2. 有限オートマトン(2): (テキスト2.3.5~2.3.7,2.5)

プレースメントテストの結果は教務課まで

前回の復習

– DFA $A_D=(Q_D, \Sigma, \delta_D, q_D, F_D)$ によって受理される言語

$L(A_D)=\{ w \mid \hat{\delta}_D(q_D,w) \in F_D\}$

$\delta_D: Q \times \Sigma \to Q$

次の状態はいつでも一意的に決まる

– NFA $A_N=(Q_N, \Sigma, \delta_N, q_N, F_N)$ によって受理される言語

$L(A_N)=\{ w \mid \hat{\delta}_N(q_N,w) \cap F_N \neq \Phi\}$

$\delta_N: Q \times \Sigma \to 2^Q$

次の状態は一意的に決まらず、複数の状態の集合となる

2. 有限オートマトン(2)

2.3.5. 決定性と非決定性の有限オートマトンの等価性

定理: NFAで受理できる言語のクラスと、DFAで受理できる言語のクラスは一致する。

おまけ: ‘集合の集合’ のことは特にクラス(Class)または族(Family)と呼ぶ。

2. 有限オートマトン(2)

2.3.5. 決定性と非決定性の有限オートマトンの等価性

証明: NFAで受理できる言語のクラスNと、DFAで受理できる言語のクラスDが一致することを示す。

- $D \subseteq N$は定義より明らかなので、$N \subseteq D$を示せばよい。
- 任意の言語 $L \in N$ が $L \in D$ となることを示せばよい。

ある言語 L が $L \in N$ であったとする。このとき、Lを受理する NFA $A_L=(Q_N, \Sigma, \delta_N, q_N, F_N)$ が存在する。
A_Lと同じ言語を受理する DFA A_L' を構成する。

2. 有限オートマトン(2)

証明: NFAで受理できる言語のクラスNと、DFAで受理できる言語のクラスDが一致することを示す。
$L \in N$ を受理する NFA $A_L=(Q_N, \Sigma, \delta_N, q_N, F_N)$ が存在する。
A_Lと同じ言語を受理する DFA A_L' を構成する。

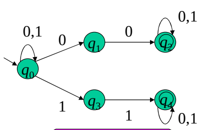
証明の直感的アイデア:
- DFAは状態がいつも1つだけ決まっている。
- NFA は状態の集合が入力に応じて変化する。

→NFAの状態の集合をDFAの1つの集合とみなす!!
サブセット構成(Subset construction)

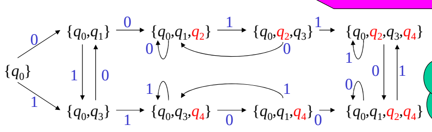
2. 有限オートマトン(2)

例:

δ	0	1
q_0	$\{q_0,q_1\}$	$\{q_0,q_3\}$
q_1	$\{q_2\}$	Φ
q_2	$\{q_2\}$	$\{q_2\}$
q_3	Φ	$\{q_4\}$
q_4	$\{q_4\}$	$\{q_4\}$



下図はDFAと見なせる



初期状態から到達できない状態もあるけど…

2. 有限オートマトン(2)

証明: NFAで受理できる言語のクラスNと、DFAで受理できる言語のクラスDが一致することを示す。
$L \in N$ を受理する NFA $A_L=(Q_N, \Sigma, \delta_N, q_N, F_N)$ が存在する。
A_Lと同じ言語を受理する DFA A_L' を次のように構成する。

$$A_L' = \{2^{Q_N}, \Sigma, \delta_D, \{q_N\}, F_D\}$$

– 状態集合は A_L の状態集合Q_Nの集合族
– 初期状態 $\{q_N\}$ は ‘q_N だけからなる集合’ であり、q_N ではない
– δ_D と F_D を定義する必要がある。

2. 有限オートマトン(2)

証明:

$L \in N$ を受理する NFA $A_L=(Q_N, \Sigma, \delta_N, q_N, F_N)$ が存在する。

A_Lと同じ言語を受理する DFA A_L' を次のように構成する。

$$A_L' = \{2^{Q_N}, \Sigma, \delta_D, \{q_N\}, F_D\}$$

– δ_D と F_D を定義する必要がある。

$$F_D = \{S \mid S \in 2^{Q_N}, S \cap F_N \neq \phi\}$$

2. 有限オートマトン(2)

証明:

$L \in N$ を受理する NFA $A_L=(Q_N, \Sigma, \delta_N, q_N, F_N)$ が存在する。

A_Lと同じ言語を受理する DFA A_L' を次のように構成する。

– δ_D と F_D を定義する必要がある。

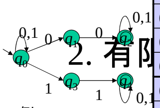
	0	1
Φ	Φ	Φ
$\{q_0\}$	$\{q_0,q_1\}$	Φ
$\{q_0,q_1,\ldots,q_k\}$	{…}	{…}

(1) 各時点で NFA A_Lの取り得る状態の集合 ($2^{|Q_N|}$通り)

(2) Σの要素(NFA A_L への可能な入力; $|\Sigma|$通り)

(1) の各状態において、(2)の入力を与えた場合に遷移できるすべての状態の集合

2. 有限オートマトン(2)

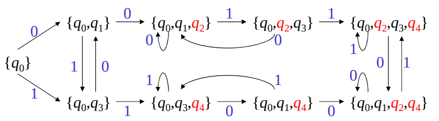


例:

δ	0	1
q_0	$\{q_0,q_1\}$	$\{q_0,q_3\}$
q_1	$\{q_2\}$	Φ
q_2	$\{q_2\}$	$\{q_2\}$
q_3	Φ	$\{q_4\}$
q_4	$\{q_4\}$	$\{q_4\}$

	現在の状態の集合 ($2^{|Q|}$通り)	入力 0	入力 1
(1)	Φ	Φ	Φ
(2)	$\{q_0\}$	$\{q_0,q_1\}$	$\{q_0,q_3\}$
(3)	$\{q_1\}$	$\{q_2\}$	Φ
(4)	$\{q_2\}$	$\{q_2\}$	$\{q_2\}$
(5)	$\{q_3\}$	Φ	[illegible]
(6)	$\{q_4\}$	$\{q_4\}$	[illegible]
(7)	$\{q_0,q_1\}$	$\{q_0,q_1,q_2\}$	$\{q_0,q_3\}$
(8)	$\{q_0,q_2\}$	$\{q_0,q_1,q_2\}$	$\{q_0,q_2,q_3\}$
…			
(32)	$\{q_0,q_1,q_2,q_3,q_4\}$	$\{q_0,q_1,q_2,q_4\}$	$\{q_0,q_2,q_3,q_4\}$

次の時刻に可能な状態の集合



2. 有限オートマトン(2)

証明: NFAで受理できる言語のクラスNと、DFAで受理できる言語のクラスDが一致することを示す。

$L \in N$ を受理する NFA $A_L=(Q_N, \Sigma, \delta_N, q_N, F_N)$ が存在する。

A_Lと同じ言語を受理する DFA A_L' を次のように構成する。

$$A_L' = \{2^{Q_N}, \Sigma, \delta_D, \{q_N\}, F_D\}$$

– 状態集合は A_L の状態集合の集合族
– 初期状態 $\{q_N\}$ は 'q_N だけからなる集合' であり、q_N ではない
– δ_D と F_D の定義方法は前述の通り。

証明すべきこと: $\hat{\delta}_N(q_N,w) \cap F_N \neq \Phi$である必要十分条件は $\hat{\delta}_D(\{q_N\},w) \in F_D$

⇒$|w|$に関する帰納法で、計算の同等性を証明する。(省略)

2. 有限オートマトン(2)

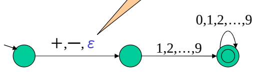
2.5. ε-動作を含む有限オートマトン(ε-NFA)

