アルゴリズム論 1

第 2 回: 有限オートマトン

関川 浩

2016/04/20

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

1. 有限オートマトン
   - 有限オートマトン (fa) の定義
   - fa が認識する言語
   - 直積オートマトン
   - 言語の補集合
2. 非決定性有限オートマトン
   - 非決定性有限オートマトン (nfa)
   - ε 入力付非決定性有限オートマトン (εnfa)

1. 有限オートマトン
2. 非決定性有限オートマトン

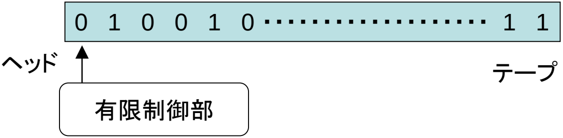
有限オートマトン (fa) の定義

定義 (有限オートマトン)

有限オートマトン (以下, fa と略): 五つ組 $M = (K, \Sigma, \delta, s_0, F)$

- K, Σ: 空ではない有限集合, $F \subseteq K$
- 状態: K の要素
  入力記号: Σ の要素
  受理状態: F の要素
  初期状態: K のある要素 s_0
- $\delta : K \times \Sigma \longrightarrow K$: 状態遷移関数

fa の概念図

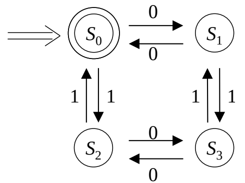


- テープと有限制御部からなる
- ヘッドがテープを左から右にスキャン
- テープはマス目に分かれ，各マス目に記号が書いてある (テープ全体で 1 個の列)
- テープをスキャン後，テープ上の列が対象の言語に属しているか否かを判定することによりその言語を表現

fa の例

例: 有限オートマトン M_1

- 状態: s_0, s_1, s_2, s_3
- 入力記号: 0, 1
- 初期状態 ($\Longrightarrow$ で表わす): s_0
- 受理状態 (二重丸で表わす): s_0
- 状態遷移関数: 状態間の矢印で表現



状態遷移図

たとえば, s_0 から出る矢印で,

$$\delta(s_0, 0) = s_1, \quad \delta(s_0, 1) = s_2$$

を表現

fa が認識する言語

定義 (状態遷移関数の拡張)

状態遷移関数 δ の定義域を $K \times \Sigma$ から $K \times \Sigma^*$ に拡張

- 任意の $s \in K$, $a \in \Sigma$, $x \in \Sigma^*$ に対して,

$$\delta(s, \varepsilon) \stackrel{\text{def}}{=} s, \quad \delta(s, xa) \stackrel{\text{def}}{=} \delta(\delta(s, x), a)$$

例: 前スライドの M_1

$$\begin{aligned} \delta(s_0, 110) &= \delta(\delta(s_0, 11), 0) = \delta(\delta(\delta(s_0, 1), 1), 0) \\ &= \delta(\delta(s_2, 1), 0) = \delta(s_0, 0) = s_1 \end{aligned}$$

定義 (言語の受理, 認識)

- $\delta(s_0, x) \in F$ なら, 列 x は受理されるという
- M が受理する列の全体を M が認識する言語という

例題 1 (1/2)

例題 1 (M_1 が認識する言語)

列 $x \in \{0,1\}^*$ に対し, $\delta(s_0, x) = s$ としたとき, 以下を示せ

$$s = s_0 \iff x \text{ において 0 と 1 の個数はともに偶数}$$
$$s = s_1 \iff x \text{ において 0 の個数のみ奇数}$$
$$s = s_2 \iff x \text{ において 1 の個数のみ奇数}$$
$$s = s_3 \iff x \text{ において 0 と 1 の個数はともに奇数}$$

よって, M_1 が認識する言語は, 0, 1 ともに偶数個である列全体

証明 (1/2)

列 x の長さによる帰納法

- $|x| = 0$ のとき ($x = \varepsilon$ のとき), $\delta(s_0, \varepsilon) = s_0$
- ε の 0 と 1 の個数はともに 0 個で偶数

