第 3 講 言語の定義と操作

林　恒俊

言語

- あるアルファベット Σ が与えられているものとする。Σ 上の記号列の有限あるいは無限集合を Σ 上の**言語** (**language**) と呼ぶ。言語を代表するためによく L のような記号を利用することがある。
- 例えばアルファベット $\{0,1\}$ 上の言語
  - すべての記号列を含む言語 $\{\Lambda, 0, 1, 00, 01, 10, 11, \ldots\}$
  - 0 のみからなる記号列の言語 $\{\Lambda, 0, 00, 000, \ldots\}$
  - 要素がない言語を**空言語** (**empty language**) という
  - 空言語はは空集合と同じで $\{\ \} = \emptyset$
  - 空列のみからなる言語 $\{\Lambda\}$ は空言語ではない
  - 長さ 2 以下の記号列からなる言語 $\{\Lambda, 0, 1, 00, 01, 10, 11\}$
- 言語は定義上集合であり集合に関する性質がそのまま言語についても成立する。

アルファベット上のすべての記号列

- アルファベット Σ 上のすべての記号列の集合は当然 Σ 上の言語でありこの言語を Σ^* と表示する。なおこの記法の由来について後で解説する。
  - $\Sigma = \{0\}$ なら $\Sigma^* = \{0\}^* = \{\Lambda, 0, 00, 000, \ldots\}$
  - $\Sigma = \{a, b\}$ なら $\Sigma^* = \{a, b\}^* = \{\Lambda, a, b, aa, ab, ba, bb, \ldots\}$

- アルファベット Σ で $\Sigma^{\leq n}$ と表示される言語は整数 n について長さが n 以下のすべての記号列からなる言語である。また $\Sigma^{=n}$ と表示される言語は長さが丁度 n のみのすべての記号列からなる言語である。
- 例えば

$$\{1\}^{\leq 5} = \{\Lambda, 1, 11, 111, 1111, 11111\}$$
$$\{0,1\}^{=2} = \{00, 01, 10, 11\}$$

言語の定義

- 言語は集合そのものであるため集合の定義と同様な手段で定義される。例えば有限な言語なら要素を列挙して定義できる。有限でない言語の場合には次のような定義手法が利用できる。
- アルファベット Σ 上の言語 L を定義するために

$$L = \{x \mid x \in \Sigma^* \wedge p(x)\}$$

これは L の要素が Σ 上の記号列であり、かつ条件 p を満たしていることを示している。Σ が既知の場合は $x \in \Sigma^*$ の部分が省略されることがある。

- 例えば

$$L = \{x \mid x \in \{0,1\}^* \wedge |x| = 2\} = \{0,1\}^{=2}$$

は長さ 2 のみの記号列から構成される言語 $\{00, 01, 10, 11\}$ を示している。(有限な言語もこの手法で定義できる)

- 同様に記号列の式と条件を示して言語を定義することもできる。例えば

$$L = \{a^n b^n \mid n \geq 0\}$$

で同じ長さの a の列と b の列が連結された記号列からなる言語

$$\{\Lambda, ab, aabb, aaabbb, \ldots\}$$

を示している。

様々な言語

- アルファベット $\Sigma = \{a, b, c\}$ の言語のいくつかを示す。
- $\{a^n \mid n \geq 0\} = \{\Lambda, a, aa, aaa, \ldots\}$
- $\{a^n \mid n \text{ は素数}\} = \{aa, aaa, aaaaa, aaaaaaa, \ldots\}$
- $\{a^m b^n \mid n, m \geq 0\} = \{\Lambda, a, b, aa, ab, bb, \ldots\}$
- $\{a^n b^n \mid n \geq 0\} = \{\Lambda, ab, aabb, aaabbb, \ldots\}$
- $\{a^n b^n c^n \mid n \geq 0\} = \{\Lambda, abc, aabbcc, \ldots\}$
- $\{a^m b^n c^{m+n} \mid m, n \geq 0\} = \{\Lambda, ac, bc, abcc, \ldots\}$

考察

上記の最後の例は非負整数の算術加算を定義する言語である。この言語の要素は最初の a の数と 2 番目の b の数の和が 3 番目の c の数と等しい記号列である。

もしこの言語を定義する形式的手段や、記号列が要素かどうか判定する形式的手段があれば、それは算術加算そのものの定義と見なすことができる。

このように言語はその要素の記号列を通して様々な概念を定義するものと考えることが可能である。

言語の演算操作

- 言語は集合として定義されているので集合に関する演算操作はすべて適用できる。
- 集合の和、結び、差、補集合など。
- 記号列を要素としているので記号列の操作に基づく演算操作も可能である。言語の

- **連結** (**set concatenation**)
- **長さ部分集合言語** (**length subsetting**)
- **逆列言語** (**reversal**)

などである。

集合演算

- アルファベット Σ 上の言語を L_1, L_2 とする。L_1, L_2 について次のような操作 (演算) を行うことができる。
- 言語の和　　$L_1 \cup L_2 = \{x \mid x \in L_1 \vee x \in L_2\}$
- 言語の結び　$L_1 \cap L_2 = \{x \mid x \in L_1 \wedge x \in L_2\}$
- 言語の差　　$L_1 - L_2 = \{x \mid x \in L_1 \wedge x \notin L_2\}$
- 補集合　　$L^c = \Sigma^* - L = \{x \mid x \in \Sigma^* \wedge x \notin L\}$
- これらの演算は集合の場合と同様に交換則、結合則などを満たしている。例えば DeMorgan の法則

$$L_1 \cap L_2 = (L_1^c \cup L_2^c)^c$$

が成立する。

- $\Sigma = \{0, 1\}, L_1 = \{\Lambda, 0, 00, 000, \ldots\}, L_2 = \{\Lambda, 1, 11, 111, \ldots\}$ とすると

$$\begin{aligned} L_1 \cap L_2 &= \{\Lambda\} \\ L_1 \cup L_2 &= \{\Lambda, 0, 1, 00, 11, 000, 111, \ldots\} \\ L_1 - L_2 &= \{0, 00, 000, \ldots\} \\ L_1^c &= \{1, 01, 10, 11, 001, \ldots\} \end{aligned}$$

