⋮	⋮	⋮

列数は固定：各行 (1 レコード) は n**-項関係を表している。**
$\mathrm{field}_1 =$ **名前**, $\mathrm{field}_2 =$ **性別**, $\mathrm{field}_3 =$ **身長** cm, $\mathrm{field}_4 =$ **住所**, $\mathrm{field}_5 =$ **メール**
行数は可変：必要なだけレコードを追加できる（しかし、レコード順には意味がない)。

データ操作言語 SQL