$\Longrightarrow$ 長さ 0 のときは成立

例題 1 (2/2)

証明 (2/2)

- 長さ n のとき成立すると仮定, 列 y ($|y| = n + 1$) を考える
  $\Longrightarrow y = xa$, ただし, $x \in \{0, 1\}^*$, $|x| = n$, $a \in \{0, 1\}$

  場合分け (0, 1 の偶奇が x で 4 通り, a で 2 通り計 8 通り)

  - x において 0 のみ奇数個, かつ, $a = 0$
    $\Longrightarrow y$ では 0, 1 ともに偶数個

    $$\delta(s_0, y) = \delta(s_0, x0) = \delta(\delta(s_0, x), 0) = \delta(s_1, 0) = s_0$$

  - 残りの 7 通りも同様に成立 ■

直積オートマトン

定義 (直積オートマトン)

アルファベットが等しい二つの fa

$$M_1 = (K_1, \Sigma, \delta_1, s_{01}, F_1), \qquad M_2 = (K_2, \Sigma, \delta_2, s_{02}, F_2)$$

の直積とは, 二つの fa を並列に動作させたシステムを模擬した fa

- 状態: 二つの fa の状態の対 $(s_1, s_2) \in K_1 \times K_2$
- 状態 (s_1, s_2) で $a \in \Sigma$ を読むと $(\delta_1(s_1, a), \delta_2(s_2, a))$ に遷移
- 初期状態, 受理状態は目的に応じて指定

直積オートマトンと言語の対応

定理 1

Σ 上の言語 L_1, L_2 が fa で認識可能なら $L_1 \cap L_2$ も fa で認識可能

証明

$M_i = (K_i, \Sigma, \delta_i, s_{0i}, F_i)$: L_i を認識する fa ($i = 1$, 2)

M: M_1 と M_2 の直積オートマトン

- 初期状態は (s_{01}, s_{02})
- (s_1, s_2) が受理状態 $\iff s_1 \in F_1$, $s_2 \in F_2$

M が $L_1 \cap L_2$ を認識するのは明らか ■

最少状態数の fa (1/4)

- fa の状態数 (有限): 記憶容量と考えられる
  $\Longrightarrow$ 同じ言語を認識するなら状態数は少ない方が効率的
- 最少状態数の fa を得るには?
  $\Longrightarrow$ 冗長な状態があれば削除
  - 明らかに冗長な状態: 初期状態から到達できない状態
  - そのほかにも冗長な状態がある

定義 (状態の等価性)

fa $M = (K, \Sigma, \delta, s_0, F)$ の二つの状態 s_1 と s_2 が等価とは,
以下の条件を満たすこと

$$\forall x \in \Sigma^* \text{ に対して, } \delta(s_1, x) \in F \Longleftrightarrow \delta(s_2, x) \in F$$

最少状態数の fa (2/4)

定理 2

仮定: 初期状態から到達できない状態は存在しない

1. fa M に等価な 2 状態が存在すれば, 認識する言語が M と同じで, 状態数が M より 1 少ない fa M' が存在
2. fa M と同じ言語を認識して状態数がより少ない fa M' が存在すれば, M には等価な 2 状態が存在

証明 (1/2)

1. $s_1 \neq s_2$: 等価な 2 状態
    - 状態 s から s_1 への遷移があれば, すべて s_2 への遷移と変更
    - s_1 を削除すれば, 状態数が 1 少ない fa M' が構成可能

最少状態数の fa (3/4)

証明 (2/2)

2. 以下を仮定して矛盾を出す
   (1) M と M' は同じ言語を認識
   (2) M の状態数は M' の状態数より多い
   (3) M の任意の二つの状態は等価ではない

