アルゴリズム論 1
第 3 回: 正規表現と有限オートマトンの関係

関川 浩

2016/04/27

第 1 回から第 3 回の内容

言語理論とオートマトンの主題
無限集合である言語をいかに表現するか

一つの方法: 言語が満たすルールをうまく書いて, それを言語の表現とする手法. 文法がその代表

第 1 回: 正規表現というシステムを紹介

第 2 回: 有限オートマトンという表現法を紹介

第 3 回: 正規表現と有限オートマトン (見掛けはかなり違う) は言語の表現能力が等しいことを証明

有限オートマトン (正規表現) の能力の限界を説明

fa と正規表現の等価性の証明手順

等価性: 言語の表現能力が等しいこと

fa と正規表現の等価性 (定理 1) の証明手順:

1. dfa と nfa の等価性, nfa と εnfa の等価性を証明
   dfa は nfa で, nfa は εnfa なので, 以下を示せば十分
   - nfa が dfa で模倣できること (補題 1)
   - εnfa が nfa で模倣できること (補題 2)
2. fa と正規表現の等価性を証明. 以下を示す
   - 正規表現が εnfa で模倣できること (補題 3)
   - nfa が正規表現で模倣できること (補題 4)

nfa の dfa による模倣 (1/2)

補題 1

言語 L が nfa で認識可能なら, L は dfa で認識可能

証明

nfa $N = (K, \Sigma, \delta, s_0, F)$: L を認識

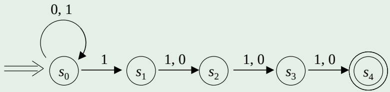
L を認識する dfa $D = (2^K, \Sigma, \delta', \{s_0\}, F')$ を構成

- D の状態は N の状態集合の部分集合
- $\delta'(S, a) \stackrel{\text{def}}{=} \bigcup_{s \in S} \delta(s, a) \qquad (S \subseteq K,\ a \in \Sigma)$
- $F' \stackrel{\text{def}}{=} \{S \mid S \subseteq K,\ S \cap F \neq \emptyset\}$

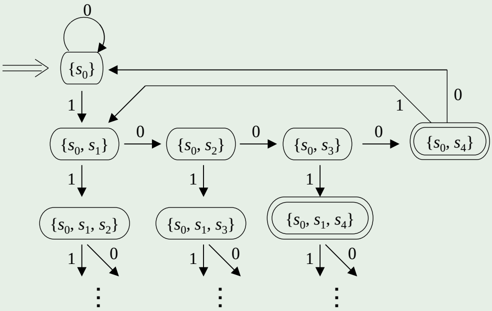
$\Longrightarrow$ D は N と同じ言語を認識 ■

nfa の dfa による模倣 (2/2)

nfa:



fa:



εnfa の nfa による模倣 (1/2)

補題 2

言語 L が εnfa で認識可能なら, L は nfa で認識可能

証明

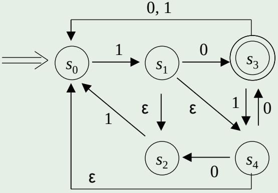
εnfa $E = (K, \Sigma, \delta, s_0, F)$: L を認識
L を認識する nfa $N = (K', \Sigma, \delta', s'_0, F')$ を構成
$S(s) \stackrel{\text{def}}{=} \{s\} \cup \{s$ から ε 遷移のみで到達できる状態 $\}$

- $K' \stackrel{\text{def}}{=} \{(s, S(s)) \mid s \in K\}$
- $s'_0 \stackrel{\text{def}}{=} (s_0, S(s_0))$
- $\delta'((s, S(s)), a) \stackrel{\text{def}}{=} \left\{(t, S(t)) \,\middle|\, t \in \bigcup_{s' \in S(s)} \delta(s', a)\right\}$
- $F' \stackrel{\text{def}}{=} \{(s, S(s)) \mid S(s) \cap F \neq \emptyset\}$

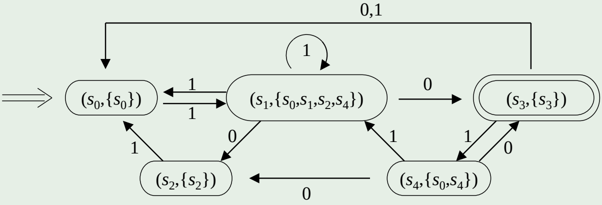
$\Longrightarrow N$ は E と同じ言語を認識 ■

εnfa の nfa による模倣 (2/2)

εnfa:



nfa:



正規表現と εnfa の関係 (1/5)

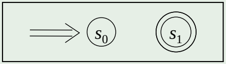
補題 3

言語 L が正規表現で表現可能なら, L は εnfa で認識可能

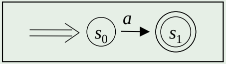
証明 (1/5)

