Script **generated by TTT**

Title: Seidl: Theoretische_Informatik (07.05.2012)

Date: Mon May 07 10:18:21 CEST 2012

Duration: 87:55 min

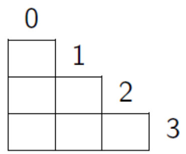
Pages: 108

Implementierung von U:

Tabelle von anfangs unmarkierten Paaren $\{p, q\}$, $p \neq q$.

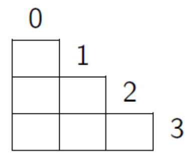
Implementierung von U:

Tabelle von anfangs unmarkierten Paaren $\{p, q\}$, $p \neq q$.



Implementierung von U:

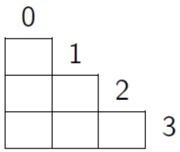
Tabelle von anfangs unmarkierten Paaren $\{p, q\}$, $p \neq q$.



for all $p \in F$, $q \in Q \setminus F$ **do** markiere $\{p, q\}$

Implementierung von U:

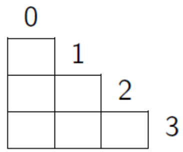
Tabelle von anfangs unmarkierten Paaren $\{p, q\}$, $p \neq q$.



for all $p \in F$, $q \in Q \setminus F$ **do** markiere $\{p, q\}$
while $\exists$ unmarkiertes $\{p, q\}$ $\exists a \in \Sigma$. $\{\delta(p, a), \delta(q, a)\}$ ist markiert

Implementierung von U:

Tabelle von anfangs unmarkierten Paaren $\{p, q\}$, $p \neq q$.



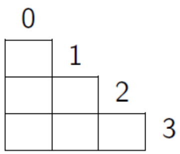
for all $p \in F$, $q \in Q \setminus F$ **do** markiere $\{p, q\}$
while $\exists$ unmarkiertes $\{p, q\}$ $\exists a \in \Sigma$. $\{\delta(p, a), \delta(q, a)\}$ ist markiert
do markiere $\{p, q\}$

Komplexität:

$$O\left(\binom{n}{2} + \binom{n}{2}\binom{n}{2}|\Sigma|\right)$$

Implementierung von U:

Tabelle von anfangs unmarkierten Paaren $\{p, q\}$, $p \neq q$.



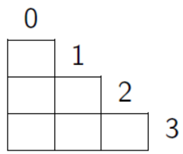
for all $p \in F$, $q \in Q \setminus F$ **do** markiere $\{p, q\}$
while $\exists$ unmarkiertes $\{p, q\}$ $\exists a \in \Sigma$. $\{\delta(p, a), \delta(q, a)\}$ ist markiert
do markiere $\{p, q\}$

Komplexität:

$$O\left(\binom{n}{2} + \binom{n}{2}\binom{n}{2}|\Sigma|\right) = O\left(\frac{n(n-1)}{2} + \frac{n^2(n-1)^2}{4}|\Sigma|\right)$$

Implementierung von U:

Tabelle von anfangs unmarkierten Paaren $\{p, q\}$, $p \neq q$.



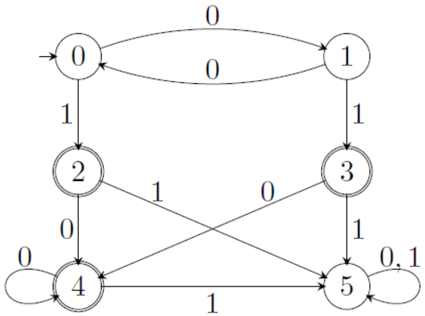
for all $p \in F$, $q \in Q \setminus F$ **do** markiere $\{p, q\}$
while $\exists$ unmarkiertes $\{p, q\}$ $\exists a \in \Sigma$. $\{\delta(p, a), \delta(q, a)\}$ ist markiert
do markiere $\{p, q\}$

Komplexität:

$$O\left(\binom{n}{2} + \binom{n}{2}\binom{n}{2}|\Sigma|\right) = O\left(\frac{n(n-1)}{2} + \frac{n^2(n-1)^2}{4}|\Sigma|\right) = O(n^4)$$

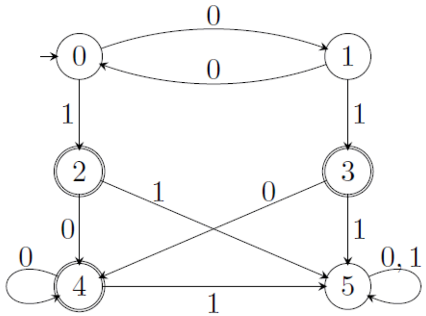
Bei fixem Σ.

