

## 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形式的定義

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

ここに、

- $Q$  : 有限集合 … 状態の集合
- $\Sigma$  : 有限集合 … 入力文字の集合 : “**alphabet**”
- $\delta : Q \times \Sigma \rightarrow Q$  : 遷移関数
- $s \in Q$  … 初期状態
- $F \subset Q$  … 受理状態の集合

## 語・言語

$\Sigma$  : 入力文字の有限集合 … **alphabet**

入力は  $\Sigma$  の元の有限列 ( **語, word** )

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

その全体  $\Sigma^*$

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

**言語 (language)** :  $\Sigma^*$  の部分集合

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

… 言語  $A$  に於いて “文法に適っている”

## 有限オートマトンによる語の受理

有限オートマトン  $M = (Q, \Sigma, \delta, s, F)$  が

語  $w = a_1 a_2 \cdots a_n \in \Sigma^*$  を**受理 (accept)** する



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

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

$L(M) : M$  が受理する語の全体  $\subset \Sigma^*$

…  $M$  が**認識 (recognize)** する言語

$M$  は言語  $L(M)$  の“文法”で、

$M$  が受理する語は“文法に適っている”

## 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？  
  
→ 正規言語・正規表現

## 語の演算

語  $v = a_1 \cdots a_k, w = b_1 \cdots b_l \in \Sigma^*$  に対し

$$vw := a_1 \cdots a_k b_1 \cdots b_l$$

: 連結・連接 (**concatnation**)

連接演算により  $\Sigma^*$  は単位的自由半群を成す

---

$S = (S, \cdot) : \text{半群 (semigroup)}$



$\cdot : S \times S \longrightarrow S : \text{二項演算で結合律を満たす}$

## 正規演算

言語  $A, B \subset \Sigma^*$  に対し、

- $A \cup B := \{w \mid w \in A \text{ または } w \in B\}$   
: 和集合演算
- $AB = A \circ B := \{vw \mid v \in A, w \in B\}$   
: 連結 ( 接続 ) 演算
- $A^* := \{w_1 w_2 \cdots w_n \mid n \geq 0, w_i \in A\}$   
: star 演算

( 言語全体の集合  $\mathcal{P}(\Sigma^*)$  上の演算 )

## 正規表現 (regular expression)

- 空集合記号  $\emptyset$  は正規表現
- 空列記号  $\varepsilon$  は正規表現
- 各文字  $a \in \Sigma$  は正規表現
- 正規表現  $R, S$  に対し  
( $R \cup S$ ) は正規表現 ( $(R|S)$  とも書く)
- 正規表現  $R, S$  に対し  
( $R \circ S$ ) は正規表現 ( $(RS)$  とも書く)
- 正規表現  $R$  に対し  $R^*$  は正規表現
- 以上のものだけが正規表現

… 帰納的導出による定義



## 正規言語 (regular language)

正規表現  $R$  に対し、言語  $L(R)$  を次で定める：

- $L(\emptyset) = \emptyset$
- $L(\varepsilon) = \{\varepsilon\}$
- $L(a) = \{a\}$  ( $a \in \Sigma$ )
- $L(R \cup S) = L(R) \cup L(S)$
- $L(R \circ S) = L(R) \circ L(S)$
- $L(R^*) = L(R)^*$

正規表現で表される言語 …… 正規言語

定理 :

L : 正規言語



L が或る有限オートマトンで認識される

このような一般論を考えるには、  
有限オートマトンの概念を  
少し一般化する方が良い

… **非決定性有限オートマトン**  
(**Non-deterministic finite automaton**)

その説明の前に、今回は、

ここで現れる色々な概念を

集合・写像などの言葉を用いて記述する練習

( 演習問題 )

