Teil IV

Reguläre Sprachen und endliche Automaten
Teil 4: Minimierung von DFAs

(Nicht-)Minimale Automaten

Minimaler DFA

Ein vollständiger DFA A heißt **minimal**, wenn kein anderer vollständiger DFA für $\mathcal{L}(A)$ weniger Zustände hat als A.

(Nicht-)Minimale Automaten

Minimaler DFA

Ein vollständiger DFA A heißt **minimal**, wenn kein anderer vollständiger DFA für $\mathcal{L}(A)$ weniger Zustände hat als A.

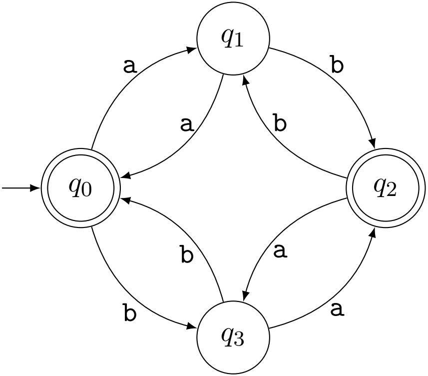
Beobachtung

Nicht jeder DFA ist minimal.

(Nicht-)Minimale Automaten

Minimaler DFA

Ein vollständiger DFA A heißt **minimal**, wenn kein anderer vollständiger DFA für $\mathcal{L}(A)$ weniger Zustände hat als A.



Beobachtung

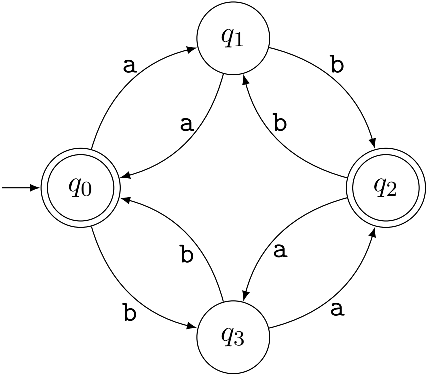
Nicht jeder DFA ist minimal.

Minimaler DFA

Ein vollständiger DFA A heißt **minimal**, wenn kein anderer vollständiger DFA für $\mathcal{L}(A)$ weniger Zustände hat als A.

Beobachtung

Nicht jeder DFA ist minimal.



Frage

Wie können wir DFAs minimieren?

Lösung: Der Äquivalenzklassenautomat

Satz

Sei Σ Alphabet, sei $L \subseteq \Sigma^*$ regulär.
Sei $A_{\equiv_L}$ der Äquivalenzklassenautomat zu L. Es gilt:

1. $A_{\equiv_L}$ ist minimal.
2. Sei $A_M = (\Sigma, Q_M, \delta_M, q_{0,M}, F_M)$ ein minimaler DFA mit $\mathcal{L}(A_M) = L$. Dann gilt für alle $x, y \in \Sigma^*$:

   $\delta_M(q_{0,M}, x) = \delta_M(q_{0,M}, y)$ genau dann, wenn $x \equiv_L y$.

Lösung: Der Äquivalenzklassenautomat

Satz

Sei Σ Alphabet, sei $L \subseteq \Sigma^*$ regulär.
Sei $A_{\equiv_L}$ der Äquivalenzklassenautomat zu L. Es gilt:

1. $A_{\equiv_L}$ ist minimal.
2. Sei $A_M = (\Sigma, Q_M, \delta_M, q_{0,M}, F_M)$ ein minimaler DFA mit $\mathcal{L}(A_M) = L$. Dann gilt für alle $x, y \in \Sigma^*$:

   $\delta_M(q_{0,M}, x) = \delta_M(q_{0,M}, y)$ genau dann, wenn $x \equiv_L y$.

Also:

Der Äquivalenzklassenautomat ist der einzige minimale Automat für L (bis auf Umbenennungen).

Lösung: Der Äquivalenzklassenautomat

Satz

Sei Σ Alphabet, sei $L \subseteq \Sigma^*$ regulär.
Sei $A_{\equiv_L}$ der Äquivalenzklassenautomat zu L. Es gilt:

1. $A_{\equiv_L}$ ist minimal.
2. Sei $A_M = (\Sigma, Q_M, \delta_M, q_{0,M}, F_M)$ ein minimaler DFA mit $\mathcal{L}(A_M) = L$. Dann gilt für alle $x, y \in \Sigma^*$:

   $\delta_M(q_{0,M}, x) = \delta_M(q_{0,M}, y)$ genau dann, wenn $x \equiv_L y$.

