

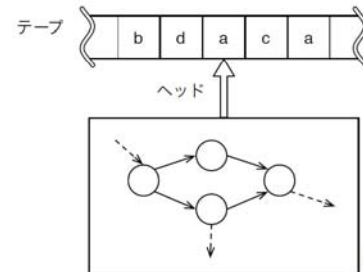
ソフトウェアモデル論(2012年度)
第3回・2012/10/12

桑原 寛明
情報理工学部 情報システム学科

オートマトンの概念図

(復習)

- 順序機械を一般的に記述するモデル
 - 状態と入力から出力を計算



ソフトウェアモデル論(2012/10/12)

2

決定性有限オートマトン (DFA)

(復習)

$$M = (Q, \Sigma, \delta, q_0, F)$$

Q : 状態の有限集合 ($\neq \emptyset$)

Σ : 入力記号の有限集合

δ : 状態遷移関数 ($Q \times \Sigma \rightarrow Q$)

q_0 : 初期状態 ($\in Q$)

F : 受理状態の集合 ($\subseteq Q$)

ソフトウェアモデル論(2012/10/12)

3

DFAの例

(復習)

- ab の繰返しを受理するDFA

$$Q : \{S_0, S_1, S_2\}$$

$$\Sigma : \{a, b\}$$

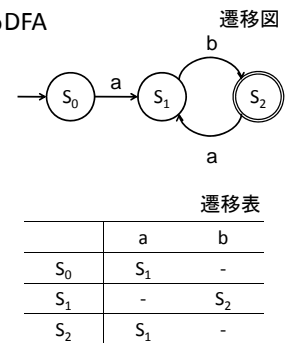
$$\delta : (S_0, a) \rightarrow S_1,$$

$$(S_1, b) \rightarrow S_2,$$

$$(S_2, a) \rightarrow S_1$$

$$q_0 : S_0$$

$$F : \{S_2\}$$



ソフトウェアモデル論(2012/10/12)

4

非決定性有限オートマトン (NFA)

$$M = (Q, \Sigma, \delta, q_0, F)$$

Q : 状態の有限集合 ($\neq \emptyset$)

Σ : 入力記号の有限集合

δ : 状態遷移関数 ($Q \times \Sigma \rightarrow 2^Q$)

q_0 : 初期状態 ($\in Q$)

F : 受理状態の集合 ($\subseteq Q$)

ソフトウェアモデル論(2012/10/12)

5

NFAにおける語の受理

- NFAでは遷移先が2つ以上の場合がある
 - e.g. $\delta(q, a) = \{q', q''\}$
- 入力された語に対して遷移列も複数ある
- 受理状態で終了する遷移列が1つでもあれば受理する

ソフトウェアモデル論(2012/10/12)

6

NFAの例

- a の1回以上の繰返しを受理するNFA

$Q : \{S_0, S_1\}$
 $\Sigma : \{a\}$
 $\delta : (S_0, a) \rightarrow \{S_0, S_1\}$
 $q_0 : S_0$
 $F : \{S_1\}$

遷移図

	a
S ₀	S ₀ , S ₁
S ₁	-

ソフトウェアモデル論(2012/10/12) 7

DFAとNFAの等価性

- DFAはNFAの特殊な場合である
 - DFAは遷移先が一意に決められるNFA
- NFAをシミュレートするDFAを作ることができる
 - 受理言語が同じ
- 受理言語が同じであればNFAの方が作りやすいことが多い

ソフトウェアモデル論(2012/10/12) 8

NFAをシミュレートするDFAの例

NFA → DFA

NFA
 $Q : \{S_0, S_1\}$
 $\Sigma : \{a\}$
 $\delta : (S_0, a) \rightarrow \{S_0, S_1\}$
 $q_0 : S_0$
 $F : \{S_1\}$

DFA
 $Q : \{\emptyset, \{S_0\}, \{S_1\}, \{S_0, S_1\}\}$
 $\Sigma : \{a\}$
 $\delta : (\{S_0\}, a) \rightarrow \{S_0, S_1\}$
 $\quad (\{S_0, S_1\}, a) \rightarrow \{S_0, S_1\}$
 $q_0 : \{S_0\}$
 $F : \{\{S_1\}, \{S_0, S_1\}\}$

ソフトウェアモデル論(2012/10/12) 9

NFAをシミュレートするDFAの例

NFA → DFA

ソフトウェアモデル論(2012/10/12) 10

NFAをシミュレートするDFAの作り方

- NFA: $M_N = (Q_N, \Sigma, \delta_N, q_0^N, F_N)$
- DFA: $M_D = (Q_D, \Sigma, \delta_D, q_0^D, F_D)$
- M_D の状態集合は M_N の状態集合のべき集合
 - $Q_D = 2^{Q_N}$
- M_N と M_D の記号の集合は同じ
- M_D の初期状態は M_N の初期状態のみからなる状態
 - $q_0^D = \{q_0^N\}$