Beispiel 2.58



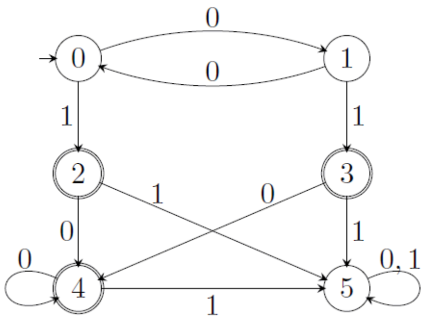
	0	1	2	3	4
1					
2					
3					
4					
5					

Beispiel 2.58



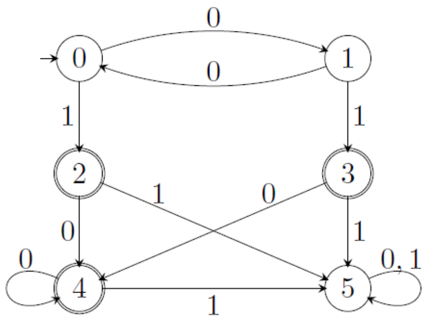
	0	1	2	3	4
1					
2	×	×			
3	×	×			
4	×	×			
5			×	×	×

Beispiel 2.58

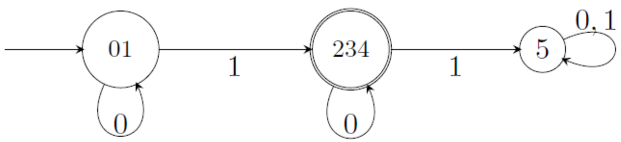


	0	1	2	3	4
1					
2	×	×			
3	×	×			
4	×	×			
5	×	×	×	×	×

Beispiel 2.58



	0	1	2	3	4
1					
2	×	×			
3	×	×			
4	×	×			
5	×	×	×	×	×



Von $O(n^4)$ zu $O(n^2)$ mit Abhängigkeitsanalyse:

Von $O(n^4)$ zu $O(n^2)$ mit Abhängigkeitsanalyse:

$\{p', q'\}$ unterscheidbar $\Longrightarrow$
$\{p, q\}$ unterscheidbar falls $\{p, q\} \xrightarrow{a} \{p', q'\}$

Von $O(n^4)$ zu $O(n^2)$ mit Abhängigkeitsanalyse:

$\{p', q'\}$ unterscheidbar $\Longrightarrow$
$\{p, q\}$ unterscheidbar falls $\{p, q\} \xrightarrow{a} \{p', q'\}$

$D[\{p', q'\}]$: Menge der Paare $\{p, q\}$ wie oben, anfangs leer

Von $O(n^4)$ zu $O(n^2)$ mit Abhängigkeitsanalyse:

$\{p', q'\}$ unterscheidbar $\Longrightarrow$
$\{p, q\}$ unterscheidbar falls $\{p, q\} \xrightarrow{a} \{p', q'\}$

$D[\{p', q'\}]$: Menge der Paare $\{p, q\}$ wie oben, anfangs leer

for all $\{p, q\} \subseteq Q$ mit $p \neq q$,

Von $O(n^4)$ zu $O(n^2)$ mit Abhängigkeitsanalyse:

$\{p', q'\}$ unterscheidbar $\implies$
$\{p, q\}$ unterscheidbar falls $\{p, q\} \xrightarrow{a} \{p', q'\}$

$D[\{p', q'\}]$: Menge der Paare $\{p, q\}$ wie oben, anfangs leer

for all $\{p, q\} \subseteq Q$ mit $p \neq q$, $a \in \Sigma$ **do**
  $p' := \delta(p, a);\ q' := \delta(q, a)$

Von $O(n^4)$ zu $O(n^2)$ mit Abhängigkeitsanalyse:

$\{p', q'\}$ unterscheidbar $\implies$
$\{p, q\}$ unterscheidbar falls $\{p, q\} \xrightarrow{a} \{p', q'\}$

$D[\{p', q'\}]$: Menge der Paare $\{p, q\}$ wie oben, anfangs leer

for all $\{p, q\} \subseteq Q$ mit $p \neq q$, $a \in \Sigma$ **do**
  $p' := \delta(p, a);\ q' := \delta(q, a)$
  if $p' \neq q'$ **then** $D[\{p', q'\}] := D[\{p', q'\}] \cup \{\{p, q\}\}$

Von $O(n^4)$ zu $O(n^2)$ mit Abhängigkeitsanalyse:

$\{p', q'\}$ unterscheidbar $\implies$
$\{p, q\}$ unterscheidbar falls $\{p, q\} \xrightarrow{a} \{p', q'\}$

$D[\{p', q'\}]$: Menge der Paare $\{p, q\}$ wie oben, anfangs leer

for all $\{p, q\} \subseteq Q$ mit $p \neq q$, $a \in \Sigma$ **do**
  $p' := \delta(p, a);\ q' := \delta(q, a)$
  if $p' \neq q'$ **then** $D[\{p', q'\}] := D[\{p', q'\}] \cup \{\{p, q\}\}$
for all $p' \in F$, $q' \in Q \setminus F$ **do** $mark(\{p', q'\})$