Also:

Der Äquivalenzklassenautomat ist der einzige minimale Automat für L (bis auf Umbenennungen).

Ansatz

Zu gegebenem DFA A den Äquivalenzklassenautomat für $\mathcal{L}(A)$ berechnen.

Lösung: Der Äquivalenzklassenautomat

Satz

Sei Σ Alphabet, sei $L \subseteq \Sigma^*$ regulär.
Sei $A_{\equiv_L}$ der Äquivalenzklassenautomat zu L. Es gilt:

1. $A_{\equiv_L}$ ist minimal.
2. Sei $A_M = (\Sigma, Q_M, \delta_M, q_{0,M}, F_M)$ ein minimaler DFA mit $\mathcal{L}(A_M) = L$. Dann gilt für alle $x, y \in \Sigma^*$:

   $\delta_M(q_{0,M}, x) = \delta_M(q_{0,M}, y)$ genau dann, wenn $x \equiv_L y$.

Also:

Der Äquivalenzklassenautomat ist der einzige minimale Automat für L (bis auf Umbenennungen).

Ansatz

Zu gegebenem DFA A den Äquivalenzklassenautomat für $\mathcal{L}(A)$ berechnen.

- Idee: Verschmelzen von Zuständen, die nicht unterscheidbar sind.

Handwerkszeug

- Sei $A := (\Sigma, Q, \delta, q_0, F)$ vollständiger DFA.
- A_q : Selber DFA, aber mit Startzustand q.

Handwerkszeug

- Sei $A := (\Sigma, Q, \delta, q_0, F)$ vollständiger DFA.
- A_q : Selber DFA, aber mit Startzustand q.

(Un-)Unterscheidbarkeitsrelation

$p, q \in Q$ sind **nicht unterscheidbar** wenn $\mathcal{L}(A_p) = \mathcal{L}(A_q)$.
Schreiben: $p \equiv_A q$

Handwerkszeug

- Sei $A := (\Sigma, Q, \delta, q_0, F)$ vollständiger DFA.
- A_q : Selber DFA, aber mit Startzustand q.
- $\equiv_A$ ist Äquivalenzrelation

(Un-)Unterscheidbarkeitsrelation

$p, q \in Q$ sind **nicht unterscheidbar** wenn $\mathcal{L}(A_p) = \mathcal{L}(A_q)$.
Schreiben: $p \equiv_A q$

Handwerkszeug

- Sei $A := (\Sigma, Q, \delta, q_0, F)$ vollständiger DFA.
- A_q : Selber DFA, aber mit Startzustand q.
- $\equiv_A$ ist Äquivalenzrelation

(Un-)Unterscheidbarkeitsrelation

$p, q \in Q$ sind **nicht unterscheidbar** wenn $\mathcal{L}(A_p) = \mathcal{L}(A_q)$.
Schreiben: $p \equiv_A q$

Lemma

Sei $A := (\Sigma, Q, \delta, q_0, F)$ vollständiger DFA, $L := \mathcal{L}(A)$.
Für alle $x, y \in \Sigma^*$ gilt:

$$x \equiv_L y \text{ genau dann, wenn } \delta(q_0, x) \equiv_A \delta(q_0, y)$$

Handwerkszeug

- Sei $A := (\Sigma, Q, \delta, q_0, F)$ vollständiger DFA.
- A_q : Selber DFA, aber mit Startzustand q.
- $\equiv_A$ ist Äquivalenzrelation

(Un-)Unterscheidbarkeitsrelation

$p, q \in Q$ sind **nicht unterscheidbar** wenn $\mathcal{L}(A_p) = \mathcal{L}(A_q)$.
Schreiben: $p \equiv_A q$

Lemma

Sei $A := (\Sigma, Q, \delta, q_0, F)$ vollständiger DFA, $L := \mathcal{L}(A)$.
Für alle $x, y \in \Sigma^*$ gilt:

$$x \equiv_L y \text{ genau dann, wenn } \delta(q_0, x) \equiv_A \delta(q_0, y)$$

Konsequenz

Wenn A reduziert ist und alle Zustandspaare unterscheidbar sind, ist A der Äquivalenzklassenautomat.