ソフトウェアモデル論(2012/10/12) 11

NFAをシミュレートするDFAの作り方

- M_D の最終状態は M_N の最終状態を1つでも含む状態すべて
 - $F_D = \{q_D \mid q_D \in Q_D \text{ かつ } q_D \cap F_N \neq \emptyset\}$
- M_D の各状態の遷移先は、その状態が含む M_N の状態の遷移先すべてを含む M_D の状態
 - $\delta_D(\{q_1^N, \dots, q_i^N\}, a) = \delta_N(q_1^N, a) \cup \dots \cup \delta_N(q_i^N, a)$

ソフトウェアモデル論(2012/10/12) 12

NFAをシミュレートするDFAの作り方

ソフトウェアモデル論(2012/10/12) 13

有限オートマトンで受理できない言語の例

- $\{ a^n b^n \mid n \text{は} 0 \text{以上の整数} \}$
- $\{ a^i \mid j=i^2 \text{で} i \text{は} 1 \text{以上の整数} \}$

• 有限オートマトンで受理できるのはどのような言語だろうか？

ソフトウェアモデル論(2012/10/12) 14

正規表現 (RE)

アルファベット Σ 上の正規表現

1. 空列 ϵ 、空集合 \emptyset 、記号 $a \in \Sigma$ は正規表現
2. P, Q が正規表現ならば
 - $(P + Q)$ 選択
 - $(P \cdot Q)$ 接続(通常は \cdot を省略する)
 - (P^*) 繰返し

は正規表現

3. 1.と2. を有限回適用して生成されるものだけが正規表現

ソフトウェアモデル論(2012/10/12) 15

正規表現が表す語の集合

1. 正規表現 ϵ は集合 $\{\epsilon\}$ を表す
2. 正規表現 \emptyset は空集合 \emptyset を表す
3. 正規表現 a は集合 $\{a\}$ を表す
4. 正規表現 P, Q が表す語の集合をそれぞれ P, Q とすると
 - $(P + Q)$ は $\{w \mid w \in P \vee w \in Q\} = P \cup Q$
 - $(P \cdot Q)$ は $\{v \cdot w \mid v \in P \wedge w \in Q\} = P \cdot Q$
 - (P^*) は $\{\epsilon\} \cup P \cup P \cdot P \cup \dots$ を表す

ソフトウェアモデル論(2012/10/12) 16

正規表現と語の集合の例 ($\Sigma=\{a,b\}$)

<ul style="list-style-type: none"> • 正規表現 - $a+b$ - $aabb$ - $a(a+b)b$ - a^* - ab^*a - $a(a+b)^*b$ 	<ul style="list-style-type: none"> • 語の集合 - $\{a, b\}$ - $\{aabb\}$ - $\{aab, abb\}$ - $\{\epsilon, a, aa, aaa, aaaa, \dots\}$ - $\{aa, aba, abba, abbba, \dots\}$ - $\{ab, aab, abb, aaab, aabb, abab, abbb, aaaab, \dots\}$
---	--

ソフトウェアモデル論(2012/10/12) 17

正規表現の変形規則

- $R(S+T) = RS + RT$
- $(R+S)T = RT + ST$
- $R^*(R+\epsilon) = R^* = (R+\epsilon)R^*$
- $R^* + R^n = R^*$
- $(R+\epsilon)^* = R^*$
- $RS^* + R = RS^*$
- $\epsilon^* = \epsilon$
- $R\epsilon = R = \epsilon R$
- $R\emptyset = \emptyset = \emptyset R$
- $R + \emptyset = R = \emptyset + R$

ソフトウェアモデル論(2012/10/12) 18

有限オートマトンと正規表現の等価性

- RE \rightarrow NFA \rightarrow DFA
 - REが表す言語を受理するNFAを作る
 - NFAをシミュレートするDFAを作る
- NFA \rightarrow DFA \rightarrow RE
 - NFAをシミュレートするDFAを作る
 - DFAをREに変換する

ソフトウェアモデル論(2012/10/12)

19

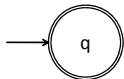
正規表現から有限オートマトンへの変換

1. 正規表現 ϵ 、 \emptyset 、 a が表す言語を受理する有限オートマトンを作る
2. 正規表現 $(P+Q)$ 、 $(P \cdot Q)$ 、 (P^*) が表す言語を受理する有限オートマトンは、 P や Q が表す言語を受理する有限オートマトンを組合せて作る

ソフトウェアモデル論(2012/10/12)

20

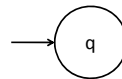
$\{\epsilon\}$ を受理する有限オートマトン



ソフトウェアモデル論(2012/10/12)

21

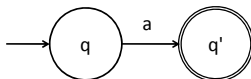
\emptyset を受理する有限オートマトン



ソフトウェアモデル論(2012/10/12)

22

$\{a\}$ を受理する有限オートマトン



ソフトウェアモデル論(2012/10/12)

23