L: アルファベット Σ 上の正規表現 R で表現

1. $R = \emptyset$ のとき, $L(R)$ $(= \emptyset)$ は下図の nfa で認識可能



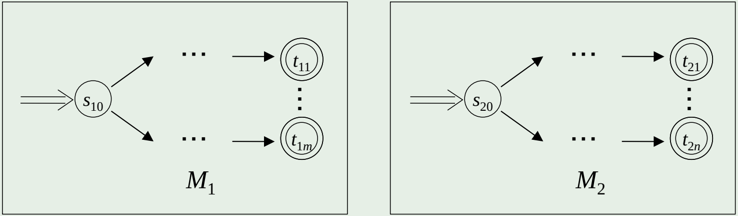
2. $R = a \in \Sigma$ のとき, $L(R)$ $(= \{a\})$ は下図の nfa で認識可能



正規表現と εnfa の関係 (2/5)

証明 (2/5)

3. R_1, R_2: 正規表現
   M_1, M_2: $L(R_1)$, $L(R_2)$ を認識する εnfa

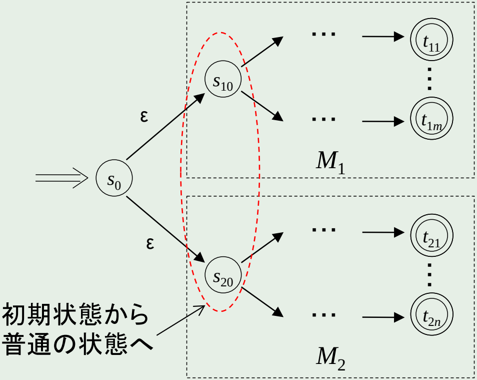


$L(R_1 + R_2)$, $L(R_1R_2)$, $L(R_1^*)$ を認識する εnfa を構成すればよい

正規表現と εnfa の関係 (3/5)

証明 (3/5)

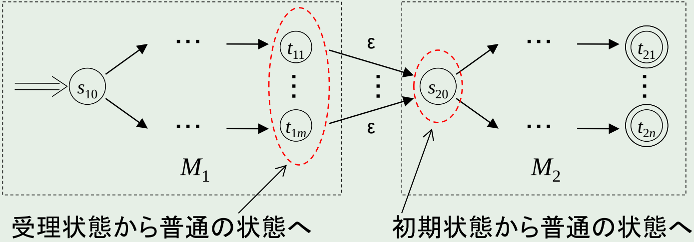
- $L(R_1 + R_2)$ $(= L(R_1) \cup L(R_2))$ を認識する εnfa:



正規表現と εnfa の関係 (4/5)

証明 (4/5)

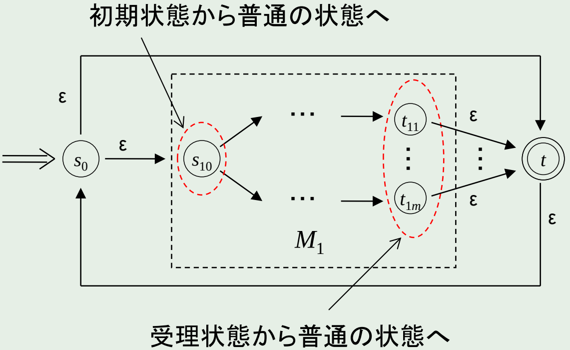
- $L(R_1R_2)$ $(= L(R_1) \cdot L(R_2))$ を認識する εnfa:



正規表現と εnfa の関係 (5/5)

証明 (5/5)

- $L(R_1^*)$ $(= L(R_1)^*)$ を認識する εnfa:



■

nfa と正規表現の関係 (1/5)

補題 4

言語 L が nfa で認識可能なら, L は正規表現によって表現可能

証明 (1/5)

nfa X が認識する言語を $L(X)$ と書く

L: nfa $N = (K, \Sigma, \delta, s_0, F)$ で認識可能

N における状態遷移を表す矢印の数による帰納法

- 矢印の数が 0 のとき
  - $s_0 \in F$ のとき $L(N) = \{\varepsilon\} = L(\emptyset^*)$
  - $s_0 \notin F$ のとき $L(N) = \emptyset = L(\emptyset)$

$\Longrightarrow$ 成立

nfa と正規表現の関係 (2/5)

証明 (2/5)

- 矢印の数が n のとき成立と仮定して $n+1$ のとき

$$x \in L(N) \iff s_0 \text{ から } x \text{ の記号にしたがって うまく矢印をたどれば受理状態に到達}$$

受理状態に到達する道筋をパスと呼ぶ (矢印の列で書ける)

α: N の一つの矢印 (s から s' への a による遷移とする)
N': N から α を取り除いた nfa

$x \in L(N)$ とし, そのことを表すパスの一つを p とする

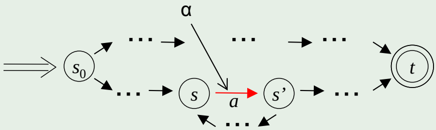
(a) p が α を含まないなら $x \in L(N')$

nfa と正規表現の関係 (3/5)