Von $O(n^4)$ zu $O(n^2)$ mit Abhängigkeitsanalyse:

$\{p', q'\}$ unterscheidbar $\implies$
$\{p, q\}$ unterscheidbar falls $\{p, q\} \xrightarrow{a} \{p', q'\}$

$D[\{p', q'\}]$: Menge der Paare $\{p, q\}$ wie oben, anfangs leer

for all $\{p, q\} \subseteq Q$ mit $p \neq q$, $a \in \Sigma$ **do**
  $p' := \delta(p, a);\ q' := \delta(q, a)$
  if $p' \neq q'$ **then** $D[\{p', q'\}] := D[\{p', q'\}] \cup \{\{p, q\}\}$
for all $p' \in F$, $q' \in Q \setminus F$ **do** $mark(\{p', q'\})$

$mark(\{p', q'\}) =$
  if $\{p', q'\}$ unmarkiert **then**
    markiere $\{p', q'\}$
    for all $pq \in D[\{p', q'\}]$ **do** $mark(pq)$

Von $O(n^4)$ zu $O(n^2)$ mit Abhängigkeitsanalyse:

$\{p', q'\}$ unterscheidbar $\implies$
$\{p, q\}$ unterscheidbar falls $\{p, q\} \xrightarrow{a} \{p', q'\}$

$D[\{p', q'\}]$: Menge der Paare $\{p, q\}$ wie oben, anfangs leer

for all $\{p, q\} \subseteq Q$ mit $p \neq q$, $a \in \Sigma$ **do**
  $p' := \delta(p, a);\ q' := \delta(q, a)$
  if $p' \neq q'$ **then** $D[\{p', q'\}] := D[\{p', q'\}] \cup \{\{p, q\}\}$
for all $p' \in F$, $q' \in Q \setminus F$ **do** $mark(\{p', q'\})$

$mark(\{p', q'\}) =$
  if $\{p', q'\}$ unmarkiert **then**
    markiere $\{p', q'\}$
    for all $pq \in D[\{p', q'\}]$ **do** $mark(pq)$

Komplexität: $O(n^2 + n^2) = O(n^2)$.

Von $O(n^4)$ zu $O(n^2)$ mit Abhängigkeitsanalyse:

$\{p', q'\}$ unterscheidbar $\implies$
$\{p, q\}$ unterscheidbar falls $\{p, q\} \xrightarrow{a} \{p', q'\}$

$D[\{p', q'\}]$: Menge der Paare $\{p, q\}$ wie oben, anfangs leer

for all $\{p, q\} \subseteq Q$ mit $p \neq q$, $a \in \Sigma$ **do**
  $p' := \delta(p, a);\ q' := \delta(q, a)$
  if $p' \neq q'$ **then** $D[\{p', q'\}] := D[\{p', q'\}] \cup \{\{p, q\}\}$
for all $p' \in F$, $q' \in Q \setminus F$ **do** $mark(\{p', q'\})$

$mark(\{p', q'\}) =$
  if $\{p', q'\}$ unmarkiert **then**
    markiere $\{p', q'\}$
    for all $pq \in D[\{p', q'\}]$ **do** $mark(pq)$

Komplexität: $O(n^2 + n^2) = O(n^2)$.
Korrektheit?

John Hopcroft.
An $n \log n$ Algorithm for Minimizing the States in a Finite Automaton. 1971.

Noch eine Anwendung: Äquivalenztest von DFAs.

Noch eine Anwendung: Äquivalenztest von DFAs.

1. Gegeben DFAs A und B, bilde disjunkte Vereiningung.
   („Male A und B nebeneinander.“)

Noch eine Anwendung: Äquivalenztest von DFAs.

1. Gegeben DFAs A und B, bilde disjunkte Vereiningung.
   („Male A und B nebeneinander.“)
2. Berechne Menge der äquivalenten Zustände.
3. $L(A) = L(B)$ gdw die beiden Startzustände äquivalent sind.

Bisher: Der Minimierungsalgorithmus (zur Erinnerung).

1. Entferne alle von q_0 aus nicht erreichbaren Zustände.
2. Berechne die *äquivalenten* Zustände des Automaten.
3. Kollabiere den Automaten durch Zusammenfassung aller äquivalenten Zustände.

Bisher: Der Minimierungsalgorithmus (zur Erinnerung).

1. Entferne alle von q_0 aus nicht erreichbaren Zustände.
2. Berechne die *äquivalenten* Zustände des Automaten.
3. Kollabiere den Automaten durch Zusammenfassung aller äquivalenten Zustände.

Jetzt: Die Präzisierung.

