

ソフトウェアモデル論(2010年度) 第4回・2010/10/18

桑原 寛明
情報理工学部 情報システム学科

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(2010/10/18)

2

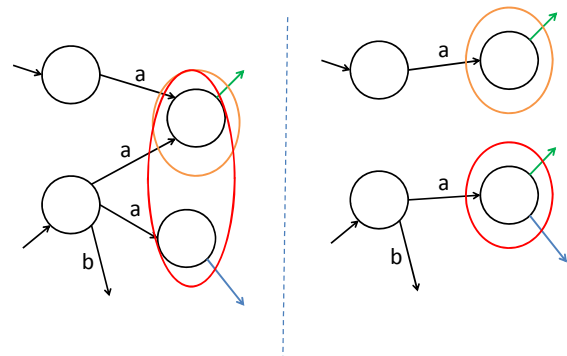
NFAをシミュレートするDFAの作り方

- 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(2010/10/18)

3

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(2010/10/18)

4

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 \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(2010/10/18)

5

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 $\{w \mid w \in P \vee w \in Q\}$
 - $(P \cdot Q)$ 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(2010/10/18)

6

正規表現と語の集合の例($\Sigma=\{a,b\}$)

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

ソフトウェアモデル論(2010/10/18)

7

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(2010/10/18)

8

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(2010/10/18)

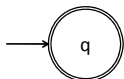
9

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

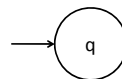
ソフトウェアモデル論(2010/10/18)