証明 (3/5)

(b) p が α をちょうど m 個 $(m \geq 1)$ 含むなら, p は,

- s_0 から α を通らずに s へ,
- 「s から α を通って s' へ, s' から α を通らずに s へ」を $m-1$ 回,
- s から α を通って s' へ, s' から α を通らずに受理状態に到達



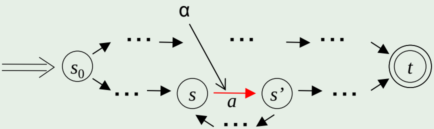
nfa と正規表現の関係 (4/5)

証明 (4/5)

$N'(t, G)$: N' の初期状態を t に, 受理状態集合を G に変えた nfa

(a), (b) より, $L(N)$ は以下と等しい

$$L(N') \cup \left(L(N'(s_0, \{s\})) \cdot (\{a\} \cdot L(N'(s', \{s\})))^* \cdot \{a\} \cdot L(N'(s', F))\right)$$



nfa と正規表現の関係 (5/5)

証明 (5/5)

$L(N)$ は以下と等しい

$$L(N') \cup \left(L(N'(s_0, \{s\})) \cdot (\{a\} \cdot L(N'(s', \{s\})))^* \cdot \{a\} \cdot L(N'(s', F))\right)$$

帰納法の仮定より，ある正規表現 R_1, R_2, R_3, R_4 が存在して，

$$L(N') = L(R_1), \qquad L(N'(s_0, \{s\})) = L(R_2),$$
$$L(N'(s', \{s\}) = L(R_3), \qquad L(N'(s', F)) = L(R_4).$$

よって，

$$\begin{aligned} L(N) &= L(R_1) \cup (L(R_2) \cdot (L(a) \cdot L(R_3))^* \cdot L(a) \cdot L(R_4)) \\ &= L(R_1) \cup (L(R_2) \cdot L((aR_3)^*) \cdot L(a) \cdot L(R_4)) \\ &= L(R_1) \cup L(R_2(aR_3)^* aR_4) \\ &= L(R_1 + R_2(aR_3)^* aR_4) \qquad \blacksquare \end{aligned}$$

fa と正規表現の等価性 (1/2)

定理 1

fa と正規表現は等価
($\Longleftrightarrow$ fa で認識可能な言語は正規表現で表現可能, 逆に, 正規表現で表現可能な言語は fa で認識可能)

証明

補題 3, 4 より成立 ■

fa と正規表現の等価性 (2/2)

定理 2

L_1, L_2: Σ 上の言語

1. L_1, L_2 が正規表現で表現可能なら, 以下も同様

$$L_1 \cup L_2, \quad L_1 \cap L_2, \quad \Sigma^* \setminus L_1, \quad L_1 L_2, \quad L_1^*$$

2. L_1, L_2 が fa で認識可能なら, 以下も同様

$$L_1 \cup L_2, \quad L_1 \cap L_2, \quad \Sigma^* \setminus L_1, \quad L_1 L_2, \quad L_1^*$$

証明

- 定理 1 (fa と正規表現の等価性)
- 第 1 回の定理 1
  (L_1, L_2 が正規表現で表現可能なら $L_1 \cup L_2$, $L_1 L_2$, L_1^* も同様)
- 第 2 回の定理 1 (L_1, L_2 が fa で認識可能なら $L_1 \cap L_2$ も同様)
  定理 3 (L が fa で認識可能なら $\Sigma^* \setminus L$ も同様)

より成立 ■

fa, 正規表現の言語表現能力

fa, 正規表現: 簡潔で分かりやすく, 実用性も高い

- fa: 状態遷移図の形でいろいろな場面で利用
- 正規表現: 文字列を表現する道具として UNIX の一部分

しかし, 言語表現能力はあまり高くない

- 理由: fa の状態が有限個しかないから
- たとえば, カッコ文 (第 1 回の例) は fa では認識できない

定義 (正規言語)

fa によって認識できる言語 (正規表現で表現できる言語) を
正規言語という

正規言語ではない言語の存在

定理 3

言語 $L = \{0^n 1^n \mid n \geq 0\}$ は正規言語ではない

証明

L がある fa $M = (K, \{0,1\}, \delta, s_0, F)$ で認識できたと仮定

n: M の状態数

$t_i \stackrel{\text{def}}{=} \delta(s_0, 0^i)$ $(i \geq 0)$

$\Longrightarrow$ t_0, t_1, ..., t_n の中には同じ状態が存在

$t_j = t_{j+k}$ $(k > 0)$ とする

$0^j 1^j$ は受理されるので, $\delta(t_j, 1^j) \in F$

$\Longrightarrow$ $\delta(t_{j+k}, 1^j) = \delta(t_j, 1^j) \in F$ より $0^{j+k} 1^j \in L$ となり矛盾 ■