1. Was ist der „kollabierte Automat“?
2. Ist das wirklich der minimale Automat?

Bisher: Der Minimierungsalgorithmus (zur Erinnerung).

1. Entferne alle von q_0 aus nicht erreichbaren Zustände.
2. Berechne die *äquivalenten* Zustände des Automaten.
3. Kollabiere den Automaten durch Zusammenfassung aller äquivalenten Zustände.

Jetzt: Die Präzisierung.

1. Was ist der „kollabierte Automat"?
2. Ist das wirklich der minimale Automat?

Eine Relation $\approx \subseteq A \times A$ ist eine Äquivalenzrelation falls

- Reflexivität: $\forall a \in A.\ a \approx a$
- Symmetrie: $\forall a, b \in A.\ a \approx b \implies b \approx a$
- Transitivität: $\forall a, b, c \in A.\ a \approx b \wedge b \approx c \implies a \approx c$

Eine Relation $\approx \subseteq A \times A$ ist eine Äquivalenzrelation falls

- Reflexivität: $\forall a \in A.\ a \approx a$
- Symmetrie: $\forall a, b \in A.\ a \approx b \implies b \approx a$
- Transitivität: $\forall a, b, c \in A.\ a \approx b \wedge b \approx c \implies a \approx c$

Äquivalenzklasse:

$$[a]_\approx \ := \ \{b \mid a \approx b\}$$

Eine Relation $\approx \subseteq A \times A$ ist eine Äquivalenzrelation falls

- Reflexivität: $\forall a \in A.\ a \approx a$
- Symmetrie: $\forall a, b \in A.\ a \approx b \implies b \approx a$
- Transitivität: $\forall a, b, c \in A.\ a \approx b \wedge b \approx c \implies a \approx c$

Äquivalenzklasse:

$$[a]_\approx \ := \ \{b \mid a \approx b\}$$

Es gilt:

$$[a]_\approx = [b]_\approx \quad \Leftrightarrow \quad a \approx b$$

Eine Relation $\approx\, \subseteq A \times A$ ist eine Äquivalenzrelation falls

- Reflexivität: $\forall a \in A.\ a \approx a$
- Symmetrie: $\forall a, b \in A.\ a \approx b \implies b \approx a$
- Transitivität: $\forall a, b, c \in A.\ a \approx b \wedge b \approx c \implies a \approx c$

Äquivalenzklasse:

$$[a]_\approx \ := \ \{b \mid a \approx b\}$$

Es gilt:

$$[a]_\approx = [b]_\approx \quad \Leftrightarrow \quad a \approx b$$

Quotientenmenge:

$$A/\!\approx \ := \ \{[a]_\approx \mid a \in A\}$$

Partition von A

Im Folgenden sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DFA ohne unerreichbare Zustände.

Im Folgenden sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DFA ohne unerreichbare Zustände.

Definition 2.59 (Äquivalenz von Zuständen)

$$p \equiv_A q \quad \Leftrightarrow \quad (\forall w \in \Sigma^*.\ \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F)$$

$\approx$

Im Folgenden sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DFA ohne unerreichbare Zustände.

Definition 2.59 (Äquivalenz von Zuständen)

$$p \equiv_A q \quad \Leftrightarrow \quad (\forall w \in \Sigma^*.\ \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F)$$

p

Fakt 2.60

$\equiv_A$ ist eine Äquivalenzrelation.

Im Folgenden sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DFA ohne unerreichbare Zustände.

Definition 2.59 (Äquivalenz von Zuständen)

$$p \equiv_A q \quad \Leftrightarrow \quad (\forall w \in \Sigma^*.\ \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F)$$

Fakt 2.60

$\equiv_A$ ist eine Äquivalenzrelation.

Wir schreiben $\equiv$ statt $\equiv_A$ wenn A klar ist.

Im Folgenden sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DFA ohne unerreichbare Zustände.

Definition 2.59 (Äquivalenz von Zuständen)

$$p \equiv_A q \quad \Leftrightarrow \quad (\forall w \in \Sigma^*.\ \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F)$$

Fakt 2.60

$\equiv_A$ ist eine Äquivalenzrelation.

Wir schreiben $\equiv$ statt $\equiv_A$ wenn A klar ist.

Erinnerung:

Lemma 2.61

$$p \equiv_A q \implies \delta(p, a) \equiv_A \delta(q, a)$$

w

$\hat\delta(\delta(p,a),w) = \hat\delta(p,aw) \in F$

$\Longleftrightarrow$ $F \Longleftrightarrow$ $\hat\delta(q,aw) = \hat\delta(\delta(q,a),w)) =$

Im Folgenden sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DFA ohne unerreichbare Zustände.

Definition 2.59 (Äquivalenz von Zuständen)

$$p \equiv_A q \quad \Leftrightarrow \quad \boxed{(\forall w \in \Sigma^*}\ \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F)$$

Fakt 2.60

$\equiv_A$ ist eine Äquivalenzrelation.