10

 $\{\epsilon\}$ を受理する有限オートマトン

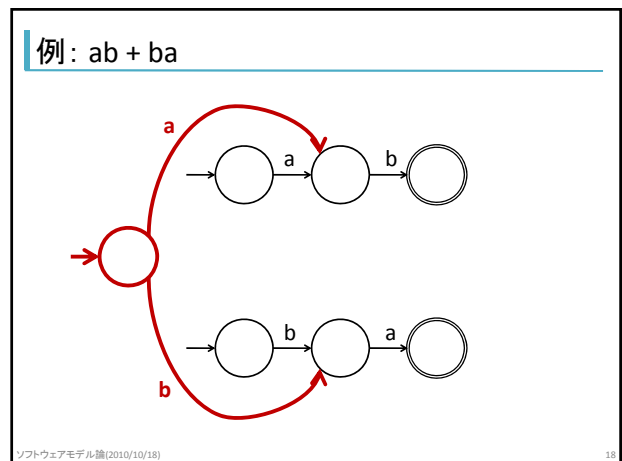
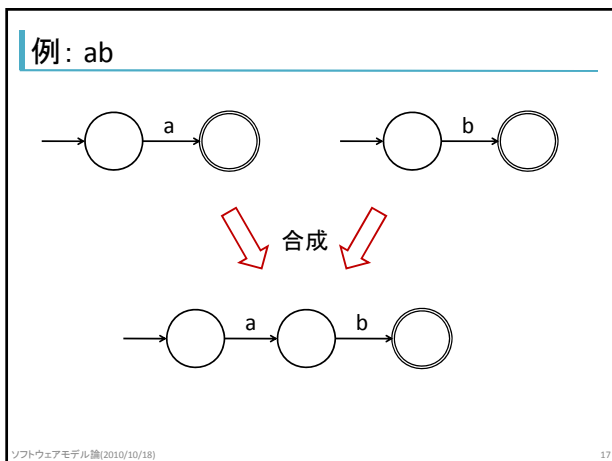
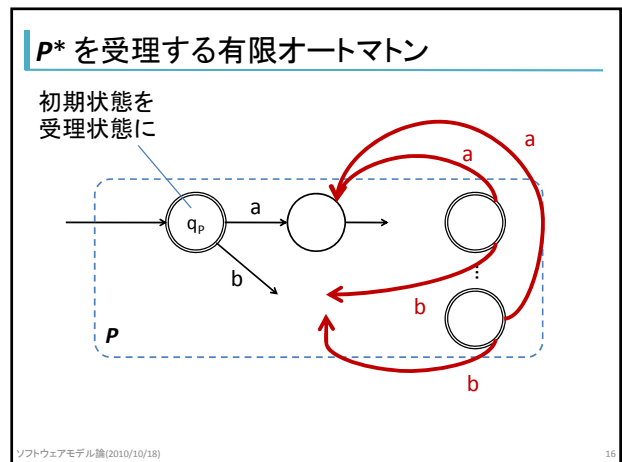
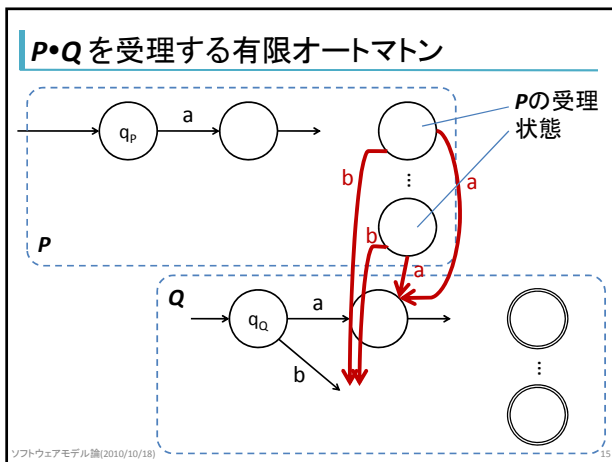
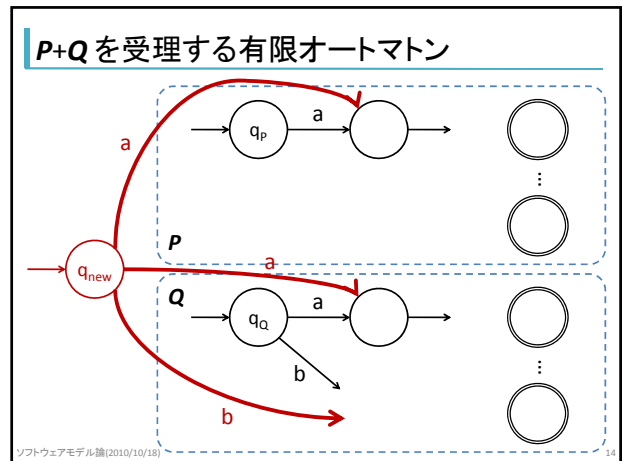
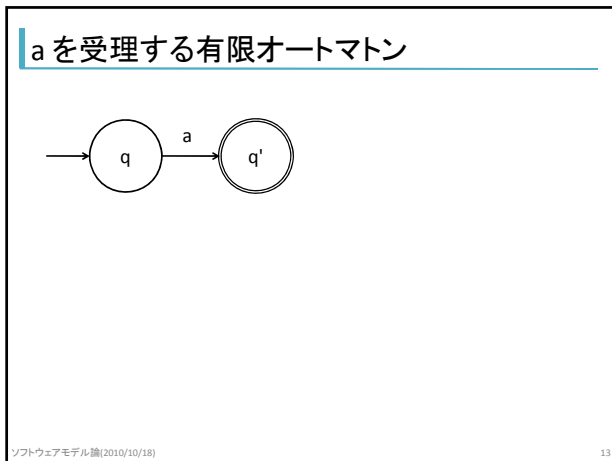
ソフトウェアモデル論(2010/10/18)

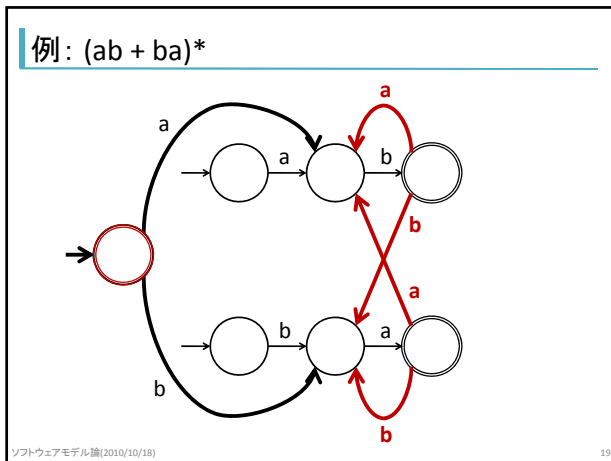
11

 \emptyset を受理する有限オートマトン

ソフトウェアモデル論(2010/10/18)