Die Idee hinter dem Algorithmus

Hauptidee

berechnen nicht $\equiv_A = \big\{\{p, q\} \mid p, q \in Q, p \equiv_A q\big\}$ (nicht unterscheidbar)

sondern $U = \big\{\{p, q\} \mid p, q \in Q, p \not\equiv_A q\big\}$ (unterscheidbar)

suchen dazu $z \in \Sigma^*$ mit $z \in \big(\mathcal{L}(A_p) \triangle \mathcal{L}(A_q)\big)$

Die Idee hinter dem Algorithmus

Hauptidee

berechnen nicht $\equiv_A = \big\{\{p,q\} \mid p,q \in Q, p \equiv_A q\big\}$ (nicht unterscheidbar)

sondern $U = \big\{\{p,q\} \mid p,q \in Q, p \not\equiv_A q\big\}$ (unterscheidbar)

suchen dazu $z \in \Sigma^*$ mit $z \in \big(\mathcal{L}(A_p) \triangle \mathcal{L}(A_q)\big)$

Konstruieren dazu:

$$U_0 := \big\{\{p,q\} \mid p \in F, q \in (Q-F)\big\}$$

$$U_{i+1} := U_i \cup \big\{\{p,q\} \mid \{\delta(p,a), \delta(q,a)\} \in U_i, a \in \Sigma\big\}$$

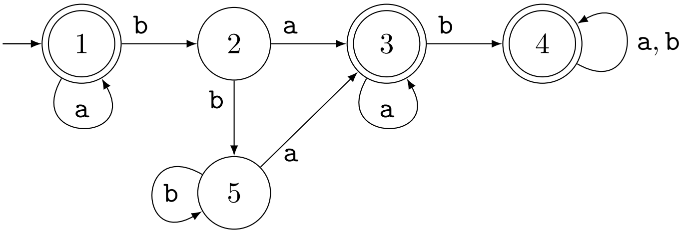
- U_i enthält die Paare, die durch (mindestens) ein Wort der Länge $\leq i$ unterschieden werden können

Algorithmus 1 : `minimiereDFA`

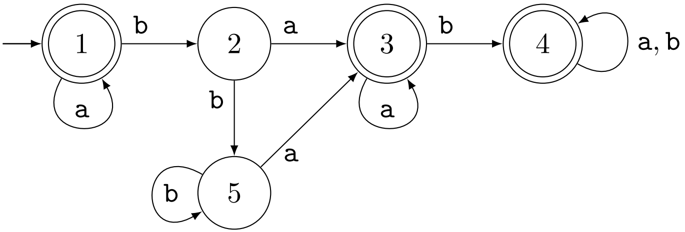
Eingabe : Ein DFA $A := (\Sigma, Q, \delta, q_0, F)$

entferne alle Zustände aus Q, die nicht erreichbar sind;
$U_0 \leftarrow \big\{\{p,q\} \mid p \in F, q \in (Q - F)\big\}$;
fertig $\leftarrow$ false; $i \leftarrow 0$;
while not fertig **do**
  fertig $\leftarrow$ true; $U_{i+1} \leftarrow U_i$;
  foreach $p, q \in Q$ **mit** $p \neq q$, $\{p,q\} \notin U_i$ **do**
    foreach $a \in \Sigma$ **do**
      if $\big\{\delta(p,a), \delta(q,a)\big\} \in U_i$ **then**
        füge $\{p,q\}$ zu U_{i+1} hinzu; fertig $\leftarrow$ false;
  $i \leftarrow i + 1$
$U \leftarrow U_i$;
$[q]_{\equiv_A} := \{p \mid \{p,q\} \notin U\}$ für alle $q \in Q$;
return Äquivalenzklassenautomat für $\equiv_A$