言語の連結

- 言語の連結演算は集合とは独立した言語固有の演算である。アルファベット Σ 上の言語を L_1, L_2 とする。L_1, L_2 を連結した言語

$$L = L_1 \circ L_2 = L_1 L_2$$

を次のように定義する。

$$L = \{x \circ y \mid x \in L_1 \wedge y \in L_2\}$$

すなわち各々の言語の要素を取出して連結したものすべてを要素とする言語である。

- 有限な言語の連結では最大で要素の数が元の言語の要素の数の積になる。すなわち
$$|L_1 \circ L_2| \leq |L_1| \times |L_2|$$
- 例えば $L_1 = \{ab, abb, bb\}, L_2 = \{bb, bbb\}$ とすると
$$L_1 \circ L_2 = \{abbb, abbbb, abbbbb, bbbb, bbbbb\}$$
なぜなら $ab \circ bbb = abb \circ bb$
- L を任意の言語とすると
$$L \circ \emptyset = \emptyset \circ L = \emptyset$$
- L を任意の言語とすると
$$L \circ \{\Lambda\} = \{\Lambda\} \circ L = L$$
- ただし言語連結演算に交換則は成立しない。$L_1 \circ L_2$ は $L_2 \circ L_1$ と常に同じではない。
- 言語連結演算についても結合則が成立する。言語 L_1, L_2, L_3 について
$$(L_1 \circ L_2) \circ L_3 = L_1 \circ (L_2 \circ L_3) = L_1 \circ L_2 \circ L_3$$
(確かめよ)
- 任意の言語 L を繰り返して n 回連結したものを L^n で示す。
$$L^n = \underbrace{L \circ L \circ L \circ \cdots \circ L}_{n}$$
これは次のように再帰的に定義することも可能である。
$$\begin{cases} L^0 & = \{\Lambda\} \\ L^{i+1} & = L^i \circ L \quad i \geq 0 \end{cases}$$
($L^0 = \emptyset$ ではないことに注意)

- 言語 $L = \{0, 11\}$ とすると

$$L^0 = \{\Lambda\}$$
$$L^1 = \{0, 11\}$$
$$L^2 = \{00, 011, 110, 1111\}$$
$$L^3 = \{000, 0011, 0110, 1100, 01111, 11011, 11110, 111111\}$$

- 次の式が成立することを検証せよ。(分配則)

$$L_1 \circ (L_2 \cup L_3) = (L_1 \circ L_2) \cup (L_1 \circ L_3)$$
$$(L_1 \cup L_2) \circ L_3 = (L_1 \circ L_3) \cup (L_2 \circ L_3)$$

長さ部分集合による言語

- ある言語から特定の長さの要素を取出して別の言語を構成することができる。長さによる部分集合という。アルファベット Σ 上の言語 L について長さ n の要素を取り出して構成された言語を $L^{=n}$ と表示する。また長さ n 以下の要素を取り出して構成された言語を $L^{\leq n}$ と表示する。

- $L = \{0^i \mid i \geq 1\} = \{0, 00, 000, \ldots\}$ について

$$L^{=3} = \{000\}$$
$$L^{\leq 4} = \{0, 00, 000, 0000\}$$

- $L = \{0, 1\}$ について

$$L^{=0} = \emptyset$$
$$L^{=1} = L$$
$$L^{\leq 1} = L$$
$$L^{=i} = \emptyset \quad i \geq 2$$
$$L^{\leq i} = L \quad i \geq 2$$

逆列言語

- 言語の要素の逆列から別の言語を構成することができる。
- 言語 L の要素の逆列から構成される言語を L^R と表示する。L^R の定義は $L^R = \{x^R \mid x \in L\}$ である。
- 例えば $L = \{0^i 1^i \mid i \geq 0\}$ なら $L^R = \{1^i 0^i \mid i \geq 0\}$
- $(L^R)^R = L$

Kleene の * 演算

- ある言語の要素を重複を認めて (0 個を含め) 複数個取出し、それを連結したものを要素とする言語を考える。このような操作を Kleene の $*$ 演算あるいは **star-閉包** (**star-closure**) という。言語 L についてその star-閉包を L^* と表示する。
- 言語 L の L^* は

$$L^* = L^0 \cup L^1 \cup L^2 \cup \cdots = \bigcup_{i \geq 0} L^i$$

  と定義される。
- L^* には必ず空列が含まれることに注意。
- $\{\}^* = \emptyset^* = \{\Lambda\}$
- $\{\Lambda\}^* = \{\Lambda\}$
- $L = \{0, 1\}$ なら L^* はアルファベット $\{0, 1\}$ 上のすべての記号列の集合である。いままでアルファベット Σ 上のすべての記号列の集合を Σ^* で表現しているのはこの理由による。
- L^+ で L を少なくとも 1 個含む閉包を代表する。

$$L^+ = L \cup L^2 \cup \cdots = \bigcup_{i \geq 1} L^i = L \circ L^*$$

- $*$ 演算は要素の繰返しを実現する手段であり、言語理論では非常に重要な役割を担っている。正規言語や正規表現の中心課題である。

- この演算を利用すれば有限なアルファベットから要素が無限の言語を構成することができる。集合和、連結、部分列、逆列演算では演算の対象の言語が有限なら結果も有限である。