演習問題： $\Sigma$  を alphabet とする。以下を記述せよ。

(1)  $L$  : 言語

- (a) 文字  $a \in \Sigma$  に対し、語に左 (resp. 右) から文字  $a$  を接続させる写像  $l_a$  (resp.  $r_a$ )
- (b) 語  $w \in \Sigma^*$  に対し、語に左 (resp. 右) から文字列  $w$  を接続させる写像  $l_w$  (resp.  $r_w$ ) (語の長さ  $|w|$  に関する帰納的定義で)
- (c) 語  $w \in \Sigma^*$  の後に接続すると  $L$  の元になる語全体の成す集合  $S_L(w)$  を与える写像  $S_L$

(2)  $M = (Q, \Sigma, \delta, s, F)$  : 有限オートマトン

- (a) 状態  $q \in Q$  にいる所から出発して語  $w \in \Sigma^*$  を読んだ後の状態  $\tilde{\delta}(q, w)$  を与える写像  $\tilde{\delta}$  (語の長さ  $|w|$  に関する帰納的定義で)
- (b) 特に、 $M$  が語  $w \in \Sigma^*$  を読んだ後の状態を与える写像  $\tilde{\delta}_0$
- (c)  $M$  が認識する言語  $L(M)$
- (d) より一般に、語  $w \in \Sigma^*$  を読んだ後に、続けて読めば受理される語全体の成す集合  $S_M(w)$  を与える写像  $S_M$

「非決定性」とは … あてずっぽう有り

状態の遷移先を一意に決めない  
(幾つかあって分岐していく)

→ どれかが受理すれば OK!!

- 手分けをして誰かが受理出来れば良い
- どの道を辿れば良いか知っていて、  
受理が検証出来れば良い

と考えることも出来る

## 非決定性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 … 状態の集合
- $\Sigma$  : 有限集合 … **alphabet**,  $\Sigma_\epsilon := \Sigma \cup \{\epsilon\}$
- $\delta : Q \times \Sigma_\epsilon \rightarrow \mathcal{P}(Q)$  : 遷移関数  
… 可能な遷移先全体の集合を与える
- $s \in Q$  … 初期状態
- $F \subset Q$  … 受理状態の集合

## 非決定性有限オートマトンによる語の受理

### 非決定性有限オートマトン

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

が、語  $w \in \Sigma^*$  を受理する



$$\exists a_1, a_2, \dots, a_n \in \Sigma_\varepsilon : w = a_1 a_2 \dots a_n$$

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

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

$L(M)$  :  $M$  が受理する語の全体

…  $M$  が認識する言語

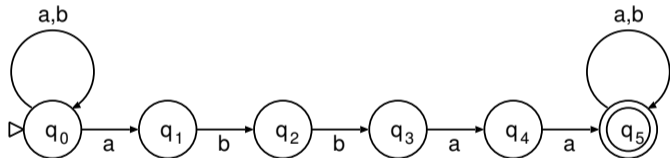


## 非決定性有限オートマトンによる語の受理

- $r_i \in \delta(r_{i-1}, \varepsilon)$  とは、  
「入力を読まずに  
状態  $r_{i-1}$  から状態  $r_i$  に移って良い」  
ということ
- $\delta(r_{i-1}, a_i) = \emptyset$  (矢印が出ていない)  
ということもある  
→ 受理されない分岐の  
行き止まりに入ってしまった  
→ 他の分岐が生きていれば問題無し

# 非決定性有限オートマトンの例

(状態遷移図による表示)



定理 :

L : 正規言語



L が或る非決定性有限オートマトンで  
認識される

これは比較的容易

正規言語の帰納的定義に沿って構成

## 正規言語を認識する NFA の構成

正規言語 :

- $L(\emptyset) = \emptyset$
- $L(\varepsilon) = \{\varepsilon\}$
- $L(a) = \{a\}$
- $L(R \cup S) = L(R) \cup L(S)$
- $L(R \circ S) = L(R) \circ L(S)$
- $L(R^*) = L(R)^*$

- (1) 言語  $L(\emptyset), L(\varepsilon), L(a)$  を認識する NFA を構成
- (2) 言語  $A, B$  を認識する NFA から、  
言語  $A \cup B, A \circ B, A^*$  を認識する NFA を構成

## $A \cup B$ を認識する非決定性有限オートマトンの構成

