第 1 講 あらましと予備知識

林　恒俊

講義の内容

- 当講義は記号と記号列に関する数学すなわち**形式言語理論** (**formal language theory**) への導入を目的としている。
- 一般にデジタル情報は記号列と見なすことができる。なぜならデジタル情報は離散値の列であり、離散値は個別の値を記号として取扱うことができるからである。
- 記号列がもつ様々な性質について考察することは、デジタル情報を取扱う上で非常に重要であると考えられる。
- さらに記号列の持つ性質が機械的に調べられるかという点が重要である。
- 記号列の性質に関する検証を行うため、そのアルゴリズムを**自動機械** (**automaton**, *pl* **automata**) という形式で実装する。
- 自動機械の動作は数式 (形式的記述) を利用して定義し、数式の変形を通して演繹する。
- 人間の恣意の介在をさけるこのような数学的手順を**形式的手法** (**formal method**) という。
- 数学とは形式的手法によりその対象の性質を厳密に究める学問である。そして形式言語理論は記号と記号列を対象とする数学である。
- 本講義の目標はこれらの話題について初等的な知識を獲得することである。それと同時に計算機科学における数学に基づいた厳密な方法論について準備する。

考察

CD に記録されている情報は 16 ビット $\times$ 2 単位のサンプルが計約 600 MByte 程度まで順に並べられたものと考えることができる。これは記号 $\{0,1\}$ から構成される長さ約 $8 \times 600 \times 10^6$ の非常に長い列と見なされる。しかし列がいかに長くても有限であることに変わりはない。有限な列と有限でない (長さに限界がない場合) 列の間には質的な違いが存在する。

参考書

- 次の本は教科書として定評があるが、本講義には高度すぎるので必要に応じて辞書として利用するとよい。この方面の学問を究めようとするなら必須である。
  - J E Hopcroft & J D Ullman, *Formal Languages and Their Relation to Automata*, Addison-Wesley

    野崎及木村訳、**言語理論とオートマトン**、サイエンス社
- 次の本は比較的入門に向いているが、計算論まで含まれているため当講義の範囲を超えている。
  - E Kinber & C Smith, *Theory of Computing: A Gentle Introduction*, Prentice Hall

    杉原及筧訳、**計算論への入門**、ピアソンエデュケーション
- 次の本は形式言語やオートマトンのみならず広範囲の内容を含むエッセーである。読本としては適切だが教科書には向いていない。時間に余裕があるときに目を通すとよい。
  - D R Hofstadter, *Gödel, Escher, Bach: An Eternal Golden Braid*, Basic Books

    野崎 昭弘訳、**ゲーデル, エッシャー, バッハ—あるいは不思議の環**、白揚社

- なお本講義で挙げる実例や表記法、証明の道筋などは次の教科書を参考にした。
  - R Greenlaw & H J Hoover, *Fundamentals of the Theory of Computation*, Morgan Kaufmann

講義の進め方と成績評価

- 講義は基本的に公表されたシラバスにそって進行する。
- 講義の成績は基本的に期末試験の成績に基づいて評価される。
- 講義期間中に小テストを 1 回予定している。日時はあらかじめ公表しない。
- 不幸にして期末試験成績が充分でない場合に限って小テストの結果が出席点として加算される。期末試験成績が充足していれば小テストが最終成績には影響することはない。
- なおすべての試験はマークシート利用を予定しているので、受験希望者は HB 以上の鉛筆を用意すること。正しく読み取れないマークは評価されない。

予備知識

- 本論に進む前に前提となる原則や記法についてあらかじめ説明する。
- もし中学・高校で学習済みならここで復習しておくとよい。

集合

- 記号に関する理論は主に有限な集合を基礎として組み立てられる。集合に関する基本的な考え方を説明する。

- **集合** (**set**) は文字通り対象となるものの集まりをいう。数値、記号などあらゆるものが対象になる。集合に関する最も重要な性質は**要素** (**member**, **element**) であるかどうか (**membership**) である。その他の性質はほとんど意味を持たない。
- 要素が重複しても無視される。
- 集合は中括弧 { } の中に要素を明示的に列挙して定義する。
- 例えば
  - $\{0, 1\}$
  - $\{a, b, c\}$
  - $\{a_1, a_2, a_3, \ldots, a_n\}$
  - $\{0, 2, 4, 6, \ldots\}$