Beispiel

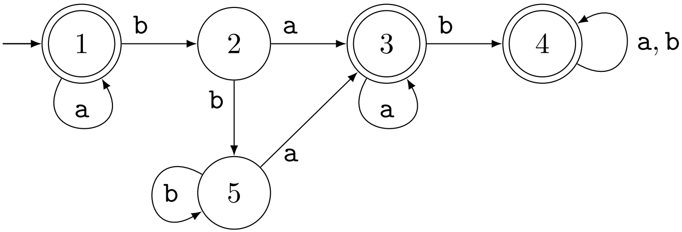


Beispiel



	a	b
1	1	2
2	3	5
3	3	4
4	4	4
5	3	5

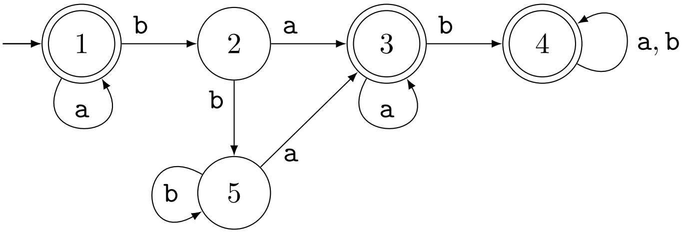
Beispiel



	a	b
1	1	2
2	3	5
3	3	4
4	4	4
5	3	5

2				
3				
4				
5				
	1	2	3	4

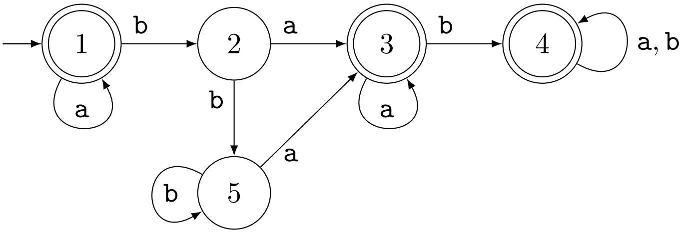
Beispiel



	a	b
1	1	2
2	3	5
3	3	4
4	4	4
5	3	5

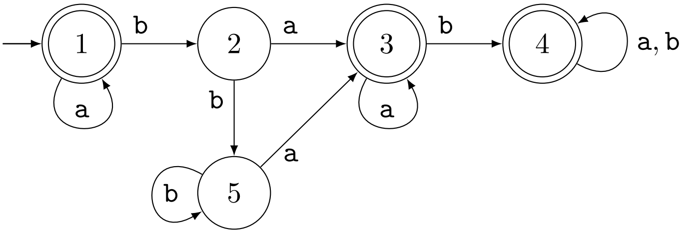
2	U_0			
3		U_0		
4		U_0		
5	U_0		U_0	U_0
	1	2	3	4

Beispiel



	a	b
1	1	2
2	3	5
3	3	4
4	4	4
5	3	5

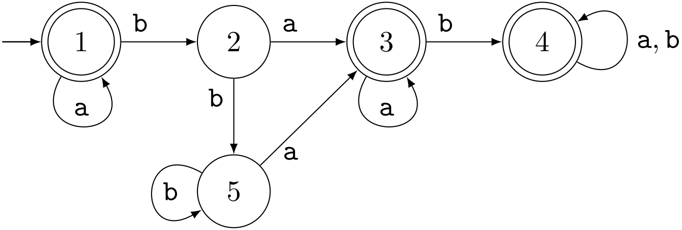
2	U_0			
3	U_1	U_0		
4		U_0		
5	U_0		U_0	U_0
	1	2	3	4