$M = (K, \Sigma, \delta, s_0, F)$, $M' = (K', \Sigma, \delta', s_0', F')$ とすると, (2) より以下を満たす列 x_1, x_2 ($x_1 \neq x_2$) が存在 (鳩の巣原理)

$$s_1 = \delta(s_0, x_1) \neq \delta(s_0, x_2) = s_2, \quad \delta'(s_0', x_1) = \delta'(s_0', x_2)$$

- s_1 と s_2 は等価ではないので, ある列 y が存在して, M では x_1y と x_2y のうち一方のみが受理
- M' では x_1y と x_2y はともに受理か非受理

$\Longrightarrow$ (1) に矛盾 ■

最少状態数の fa (4/4)

- 定理 2 がいっていること:
  仮定「初期状態から到達できない状態は存在しない」の下で,
  M が最少状態数 $\iff$ M に等価な 2 状態は存在しない
- よって, 初期状態から到達できない状態を削除の上,

  等価な状態の対があれば, 定理 2 (1) の証明中に述べた方法で一方の状態を削除,

  を繰り返せば, 最少状態数の fa が得られる

問題

2 状態が等価か否かを判定せよ

定義通りでは無限個の列を調べることが必要 $\Longrightarrow$ 不可能

2 状態 s_1, s_2 が等価か否かの判定法

1. M と M の直積オートマトンを構成
    - 初期状態: (s_1, s_2)
    - (s_3, s_4) が受理状態 $\overset{\text{def}}{\Longleftrightarrow}$ s_3, s_4 がともに M の受理状態
2. 初期状態から到達できない状態を削除
3. 残った対 (すべて初期状態から到達可能) (s_3, s_4) をすべてチェック (有限個だから可能)
    - 一方が受理状態, 他方が非受理状態である対が存在すれば s_1 と s_2 は非等価
    - そうでなければ等価

言語の補集合

定理 3

Σ 上の言語 L が fa で認識できるなら, L の補集合 $\Sigma^* \setminus L$ も fa で認識できる

証明

- M: L を認識する fa
- M': M の受理状態と非受理状態を入れ換えた fa
  $\Longrightarrow M'$ は $\Sigma^* \setminus L$ を認識する fa ■

1. 有限オートマトン
2. 非決定性有限オートマトン

非決定性有限オートマトン (nfa) の定義

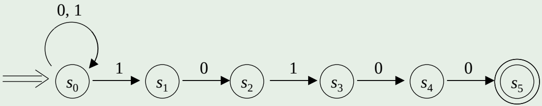
非決定性有限オートマトン (以下, nfa と略) $M = (K, \Sigma, \delta, s_0, F)$

- 状態遷移関数 δ のみが fa の場合と異なる
  $\delta : K \times \Sigma \longrightarrow 2^K$ (K のべき集合 (すべての部分集合の集合))

注意

今までの fa を, 決定性有限オートマトン, 決定性 fa, dfa と呼ぶこともある

例



$$\delta(s_0, 1) = \{s_0, s_1\}, \qquad \delta(s_1, 1) = \emptyset$$

nfa が認識する言語

定義 (状態遷移関数の拡張)

δ の定義域を $K \times \Sigma$ から $K \times \Sigma^*$ に拡張

- 任意の $s \in K$, $a \in \Sigma$, $x \in \Sigma^*$ に対して,

$$\delta(s, \varepsilon) \stackrel{\text{def}}{=} \{s\},$$

$$\delta(s, xa) \stackrel{\text{def}}{=} \{q \mid \exists p \in K \text{ s.t. } p \in \delta(s, x) \text{ かつ } q \in \delta(p, a)\}$$

例

前スライドの例では, $\delta(s_0, 101) = \{s_0, s_1, s_3\}$

定義 (言語の受理, 認識)

- $\delta(s_0, x) \cap F \neq \emptyset$ なら, 列 x は受理されるという
- M が受理する列の全体を M が認識する言語という