  などである。
- これらの集合の最初のものは 1 桁 2 進数を要素とする集合、次は 3 要素の集合、3 番目は n 要素の有限集合、最後は非負の偶数の集合を表している。
- 最後のものは有限でない集合を表示している。しかしすべての要素を列挙することができないため表示が曖昧である。
- このような場合集合の要素を判定する (論理) 式を利用した定義が行われる。

  $$S = \{x \mid p(x)\}$$

  により S は条件 $p(x)$ を満たす要素 x の集合と定義される。非負の偶数の集合なら

  $$\{n \mid n \geq 0 \wedge n \bmod 2 = 0\}$$

  と表示することができる。
- あるいは

  $$S = \{e(x) \mid p(x)\}$$

  により条件 $p(x)$ を満たす値 x の式 $e(x)$ を要素とする集合を定義する。非負の偶数の集合は

  $$\{2n \mid n \geq 0 \wedge n \text{ は整数 }\}$$

- 有限な集合 S では $|S|$ で集合の要素の数を示す。例えば $|\{0,1\}|=2$ である。
- 要素の数が 0 である集合を**空集合** (**empty set**) といい $\{\}=\emptyset$ で表記する。当然 $|\emptyset|=0$ である。
- 集合の要素であることを $0\in\{0,1\}$ と表記する。要素でないことを $d\notin\{a,b,c\}$ と表記する。$\in$ は要素演算を示している。

考察

上記の論理式中に $\wedge$ という記号が使われているがこれは “**および (and)**” 演算を表示する演算子である。同様に “**または (or)**” 演算は $\vee$ で、“**否定 (not)**” 演算は $\neg$ で表示する。これらの演算子の演算結果は T を “**真**”、F を “**偽**” として次の表で与えられる。

$\wedge$	T	F
T	T	F
F	F	F

$\vee$	T	F
T	T	T
F	T	F

$\neg$	
T	F
F	T

これらの演算は互いに関係があり a,b を論理値変数とすると $a\wedge b=\neg(\neg a\vee\neg b)$ である。この式を論理式の DeMorgan の法則という。確かめよ。

このような論理値に基づいた数学は**数理論理学** (**mathematical logic**) と呼ばれ数学のみならず理学や工学の基盤として重要な役割を果たしている。

数学では集合の他に様々な対象が考察されている。例えば**体** (**field**)、**群** (**group**)、**環** (**ring**)、**束** (**lattice**) などである。これらの対象は要素間の演算の性質に基づいて定義される。集合は最も単純な対象と考えられる。

集合の演算・操作

集合間に次のようにいくつか演算を定義することができる。

部分集合 (subset) ある集合 A の要素が必ず集合 B の要素である時 A は B の部分集合といい $A \subseteq B$ と表示する。

同一性 (equality) $A \subseteq B \wedge A \supseteq B$ の時 2 個の集合 A と B は同一の要素から構成される。これを $A = B$ と表示する。

和集合 (union) 2 個の集合 A, B の要素を合わせた集合を和集合といい $A \cup B$ と表示する。$A \cup B = \{x \mid x \in A \vee x \in B\}$ と定義される。

結び (intersection) 2 個の集合 A, B の共通の要素からなる集合を結び集合といい $A \cap B$ と表示する。$A \cap B = \{x \mid x \in A \wedge x \in B\}$ と定義される。

差集合 (difference) ある集合 A からもう 1 個の集合 B の要素を取除いた集合を差集合といい $A - B$ と表示する。

$$A - B = \{x \mid x \in A \wedge x \notin B\}$$

と定義される。

補集合 (complement) すべての (関連する) 要素の集合からある集合 A の要素を取除いた集合を補集合といい A^c と表示する。

$$A^c = \{x \mid x \notin A\}$$

と定義される。

考察

これらの定義から理解されるように、集合の和、結び、補集合演算は論理演算の “または”、“および”、“否定” 演算と非常に関連が深い。

集合演算に関するいくつかの定理

- つぎにこれらの演算の間に成立するいくつかの定理を挙げる。各自確かめておくこと。なお A, B, C は任意の集合とする。また U ですべての関連する要素を含む集合を代表する。