– 「入力」として「空文字ε」を許す。つまり入力を読まずに状態を変化することを許す。

例: 「0でない整数」

1. 最初は「＋」か「－」か何もない
2. 次は1～9が1つ
3. それ以降は0～9が0個以上続く

εを使わずに自然な表現をするのは困難



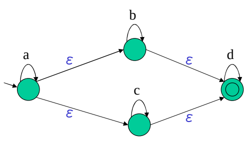
2. 有限オートマトン(2)

2.5. ε-動作を含む有限オートマトン(ε-NFA)

例:

1. まずaが0個以上続き、
2. 次に[bが0個以上]または[cが0個以上]続き、
3. 最後にdが0個以上続く

εを使わずに自然な表現をするのは困難



2. 有限オートマトン(2)

2.5. ε-動作を含む有限オートマトン(ε-NFA)

– ε-NFA $A=(Q, \Sigma, \delta, q, F)$の定義:
  - Q:状態集合
  - Σ:アルファベット
  - δ: $Q \times \Sigma \cup \{\varepsilon\} \rightarrow 2^Q$
  - q: 初期状態
  - F: 受理状態

– ε-NFA A によって受理される言語…
  - $\hat{\delta}$の定義??

2. 有限オートマトン(2)

2. 5. ε-NFAとDFAの等価性

証明: ε-NFAで受理できる言語のクラスNと、DFAで受理できる言語のクラスDが一致することを示す。

- $D \subseteq N$は定義より明らかなので、$N \subseteq D$を示せばよい。
- 任意の言語 $L \in N$ が $L \in D$ となることを示せばよい。

ある言語 L が $L \in N$ であったとする。このとき、Lを受理する ε-NFA $A_L=(Q_N, \Sigma, \delta_N, q_N, F_N)$ が存在する。
A_Lと同じ言語を受理する DFA A_L' を構成する。

Subset 構成において、ε-遷移をどう処理するか…

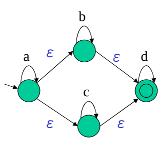
2. 有限オートマトン(2)

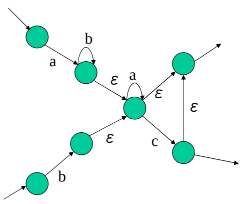
2. 5. ε-NFAとDFAの等価性

Subset 構成において、ε-遷移をどう処理するか…

直感的には「ε で移動できる状態たち」を同一視すればOK…?

→それほど自明でない:





2. 有限オートマトン(2)

2. 5. ε-NFAとDFAの等価性

Subset 構成において、ε-遷移をどう処理するか…

状態qの ε-閉包とは、状態qから ε-遷移だけで遷移できる状態の集合(q自身も含む)

ECLOSE(q) := { q' | q'はq から ε-遷移だけで遷移できる}

1. q は ECLOSE(q) の要素
2. 任意の $q' \in$ECLOSE(q)に対して、q'からq''に ε-遷移で遷移できるなら、q''もECLOSE(q)の要素

2. 有限オートマトン(2)

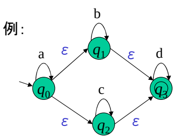
2. 5. ε-NFAとDFAの等価性

Subset 構成において、ε-遷移をどう処理するか…

状態qの ε-閉包とは、状態qから ε-遷移だけで遷移できる状態の集合(q自身も含む)

ECLOSE(q) := { q' | q'はq から ε-遷移だけで遷移できる}

例:



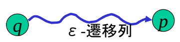
ECLOSE(q_0)={q_0,q_1,q_2,q_3}
ECLOSE(q_1)={q_1,q_3}
ECLOSE(q_2)={q_2,q_3}
ECLOSE(q_3)={q_3}

2. 有限オートマトン(2)

2. 5. ε-NFAとDFAの等価性

Subset 構成において、ε-遷移をどう処理するか…

観測: ε-NFA Aにおいて、ECLOSE(q)に状態 p が入っているとき、「Aがある時点で取りえる状態」の集合Sは、[$q \in S$ かつ $p \notin S$] はありえない。