Wir schreiben $\equiv$ statt $\equiv_A$ wenn A klar ist.

Erinnerung:

Lemma 2.61

$$p \equiv_A q \implies \delta(p, a) \equiv_A \delta(q, a)$$

Lemma 2.62
Algorithmus U liefert die unterscheidbaren Zustände, also $\not\equiv$.

Im Folgenden sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DFA ohne unerreichbare Zustände.

Definition 2.59 (Äquivalenz von Zuständen)

$$p \equiv_A q \quad \Leftrightarrow \quad (\forall w \in \Sigma^*.\ \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F)$$

Fakt 2.60

$\equiv_A$ ist eine Äquivalenzrelation.

Wir schreiben $\equiv$ statt $\equiv_A$ wenn A klar ist.

Erinnerung:

Lemma 2.61

$$p \equiv_A q \implies \delta(p, a) \equiv_A \delta(q, a)$$

Lemma 2.62
Algorithmus U liefert die unterscheidbaren Zustände, also $\not\equiv$.

In der weiteren Analyse beziehen wir uns direkt auf $\equiv$, nicht mehr auf den Algorithmus.

Die „Kollabierung“ von A bzgl. $\equiv$ ist der *Quotientenautomat*:

Definition 2.63 (Quotientenautomat)

$$A/\equiv \quad :=$$

Die „Kollabierung“ von A bzgl. $\equiv$ ist der *Quotientenautomat*:

Definition 2.63 (Quotientenautomat)

$$A/\equiv \quad := \quad (Q/\equiv,$$

Die „Kollabierung“ von A bzgl. $\equiv$ ist der *Quotientenautomat*:

Definition 2.63 (Quotientenautomat)

$$A/\equiv \quad := \quad (Q/\equiv,\ \Sigma,\ \delta',\ [q_0]_\equiv,\ F/\equiv)$$

$$\{[q]_\equiv \mid q \in F\}$$

Die „Kollabierung“ von A bzgl. $\equiv$ ist der *Quotientenautomat*:

Definition 2.63 (Quotientenautomat)

$$\begin{aligned} A/\equiv \quad &:= \quad (Q/\equiv,\ \Sigma,\ \delta',\ [q_0]_\equiv,\ F/\equiv) \\ \delta'([p]_\equiv, a) \quad &:= \quad [\delta(p,a)]_\equiv \end{aligned}$$

Die „Kollabierung“ von A bzgl. $\equiv$ ist der *Quotientenautomat*:

Definition 2.63 (Quotientenautomat)

$$A/{\equiv} \quad := \quad (Q/{\equiv},\ \Sigma,\ \delta',\ [q_0]_{\equiv},\ F/{\equiv})$$

Die „Kollabierung“ von A bzgl. $\equiv$ ist der *Quotientenautomat*:

Definition 2.63 (Quotientenautomat)

$$A/{\equiv} \quad :=$$

Beobachtung
Für $p := \hat{\delta}(q_0, u)$ und $q := \hat{\delta}(q_0, v)$ gilt:

$$p \equiv_A q \quad \Leftrightarrow \quad \forall w \in \Sigma^*.\ \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F$$

Beobachtung
Für $p := \hat{\delta}(q_0, u)$ und $q := \hat{\delta}(q_0, v)$ gilt:

$$\begin{aligned} p \equiv_A q \quad &\Leftrightarrow \quad \forall w \in \Sigma^*.\ \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F \\ &\Leftrightarrow \quad \forall w \in \Sigma^*.\ \hat{\delta}(q_0, uw) \in F \Leftrightarrow \hat{\delta}(q_0, vw) \in F \end{aligned}$$

Beobachtung
Für $p := \hat{\delta}(q_0, u)$ und $q := \hat{\delta}(q_0, v)$ gilt:

$$\begin{aligned} p \equiv_A q \quad &\Leftrightarrow \left(\forall w \in \Sigma^*.\ \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F \right) \\ &\Leftrightarrow \left(\forall w \in \Sigma^*.\ \hat{\delta}(q_0, uw) \in F \Leftrightarrow \hat{\delta}(q_0, vw) \in F \right) \\ &\Leftrightarrow \left(\forall w \in \Sigma^*.\ uw \in L(A) \Leftrightarrow vw \in L(A) \right) \end{aligned}$$

Beobachtung
Für $p := \hat{\delta}(q_0, u)$ und $q := \hat{\delta}(q_0, v)$ gilt:

$$\begin{aligned} p \equiv_A q \quad &\Leftrightarrow \quad \forall w \in \Sigma^*.\ \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F \\ &\Leftrightarrow \quad \forall w \in \Sigma^*.\ \hat{\delta}(q_0, uw) \in F \Leftrightarrow \hat{\delta}(q_0, vw) \in F \\ &\Leftrightarrow \quad \forall w \in \Sigma^*.\ \boxed{u}w \in L(A) \Leftrightarrow \boxed{v}w \in L(A) \end{aligned}$$