	a	b
1	1	2
2	3	5
3	3	4
4	4	4
5	3	5

2	U_0			
3	U_1	U_0		
4	U_1	U_0		
5	U_0		U_0	U_0
	1	2	3	4

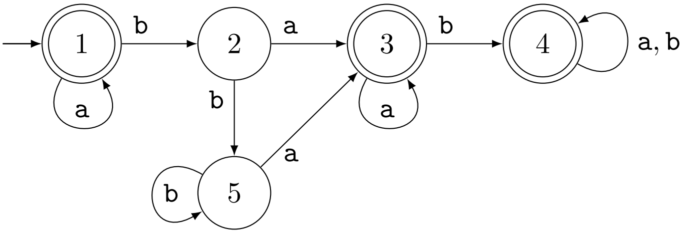
Beispiel



	a	b
1	1	2
2	3	5
3	3	4
4	4	4
5	3	5

2	U_0			
3	U_1	U_0		
4	U_1	U_0	$\equiv$	
5	U_0	$\equiv$	U_0	U_0
	1	2	3	4

Beispiel



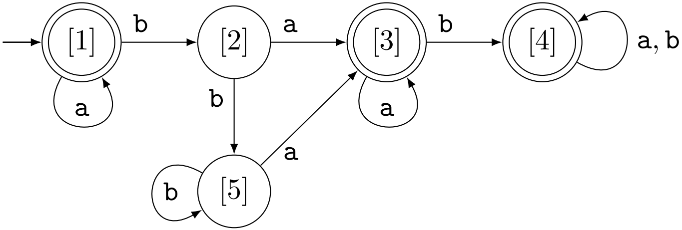
	a	b
1	1	2
2	3	5
3	3	4
4	4	4
5	3	5

Äquivalenzklassen von $\equiv_A$

$$[1] = \{1\},$$
$$[2] = [5] = \{2, 5\},$$
$$[3] = [4] = \{3, 4\}.$$

2	U_0			
3	U_1	U_0		
4	U_1	U_0	$\equiv$	
5	U_0	$\equiv$	U_0	U_0
	1	2	3	4

Beispiel



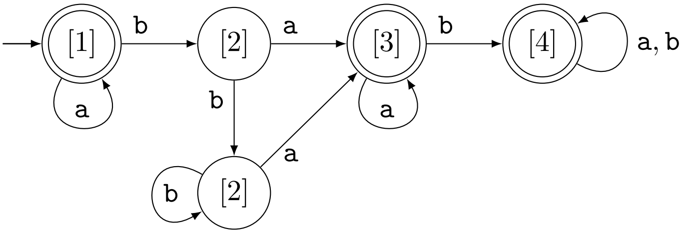
	a	b
[1]	[1]	[2]
[2]	[3]	[5]
[3]	[3]	[4]
[4]	[4]	[4]
[5]	[3]	[5]

Äquivalenzklassen von $\equiv_A$

$$\begin{aligned}[1] &= \{1\},\\ [2] = [5] &= \{2,5\},\\ [3] = [4] &= \{3,4\}.\end{aligned}$$

2	U_0			
3	U_1	U_0		
4	U_1	U_0	$\equiv$	
5	U_0	$\equiv$	U_0	U_0
	1	2	3	4

Beispiel



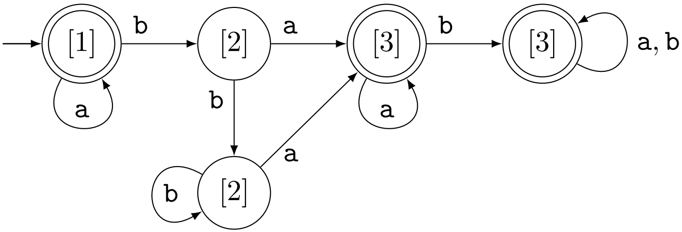
	a	b
[1]	[1]	[2]
[2]	[3]	[2]
[3]	[3]	[4]
[4]	[4]	[4]
[2]	[3]	[2]

Äquivalenzklassen von $\equiv_A$

$$[1] = \{1\},$$
$$[2] = [5] = \{2, 5\},$$
$$[3] = [4] = \{3, 4\}.$$

2	U_0			
3	U_1	U_0		
4	U_1	U_0	$\equiv$	
5	U_0	$\equiv$	U_0	U_0
	1	2	3	4

Beispiel



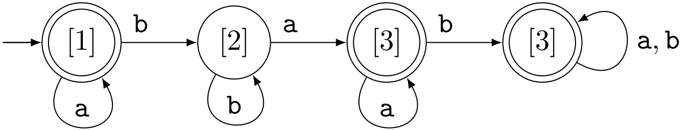
	a	b
[1]	[1]	[2]
[2]	[3]	[2]
[3]	[3]	[3]
[3]	[3]	[3]
[2]	[3]	[2]