- $A \cap A = A$, $A \cup A = A$
- 交換則
$$A \cup B = B \cup A$$
$$A \cap B = B \cap A$$
- 結合則
$$A \cup (B \cup C) = (A \cup B) \cup C = A \cup B \cup C$$
$$A \cap (B \cap C) = (A \cap B) \cap C = A \cap B \cap C$$
- $A \cup \emptyset = \emptyset \cup A = A$, $A \cap \emptyset = \emptyset \cap A = \emptyset$
- $A - \emptyset = A$, $\emptyset - A = \emptyset$
- $A^c = U - A$, $(A^c)^c = A$, $\emptyset^c = U$, $U^c = \emptyset$
- $A - B = A \cap B^c$
- 集合の DeMorgan の法則
$$A \cap B = (A^c \cup B^c)^c$$

記号集合上の関数

- 解析学のような連続した数を対象とする場合、関数は一般に代数式で表現される。例えば $f(x) = x^2$ で関数 f が引数の 2 乗の値をとることを示している。しかし記号や記号列のような離散値を対象とする関数は数式で表現できないことが多い。
- 関数は一般に引数値を与えると対応する関数値を返すような仕組みである。いいかえると引数値と関数値を対応づける対応表のようなものと考えらる。このような対応を**写像** (**mapping**) という。
- 記号や集合を対象とする関数は引数の値の集合すなわち**領域** (**domain**) の値の要素と関数値の集合すなわち**値域** (**range**) の要素の対応を示している組の集合として定義することができる。
- 様々な値の組合せを要素とする集合を考える。一般にこのような組合せを**組** (**tuple**) と呼ぶ。特に

- 2 個の要素からなる組を**対** (**pair**)
- 3 個の要素からなる組を **3 つ組** (**triple**)
- 4 個の要素からなる組を **4 つ組** (**quadruple**)
- n 個の要素からなる組を n-tuple

という。これらの組の値は座標表記と同様にその要素を括弧で括って表記する。

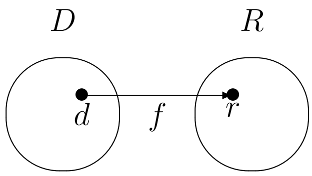
- 領域集合を D、値域集合を R とすると D の値 d を R の値 r に写す関数 f は

$$r = f(d)$$

または

$$f : d \to r$$

と表記される。



そして f は対 (d, r) の集合

$$f = \{(d, r) \mid d \in D, r \in R\}$$

として定義できる。

直交積、関係、関数

- 2 個の集合の要素からなるすべての組を集めた集合を**直交積** (**Cartesian product**) と呼ぶ。集合を S_1, S_2 とすると S_1 と S_2 の直交積は

$$S_1 \times S_2 = \{(s_1, s_2) \mid s_1 \in S_1, s_2 \in S_2\}$$

と表示する。

- 2 個以上の集合についてもこの定義を拡張して直交積を定義する。

$$S_1 \times S_2 \times \cdots \times S_n = \{(s_1, s_2, \ldots, s_n) \mid s_1 \in S_1, s_2 \in S_2, \ldots, s_n \in S_n\}$$

- 直交積の部分集合を**関係** (**relation**) と呼ぶ。例えば**同値関係** (**equivalence relation**) は 1 個の集合 S の直交積 $S \times S$ から構成される関係である。
- 一見関数 $f : d \to r$ の定義は領域と値域の直交積 $D \times R$ と同一の表現である。しかし関数の定義は次のように限定されので、f が $D \times R$ の要素をすべて含むわけではない。f は $D \times R$ の部分集合と考えられる ($f \subseteq D \times R$)。
- 関数ではある領域値について関数値は高々1 個しか存在しない。したがって領域値が重複しない要素対の集合を**関数** (**function**) と呼ぶ。
- 関数が領域のすべての要素について定義されていれば**完全関数** (**total function**) という。そうでない場合は**部分関数** (**partial function**) という。
- 領域や値域がさらに直交積の場合には関数は多数の集合の直交積の部分集合として定義できる。例えば領域 $D_1, D_2, \ldots, D_n$ 上の値域 R の関数 $r = f(d_1, d_2, \ldots, d_n)$ は直交積 $D_1 \times D_2 \times \cdots \times D_n \times R$ の部分集合として定義される。