Definition 2.65
Jede Sprache $L \subseteq \Sigma^*$ induziert eine Äquivalenzrelation $\equiv_L \subseteq \Sigma^* \times \Sigma^*$:

Beobachtung
Für $p := \hat{\delta}(q_0, u)$ und $q := \hat{\delta}(q_0, v)$ gilt:

$$\begin{aligned} p \equiv_A q \quad &\Leftrightarrow \quad \forall w \in \Sigma^*.\ \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F \\ &\Leftrightarrow \quad \forall w \in \Sigma^*.\ \hat{\delta}(q_0, uw) \in F \Leftrightarrow \hat{\delta}(q_0, vw) \in F \\ &\Leftrightarrow \quad \forall w \in \Sigma^*.\ uw \in L(A) \Leftrightarrow vw \in L(A) \end{aligned}$$

Definition 2.65
Jede Sprache $L \subseteq \Sigma^*$ induziert eine Äquivalenzrelation $\equiv_L \subseteq \Sigma^* \times \Sigma^*$:

$$u \equiv_L v \quad \Leftrightarrow \quad \forall w \in \Sigma^*.\ uw \in L \Leftrightarrow vw \in L$$

Beobachtung
Für $p := \hat{\delta}(q_0, u)$ und $q := \hat{\delta}(q_0, v)$ gilt:

$$\begin{aligned} p \equiv_A q \quad &\Leftrightarrow \quad \forall w \in \Sigma^*.\ \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F \\ &\Leftrightarrow \quad \forall w \in \Sigma^*.\ \hat{\delta}(q_0, uw) \in F \Leftrightarrow \hat{\delta}(q_0, vw) \in F \\ &\Leftrightarrow \quad \forall w \in \Sigma^*.\ uw \in L(A) \Leftrightarrow vw \in L(A) \end{aligned}$$

Definition 2.65
Jede Sprache $L \subseteq \Sigma^*$ induziert eine Äquivalenzrelation $\equiv_L \subseteq \Sigma^* \times \Sigma^*$:

$$u \equiv_L v \quad \Leftrightarrow \quad \forall w \in \Sigma^*.\ uw \in L \Leftrightarrow vw \in L$$

D.h. u und v sind duch Anhängen von Wörtern bzgl $\in L$ nicht unterscheidbar.

Beobachtung

Für $p := \hat{\delta}(q_0, u)$ und $q := \hat{\delta}(q_0, v)$ gilt:

$$\begin{aligned} p \equiv_A q \quad &\Leftrightarrow \quad \forall w \in \Sigma^*.\ \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F \\ &\Leftrightarrow \quad \forall w \in \Sigma^*.\ \hat{\delta}(q_0, uw) \in F \Leftrightarrow \hat{\delta}(q_0, vw) \in F \\ &\Leftrightarrow \quad \forall w \in \Sigma^*.\ uw \in L(A) \Leftrightarrow vw \in L(A) \end{aligned}$$

Definition 2.65

Jede Sprache $L \subseteq \Sigma^*$ induziert eine Äquivalenzrelation $\equiv_L \subseteq \Sigma^* \times \Sigma^*$:

$$u \equiv_L v \quad \Leftrightarrow \quad \forall w \in \Sigma^*.\ uw \in L \Leftrightarrow vw \in L$$

D.h. u und v sind duch Anhängen von Wörtern bzgl $\in L$ nicht unterscheidbar.

Obige Beobachtung lässt sich nun schreiben als

$$\boxed{u \equiv_{L(A)} v} \quad \Leftrightarrow \quad \boxed{\hat{\delta}(q_0, u) \equiv_A \hat{\delta}(q_0, v)}$$

$$u \equiv_{L(A)} v \quad \Leftrightarrow \quad \hat{\delta}(q_0, u) \equiv_A \hat{\delta}(q_0, v)$$

$$u \equiv_{L(A)} v \quad \Leftrightarrow \quad \hat{\delta}(q_0, u) \equiv_A \hat{\delta}(q_0, v)$$

Da alle Zustände von q_0 erreichbar sind, gilt sogar: Die Abbildung

$$[u]_{\equiv_{L(A)}} \quad \mapsto \quad [\hat{\delta}(q_0, u)]_{\equiv_A}$$

ist eine Bijektion zwischen den $\boxed{\equiv_{L(A)}}$ und $\boxed{\equiv_A}$ Äquivalenzklassen.

$$u \equiv_{L(A)} v \quad \Leftrightarrow \quad \hat{\delta}(q_0, u) \equiv_A \hat{\delta}(q_0, v)$$

Da alle Zustände von q_0 erreichbar sind, gilt sogar: Die Abbildung

$$[u]_{\equiv_{L(A)}} \quad \mapsto \quad [\hat{\delta}(q_0, u)]_{\equiv_A}$$

ist eine Bijektion zwischen den $\equiv_{L(A)}$ und $\equiv_A$ Äquivalenzklassen.

