計算の理論

命令の実行 (＝「計算」) とは、

レジスタまたは主記憶の
現在の値 (状態) に従って、

その値を変更 (書込) すること

であった。

プログラム内蔵方式 (von Neumann 型) では、プログラム・データを区別なくメモリ上に置いていたが、やはり本質的に違うもの。

- プログラム: 一つの問題では固定
- データ: 可変な入力

$\Downarrow$

どんな (有効な) データ (入力) が来ても、
所定の出力を返すことが要請される。

或る問題の「計算が可能」

$\Updownarrow$

その計算を行なうプログラムが存在

$\Downarrow$

計算機の機能 ($=$ 「計算」のモデル)
を決めて議論する

ここでは、代表的な「計算のモデル」を
幾つか紹介する。

入力 (データ)

$\Downarrow$

プログラム

$\Downarrow$

出力・動作

原理・理論を考える際には、
出力は最も単純に「0 か 1 か」とする。

- 0 ： **拒否 (reject)**
- 1 ： **受理 (accept)**

入力 (データ)

⇓

プログラム

⇓

受理か拒否か

解くべき「問題」：入力を受理する条件

「問題」の例

入力の範囲: 文字 a, b から成る文字列

- **a と b との個数が同じ**
- **a が幾つか続いた後に**
  b が幾つか続いたもの
- **a で始まり a, b が交互に並んで b で終わる**
- **同じ文字列 2 回の繰返しから成る**
- **回文 (palindrome)**

などなど

それぞれの「問題」に対し、
定められた計算モデルで、
受理／拒否判定が可能 (問題が解ける) か？

受理される文字列が
「文法的に正しい」文字列だと思えば、

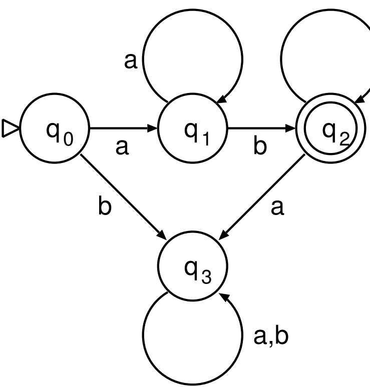
「問題」とは「文法 (言語)」である。

「文法的に正しい」かどうかの判定
・・・「構文解析」

代表的な計算モデル

- 有限オートマトン (有限状態機械)
- プッシュダウンオートマトン
- チューリングマシン

有限オートマトンの例 (状態遷移図による表示)



有限オートマトンの形式的定義

$$M = (Q, \Sigma, \delta, s, F)$$

ここに、

- Q : 有限集合 $\cdots$ **状態**の集合
- Σ : 有限集合 $\cdots$ 入力文字の集合
  "**alphabet**"
- $\delta : Q \times \Sigma \to Q$: **遷移関数**
- $s \in Q$ $\cdots$ **初期状態**
- $F \subset Q$ $\cdots$ **受理状態**の集合

先の例では、

- $Q = \{q_0, q_1, q_2, q_3\}$
- $\Sigma = \{a, b\}$
- $\delta : Q \times \Sigma \to \Sigma$ **:**

	q_0	q_1	q_2	q_3
a	q_1	q_1	q_3	q_3
b	q_3	q_2	q_2	q_3

- $s = q_0 \in Q$
- $F = \{q_2\} \subset Q$

Σ : **入力文字の有限集合** $\cdots$ **alphabet**

入力は Σ **の元の有限列 (語)**

$$w = a_1 a_2 \cdots a_n \;\; (a_i \in \Sigma)$$

その全体 Σ^*

$$\Sigma^* := \bigcup_{n=0}^{\infty} \Sigma^n \qquad (\Sigma^0 = \{\varepsilon\} : \text{空列})$$

言語 : Σ^* **の部分集合**

言語 $A \subset \Sigma^*$ **に属する語** $w \in A$

$\cdots$ **言語** A **に於いて "文法的に正しい"**

有限オートマトン $M = (Q, \Sigma, \delta, s, F)$ **が**
語 $w = a_1 a_2 \cdots a_n$ **を受理する**

$$\Updownarrow$$

$\exists r_0, r_1, \ldots, r_n \in Q$:

- $r_0 = s$
- $\delta(r_{i-1}, a_i) = r_i$ **(**$i = 1, \ldots, n$**)**
- $r_n \in F$

$L(M)$ **:** M **が受理する語の全体**
$\cdots$ M **が認識する言語**

M **は言語** $L(M)$ **の "文法" で、**
M **が受理する語は "文法的に正しい"**

有限オートマトンでの計算可能性問題

- 言語 $A \subset \Sigma^*$ に対し、
  A を認識する有限オートマトン M
  が存在するか？

- 有限オートマトンによって
  認識可能な言語はどのようなものか？

$\longrightarrow$ **正規言語・正規表現**

演習問題:

$\Sigma = \{a, b\}$ とする。

次の言語を認識する有限オートマトンを構成し、
状態遷移図で表せ。

(1) $A = \{a^{2n}b^{2m+1} | n, m \geq 0\}$
(a が偶数個 (0 個も可) 続いた後に、
b が奇数個続く)

(2) $B = \{vabbaaw | v, w \in \Sigma^*\}$
(部分列として $abbaa$ を含む)