

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
- Σ : 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

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



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

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

$L(M) : M$ 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”

語の演算

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

$$vw := a_1 \cdots a_k b_1 \cdots b_l \in \Sigma^*$$

: 連結・連接 (concatnation)

連接演算により Σ^* 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、

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

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

正規表現 (regular representation)

- 空集合記号 $\emptyset$ は正規表現
- 空列記号 ε は正規表現
- 各 alphabet $a \in \Sigma$ 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：

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

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
少し一般化の方が良い

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

「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
- Σ : 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

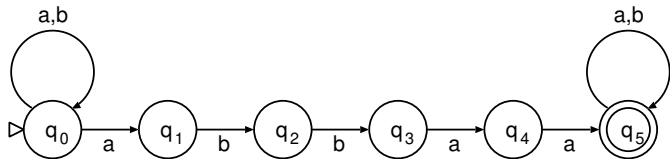
... M が認識する言語

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 が或る (決定性) 有限オートマトンで
認識される



L が或る非決定性有限オートマトンで
認識される

非決定性有限オートマトン M に対し、

$L(M)$ を認識する決定性有限オートマトン $\widetilde{M}$ が
構成できる

定理:

L : 正規言語



L が或る決定性有限オートマトンで
認識される



L が或る非決定性有限オートマトンで
認識される

正規言語:

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

言語 A, B を認識する

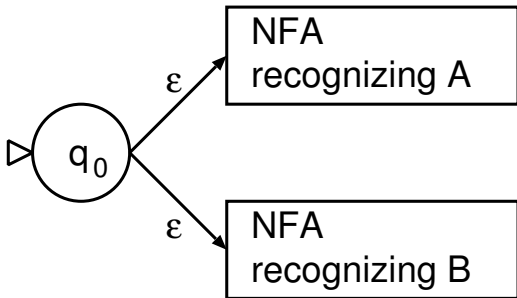
非決定性有限オートマトンから、

言語 $A \cup B, A \circ B, A^*$ を認識する

非決定性有限オートマトンが

構成できれば良い(それは比較的容易)

$A \cup B$ を認識する非決定性有限オートマトンの構成



有限オートマトンでの計算可能性問題

非決定性有限オートマトンで認識できない
言語が存在する!!
($\iff$ 正規でない言語が存在する)

例: $A = \{a^n b^n \mid n \geq 0\}$ (a と b との個数が同じ)

証明は部屋割り論法
(の一種の pumping lemma)
による

Pumping Lemma (注入補題・反復補題)

正規言語 A に対し、

$\exists n \in \mathbf{N} :$

$\forall w \in A, |w| \geq n :$

$\exists x, y, z \in \Sigma^* : w = xyz$

(1) $y \neq \varepsilon$

(2) $|xy| \leq n$

(3) $\forall k \geq 0 : xy^kz \in A$

より強力な計算モデルが必要



- プッシュダウンオートマトン
- チューリングマシン