Satz 2.66

Ist A ein DFA ohne unerreichbare Zustände, so ist der von Algorithmus U berechnete Quotientenautomat $A/\equiv$ ein minimaler DFA für $L(A)$.

$$u \equiv_{L(A)} v \quad \Leftrightarrow \quad \hat{\delta}(q_0, u) \equiv_A \hat{\delta}(q_0, v)$$

Da alle Zustände von q_0 erreichbar sind, gilt sogar: Die Abbildung

$$[u]_{\equiv_{L(A)}} \quad \mapsto \quad [\hat{\delta}(q_0, u)]_{\equiv_A}$$

ist eine Bijektion zwischen den $\equiv_{L(A)}$ und $\equiv_A$ Äquivalenzklassen.

Satz 2.66

Ist A ein DFA ohne unerreichbare Zustände, so ist der von Algorithmus U *berechnete Quotientenautomat $A/\equiv$ ein minimaler DFA für $L(A)$.*

Beweis:

Sei $L := L(A)$ und A' ein DFA mit $L(A') = L$.

$$u \equiv_{L(A)} v \quad \Leftrightarrow \quad \hat{\delta}(q_0, u) \equiv_A \hat{\delta}(q_0, v)$$

Da alle Zustände von q_0 erreichbar sind, gilt sogar: Die Abbildung

$$[u]_{\equiv_{L(A)}} \quad \mapsto \quad [\hat{\delta}(q_0, u)]_{\equiv_A}$$

ist eine Bijektion zwischen den $\equiv_{L(A)}$ und $\equiv_A$ Äquivalenzklassen.

Satz 2.66

Ist A ein DFA ohne unerreichbare Zustände, so ist der von Algorithmus U *berechnete Quotientenautomat $A/\equiv$ ein minimaler DFA für $L(A)$.*

Beweis:

Sei $L := L(A)$ und A' ein DFA mit $L(A') = L$. Dann gilt:

$$|Q'|$$

$$u \equiv_{L(A)} v \quad \Leftrightarrow \quad \hat{\delta}(q_0, u) \equiv_A \hat{\delta}(q_0, v)$$

Da alle Zustände von q_0 erreichbar sind, gilt sogar: Die Abbildung

$$[u]_{\equiv_{L(A)}} \quad \mapsto \quad [\hat{\delta}(q_0, u)]_{\equiv_A}$$

ist eine Bijektion zwischen den $\equiv_{L(A)}$ und $\equiv_A$ Äquivalenzklassen.

Satz 2.66

Ist A ein DFA ohne unerreichbare Zustände, so ist der von Algorithmus U *berechnete Quotientenautomat $A/\equiv$ ein minimaler DFA für $L(A)$.*

Beweis:

Sei $L := L(A)$ und A' ein DFA mit $L(A') = L$. Dann gilt:

$$|Q'| \geq |Q'/{\equiv_{A'}}|$$

$$u \equiv_{L(A)} v \quad \Leftrightarrow \quad \hat{\delta}(q_0, u) \equiv_A \hat{\delta}(q_0, v)$$

Da alle Zustände von q_0 erreichbar sind, gilt sogar: Die Abbildung

$$[u]_{\equiv_{L(A)}} \quad \mapsto \quad [\hat{\delta}(q_0, u)]_{\equiv_A}$$

ist eine Bijektion zwischen den $\equiv_{L(A)}$ und $\equiv_A$ Äquivalenzklassen.

Satz 2.66

Ist A ein DFA ohne unerreichbare Zustände, so ist der von Algorithmus U *berechnete Quotientenautomat $A/\equiv$ ein minimaler DFA für $L(A)$.*

Beweis:

Sei $L := L(A)$ und A' ein DFA mit $L(A') = L$. Dann gilt:

$$|Q'| \geq |Q'/{\equiv_{A'}}| = |\Sigma^*/{\equiv_L}|$$

$$u \equiv_{L(A)} v \quad \Leftrightarrow \quad \hat\delta(q_0, u) \equiv_A \hat\delta(q_0, v)$$

Da alle Zustände von q_0 erreichbar sind, gilt sogar: Die Abbildung

$$[u]_{\equiv_{L(A)}} \quad \mapsto \quad [\hat\delta(q_0, u)]_{\equiv_A}$$

ist eine Bijektion zwischen den $\equiv_{L(A)}$ und $\equiv_A$ Äquivalenzklassen.

Satz 2.66

Ist A ein DFA ohne unerreichbare Zustände, so ist der von Algorithmus U berechnete Quotientenautomat $A/\equiv$ ein minimaler DFA für $L(A)$.