⇒ ε-NFA Aにおいて、「Aがある時点で取りうる状態」の集合Sは、$q \in S$ なら ECLOSE(q)$\subseteq S$。

⇒Subset 構成において 2^Q がすべて現れるわけではない。

2. 有限オートマトン(2)

2.5.4. 遷移関数の拡張と ε -NFAの言語

– ε -NFA $A=(Q, \Sigma, \delta, q, F)$の定義：
- $\delta: Q\times \Sigma\cup\{\varepsilon\} \rightarrow 2^Q$

– ε -NFA A によって受理される言語…
- $\hat{\delta}: Q\times(\Sigma\cup\{\varepsilon\})^* \rightarrow 2^Q$の定義:
  1. $\hat{\delta}(q,\varepsilon) := \mathrm{ECLOSE}(q)$
  2. $\hat{\delta}(q,xa)$ (ただし $x\in\Sigma^*$, $a\in\Sigma$):
     – $\hat{\delta}(q,x) = \{p_1,p_2,\ldots,p_k\}$ とする。
     – 和集合 $\bigcup_{i=1}^{k}\hat{\delta}(p_i,a)$ を $\{r_1,r_2,\ldots,r_m\}$とする。

$$\hat{\delta}(q,xa) := \bigcup_{j=1}^{m}\mathrm{ECLOSE}(r_j)$$

状態qに入力xaが与えられたときに到達可能なすべての状態の集合

– Aによって受理される言語
$L(A) := \{ w \mid \hat{\delta}(q,w)\cap F\neq\Phi\}$

19/26

2. 有限オートマトン(2)

2. 5. ε -NFAとDFAの等価性

証明: ε -NFAで受理できる言語のクラスNと、DFAで受理できる言語のクラスDが一致することを示す。
- $D\subseteq N$は定義より明らかなので、$N\subseteq D$を示せばよい。
- 任意の言語 $L\in N$ が $L\in D$ となることを示せばよい。

ある言語 L が $L\in N$ であったとする。このとき、Lを受理する ε -NFA $A_L=(Q_N, \Sigma, \delta_N, q_N, F_N)$ が存在する。
A_Lと同じ言語を受理する DFA A_L' を構成する。

Subset 構成において、ε -遷移をどう処理するか…

ECLOSEを使って遷移可能な状態の集合を表現する

20/26

2. 有限オートマトン(2)

2. 5. ε -NFAとDFAの等価性

証明: ある言語 L が $L\in N$ であったとする。このとき、Lを受理する ε -NFA $A_L=(Q_N, \Sigma, \delta_N, q_N, F_N)$ が存在する。
A_Lと同じ言語を受理する DFA $A_L'=(Q_D, \Sigma, \delta_D, q_D, F_D)$を構成する。

1. Q_D : 2^{Q_N}だと無駄が多い。以下を満たすSだけで十分。

$$S = \bigcup_{q\in S}\mathrm{ECLOSE}(q)$$

与えられた ε -NFAから動的に作ればよい。

2. $q_D := \mathrm{ECLOSE}(q_N)$
3. $F_D := \{ S\in Q_D \mid S \cap F_N \neq \Phi\}$
4. δ_D …

21/26

2. 有限オートマトン(2)

2. 5. ε -NFAとDFAの等価性

証明: ある言語 L が $L\in N$ であったとする。このとき、Lを受理する ε -NFA $A_L=(Q_N, \Sigma, \delta_N, q_N, F_N)$ が存在する。
A_Lと同じ言語を受理する DFA $A_L'=(Q_D, \Sigma, \delta_D, q_D, F_D)$ を構成する。