12





有限オートマトンから正規表現への変換

- 有限オートマトン $M = (\{1, \dots, n\}, \Sigma, \delta, 1, F)$
- r_{ij}^k を求める
 - 状態 i から状態 j へ k 以下の状態のみを通過して到達する記号列の正規表現
 - r_{ij}^0 から順に帰納的に
- $r_{1f_1}^n + \dots + r_{1f_1}^n$ が初期状態から受理状態へ到達する記号列の正規表現
 - $\{f_1, \dots, f_l\}$ は受理状態の集合

ソフトウェアモデル論(2010/10/18) 20

r_{ij}^0 を求める

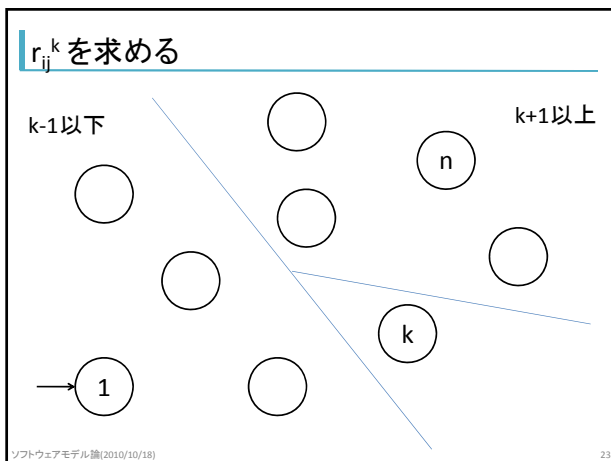
- 状態 i から状態 j へ直接到達
- $\delta(i, a) = j$ を満たす a の集合を $\{a_1, \dots, a_l\}$ とする
 - そのような a がなければ \emptyset
- $i \neq j$ ならば $r_{ij}^0 = a_1 + \dots + a_l$
- $i = j$ ならば $r_{ij}^0 = a_1 + \dots + a_l + \epsilon$

ソフトウェアモデル論(2010/10/18) 21

r_{ij}^k を求める

- $r_{ij}^k = r_{ik}^{k-1} (r_{kk}^{k-1})^* r_{kj}^{k-1} + r_{ij}^{k-1}$
- 状態 k を通過するかしないかの2択
 - 通過する
 - i から初めて k に到達
 - $k-1$ 以下のみ通過してまた k に到達 (を繰り返す)
 - k から $k-1$ 以下を通過して j に到達
 - 通過しない
 - $k-1$ 以下の状態のみ通過

ソフトウェアモデル論(2010/10/18) 22



例

$r_{12}^1 = r_{11}^0 r_{11}^0 r_{12}^0 + r_{12}^0$ $= \epsilon \epsilon a + a$ $= a$	$r_{23}^1 = r_{21}^0 r_{11}^0 r_{13}^0 + r_{23}^0$ $= \emptyset \epsilon \emptyset + b$ $= b$	$r_{13}^2 = r_{12}^1 r_{22}^1 r_{23}^1 + r_{13}^1$ $= a \epsilon b + \emptyset$ $= ab$
$r_{13}^1 = r_{11}^0 r_{11}^0 r_{13}^0 + r_{13}^0$ $= \epsilon \epsilon \emptyset + \emptyset$ $= \emptyset$	$r_{32}^1 = r_{31}^0 r_{11}^0 r_{12}^0 + r_{32}^0$ $= \emptyset \epsilon a + \emptyset$ $= \emptyset$	$r_{33}^2 = r_{32}^1 r_{22}^1 r_{23}^1 + r_{33}^1$ $= \emptyset \epsilon \epsilon + \epsilon$ $= \epsilon$
$r_{22}^1 = r_{21}^0 r_{11}^0 r_{12}^0 + r_{22}^0$ $= \emptyset \epsilon a + \epsilon$ $= \epsilon$	$r_{33}^1 = r_{31}^0 r_{11}^0 r_{13}^0 + r_{33}^0$ $= \emptyset \epsilon \emptyset + \epsilon$ $= \epsilon$	$r_{13}^3 = r_{12}^2 r_{22}^2 r_{23}^2 + r_{13}^2$ $= ab \epsilon \epsilon + ab$ $= ab$

ソフトウェアモデル論(2010/10/18) 24

正規言語

- 正規文法が生成する言語
- 有限オートマトンが受理できる言語
- 正規表現で表現できる言語

ある言語が正規言語か否か判定するにはどうすればよいか