Äquivalenzklassen von $\equiv_A$

$$[1] = \{1\},$$
$$[2] = [5] = \{2, 5\},$$
$$[3] = [4] = \{3, 4\}.$$

2	U_0			
3	U_1	U_0		
4	U_1	U_0	$\equiv$	
5	U_0	$\equiv$	U_0	U_0
	1	2	3	4

Beispiel

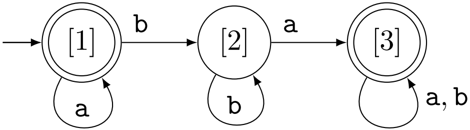


	a	b
[1]	[1]	[2]
[2]	[3]	[2]
[3]	[3]	[3]
[3]	[3]	[3]

Äquivalenzklassen von $\equiv_A$

$$[1] = \{1\},$$
$$[2] = [5] = \{2, 5\},$$
$$[3] = [4] = \{3, 4\}.$$

2	U_0			
3	U_1	U_0		
4	U_1	U_0	$\equiv$	
5	U_0	$\equiv$	U_0	U_0
	1	2	3	4



	a	b
[1]	[1]	[2]
[2]	[3]	[2]
[3]	[3]	[3]

Äquivalenzklassen von $\equiv_A$

$$[1] = \{1\},$$
$$[2] = [5] = \{2, 5\},$$
$$[3] = [4] = \{3, 4\}.$$

2	U_0			
3	U_1	U_0		
4	U_1	U_0	$\equiv$	
5	U_0	$\equiv$	U_0	U_0
	1	2	3	4

Analyse von `minimiereDFA`

Satz

Sei $A := (\Sigma, Q, \delta, q_0, F)$ ein vollständiger DFA.
Sei $k := |\Sigma|$ und $n := |Q|$.
`minimiereDFA` berechnet aus A einen minimalen DFA für $\mathcal{L}(A)$ in maximal $O(kn^3)$ Schritten.

Analyse von minimiereDFA

Satz

Sei $A := (\Sigma, Q, \delta, q_0, F)$ ein vollständiger DFA.
Sei $k := |\Sigma|$ und $n := |Q|$.
minimiereDFA berechnet aus A einen minimalen DFA für $\mathcal{L}(A)$ in maximal $O(kn^3)$ Schritten.

Lemma

$\{p, q\} \in U_i$ genau dann, wenn ein $z \in \Sigma^*$ existiert, für das $|z| \leq i$ und $z \in \big(\mathcal{L}(A_p) \triangle \mathcal{L}(A_q)\big)$

Analyse von `minimiereDFA`

Satz

Sei $A := (\Sigma, Q, \delta, q_0, F)$ ein vollständiger DFA.
Sei $k := |\Sigma|$ und $n := |Q|$.
`minimiereDFA` berechnet aus A einen minimalen DFA für $\mathcal{L}(A)$ in maximal $O(kn^3)$ Schritten.

Lemma

$\{p, q\} \in U_i$ genau dann, wenn ein $z \in \Sigma^*$ existiert, für das $|z| \leq i$ und $z \in \big(\mathcal{L}(A_p) \triangle \mathcal{L}(A_q)\big)$

Lemma

Beim Aufruf von `minimiereDFA` auf A wird die while-Schleife in Zeile 4 maximal $(n-1)$-mal durchlaufen.

Algorithmus 1 : `minimiereDFA`

Eingabe : Ein DFA $A := (\Sigma, Q, \delta, q_0, F)$

entferne alle Zustände aus Q, die nicht erreichbar sind;
$U_0 \leftarrow \big\{\{p,q\} \mid p \in F, q \in (Q - F)\big\}$;
fertig $\leftarrow$ false; $i \leftarrow 0$;
while **not** fertig **do**
    fertig $\leftarrow$ true; $U_{i+1} \leftarrow U_i$;
    foreach $p, q \in Q$ **mit** $p \neq q$, $\{p,q\} \notin U_i$ **do**
        foreach $a \in \Sigma$ **do**
            if $\big\{\delta(p,a), \delta(q,a)\big\} \in U_i$ **then**
                füge $\{p,q\}$ zu U_{i+1} hinzu; fertig $\leftarrow$ false;
    $i \leftarrow i + 1$
$U \leftarrow U_i$;
$[q]_{\equiv_A} := \{p \mid \{p,q\} \notin U\}$ für alle $q \in Q$;
return **Äquivalenzklassenautomat für** $\equiv_A$