4. δ_D : Q_Dの要素SとΣの要素 a に対して、以下の手順で構成する。
   1. $S = \{p_1,p_2,\ldots,p_k\}$ とする。
   2. $\bigcup_{i=1}^{k}\delta(p_i,a)$ の結果を $\{r_1,r_2,\ldots,r_m\}$ とする。
   3. $\delta_D(S,a) := \bigcup_{j=1}^{m}\mathrm{ECLOSE}(r_j)$

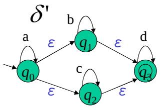
22/26

2. 有限オートマトン(2)

2. 5. ε -NFAとDFAの等価性

例:

δ'



ECLOSE(q_0)=$\{q_0,q_1,q_2,q_3\}$
ECLOSE(q_1)=$\{q_1,q_3\}$
ECLOSE(q_2)=$\{q_2,q_3\}$
ECLOSE(q_3)=$\{q_3\}$

上記の ε -NFAと等価な DFA $A=(Q,\{a,b,c,d\}, \delta, q, F)$ を構成
- q = ECLOSE(q_0)=$\{q_0,q_1,q_2,q_3\}$
- $\delta(q,b)$: $\bigcup_{q_i\in q}\delta'(q_i,b)=\{q_1\}$ なので、$\delta(q,b)$ = ECLOSE(q_1)=$\{q_1,q_3\}$
- 同様に

  $\delta(q,a)$ = ECLOSE(q_0) = $\{q_0,q_1,q_2,q_3\}$
  $\delta(q,c)$ = ECLOSE(q_2) = $\{q_2,q_3\}$
  $\delta(q,d)$ = ECLOSE(q_3) = $\{q_3\}$

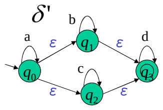
23/26

2. 有限オートマトン(2)

2. 5. ε -NFAとDFAの等価性

例:

δ'



ECLOSE(q_0)=$\{q_0,q_1,q_2,q_3\}$
ECLOSE(q_1)=$\{q_1,q_3\}$
ECLOSE(q_2)=$\{q_2,q_3\}$
ECLOSE(q_3)=$\{q_3\}$

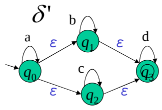
上記の ε -NFAと等価な DFA $A=(Q,\{a,b,c,d\}, \delta, q, F)$ を構成
同様に
$\delta(\{q_1,q_3\},a)$ = ECLOSE(Φ) = $\{\Phi\}$
$\delta(\{q_1,q_3\},b)$ = ECLOSE(q_1) = $\{q_1,q_3\}$
$\delta(\{q_1,q_3\},c)$ = ECLOSE(Φ) = $\{\Phi\}$
$\delta(\{q_1,q_3\},d)$ = ECLOSE(q_3) = $\{q_3\}$…

24/26

2. 有限オートマトン(2)

2. 5. ε -NFAとDFAの等価性

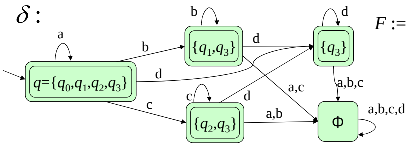
例:



ECLOSE(q_0)={q_0,q_1,q_2,q_3}
ECLOSE(q_1)={q_1,q_3}
ECLOSE(q_2)={q_2,q_3}
ECLOSE(q_3)={q_3}

上記の ε -NFAと等価な DFA A=(Q,{a,b,c,d}, δ,q,F) を構成

δ :



F := {{q_0,q_1,q_2,q_3},
{q_1,q_3},
{q_2,q_3},{q_3}}

2. 有限オートマトン(2)

[大雑把なまとめとコメント]

- DFA, NFA, ε -NFA
  - DFA: 決定的
  - NFA: 非決定的
  - ε -NFA: 入力がなくても状態が変化しうる
- 「受理できる言語」という観点では同等
  - NFA, ε -NFAをDFAにsubset constructionで変換すると、最悪の場合は状態数は指数関数的に増える (n→ 2^n)(実際にそうなる例もある)
  - 実用上は指数関数的に増えない場合が多い
  - システムによってはNFA, ε -NFAのまま管理するものもある