例

前スライドの nfa が認識する言語は, 10100 で終る列の全体

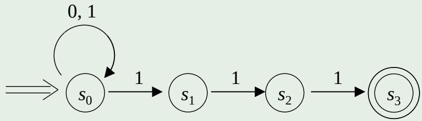
例題 2

例題 2

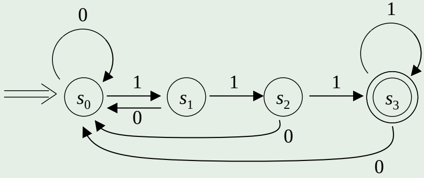
言語 $\{x111 \mid x \in \{0,1\}^*\}$ を認識する nfa と fa を与えよ

解答

- nfa



- fa



例題 3

例題 3

例題 2 の言語の補集合を認識する nfa を求めよ

解答

定理 3 の証明を利用

$\Longrightarrow$ fa も nfa なので (fa は nfa の特別な場合), 前スライドの fa の受理状態と非受理状態を入れ換え

注意

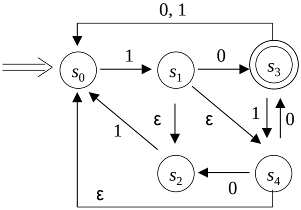
fa ではない nfa に対して定理 3 の証明を利用するのは不可

- この例では s_0 が受理状態に $\Longrightarrow$ $\{0,1\}^*$ のすべての列を受理

ε 入力付非決定性有限オートマトン (εnfa)

- 今までの fa の 1 ステップは以下の 3 動作
  - 入力を読む
  - ヘッドを 1 コマ右に動かす
  - 状態を変える
- εnfa では入力記号を読まないことも可
  (ヘッドを動かさず状態遷移可)
- 形式的には状態遷移関数 δ が

$$\delta : K \times (\Sigma \cup \{\varepsilon\}) \longrightarrow 2^K$$



様相 (1/2)

定義 (様相)

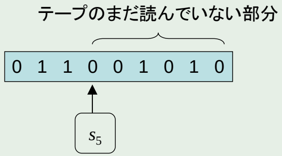
様相 (x, p): アルファベット Σ 上の列 x と状態 p の対

気持ち: オートマトンの動作途中の状況

- x: テープのまだ読んでいない部分の列
- p: そのときの状態

例

図の状況を様相 $(001010, s_5)$ で表す



様相が与えられればオートマトンの今後の動作が決まる

様相 (2/2)

定義 ($\Rightarrow$, $\overset{*}{\Rightarrow}$)

- 様相 $c_1 = (x_1, p_1)$ と $c_2 = (x_2, p_2)$ は, 条件 (1) または (2) を満たすとき, c_1 から c_2 へ 1 ステップで移れるといい, $c_1 \Rightarrow c_2$ と書く
  (1) $x_1 = x_2$ かつ $p_2 \in \delta(p_1, \varepsilon)$
  (2) $x_1 = ax_2$ かつ $p_2 \in \delta(p_1, a)$
- $c_0 \Rightarrow c_1 \Rightarrow \cdots \Rightarrow c_k$ のとき, c_0 から c_k へ何ステップかで移れるといい, $c_0 \overset{*}{\Rightarrow} c_k$ と書く ($k = 0$ でもよい)

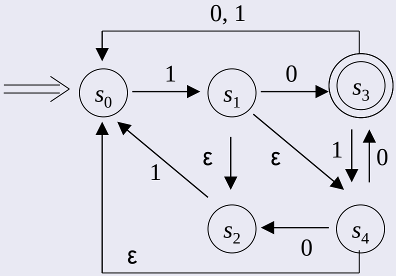
定義 (列の受理)

列 x は, 初期状態 s_0, ある受理状態 s_F に対し, $(x, s_0) \overset{*}{\Rightarrow} (\varepsilon, s_F)$ のとき, 受理されるという

例題 4

例題 4

下図のオートマトンで列 110 が受理されることを示せ



解答

$$(110, s_0) \Rightarrow (10, s_1) \Rightarrow (10, s_4) \Rightarrow (10, s_0) \Rightarrow (0, s_1) \Rightarrow (\varepsilon, s_3)$$