Beweis:

Sei $L := L(A)$ und A' ein DFA mit $L(A') = L$. Dann gilt:

$$|Q'| \geq |Q'/\equiv_{A'}| = |\Sigma^*/\equiv_L| = |Q/\equiv_A|$$

□

$$u \equiv_{L(A)} v \quad \Leftrightarrow \quad \hat\delta(q_0, u) \equiv_A \hat\delta(q_0, v)$$

Da alle Zustände von q_0 erreichbar sind, gilt sogar: Die Abbildung

$$[u]_{\equiv_{L(A)}} \quad \mapsto \quad [\hat\delta(q_0, u)]_{\equiv_A}$$

ist eine Bijektion zwischen den $\equiv_{L(A)}$ und $\equiv_A$ Äquivalenzklassen.

Satz 2.66

Ist A ein DFA ohne unerreichbare Zustände, so ist der von Algorithmus U berechnete Quotientenautomat $A/\equiv$ ein minimaler DFA für $L(A)$.

Beweis:

Sei $L := L(A)$ und A' ein DFA mit $L(A') = L$. Dann gilt:

$$|Q'| \geq |Q'/\equiv_{A'}| = |\Sigma^*/\equiv_L| = |Q/\equiv_A|$$

□

$$u \equiv_{L(A)} v \quad \Leftrightarrow \quad \hat\delta(q_0, u) \equiv_A \hat\delta(q_0, v)$$

Da alle Zustände von q_0 erreichbar sind, gilt sogar: Die Abbildung

$$[u]_{\equiv_{L(A)}} \quad \mapsto \quad [\hat\delta(q_0, u)]_{\equiv_A}$$

ist eine Bijektion zwischen den $\equiv_{L(A)}$ und $\equiv_A$ Äquivalenzklassen.

Satz 2.66

Ist A ein DFA ohne unerreichbare Zustände, so ist der von Algorithmus U berechnete Quotientenautomat $A/\equiv$ ein minimaler DFA für $L(A)$.

Beweis:

Sei $L := L(A)$ und A' ein DFA mit $L(A') = L$. Dann gilt:

$$|Q'| \geq |Q'/\equiv_{A'}| = |\Sigma^*/\equiv_L| = |Q/\equiv_A|$$

□

Es gilt sogar (Übung!):

Fakt 2.67

Alle Quotientenautomaten $A/\equiv_A$ für die gleiche Sprache $L(A)$ haben die gleiche Struktur, d.h. sie unterscheiden sich nur durch eine Umbenennung der Zustände.

Es gilt sogar (Übung!):

Fakt 2.67
Alle Quotientenautomaten $A/{\equiv_A}$ für die gleiche Sprache $L(A)$ haben die gleiche Struktur, d.h. sie unterscheiden sich nur durch eine Umbenennung der Zustände.

Daher beschriften wir die Zustände des kanonischen Minimalautomaten für eine Sprache L mit $\equiv_L$ Äquivalenzklassen.

Beispiel 2.68
Sei $L := \{w \in \{0,1\}^* \mid w \text{ endet mit } 00\}$.

$\equiv_L$

Es gilt sogar (Übung!):

Fakt 2.67
Alle Quotientenautomaten $A/{\equiv_A}$ für die gleiche Sprache $L(A)$ haben die gleiche Struktur, d.h. sie unterscheiden sich nur durch eine Umbenennung der Zustände.

Daher beschriften wir die Zustände des kanonischen Minimalautomaten für eine Sprache L mit $\equiv_L$ Äquivalenzklassen.

Beispiel 2.68
Sei $L := \{w \in \{0,1\}^* \mid w \text{ endet mit } 00\}$.
Die einzigen drei $\equiv_L$ Äquivalenzklassen sind:

$$[\epsilon]_{\equiv_L} = \{w \mid w \text{ endet nicht mit } 0\}$$

Es gilt sogar (Übung!):

Fakt 2.67
Alle Quotientenautomaten $A/{\equiv_A}$ für die gleiche Sprache $L(A)$ haben die gleiche Struktur, d.h. sie unterscheiden sich nur durch eine Umbenennung der Zustände.

Daher beschriften wir die Zustände des kanonischen Minimalautomaten für eine Sprache L mit $\equiv_L$ Äquivalenzklassen.

Beispiel 2.68
Sei $L := \{w \in \{0,1\}^* \mid w \text{ endet mit } 00\}$.
Die einzigen drei $\equiv_L$ Äquivalenzklassen sind:

$$[\epsilon]_{\equiv_L} = \{w \mid w \text{ endet nicht mit } 0\}$$

$$[0]_{\equiv_L} = \{w \mid w \text{ endet mit 0, aber nicht mit } 00\}$$

Es gilt sogar (Übung!):

Fakt 2.67
Alle Quotientenautomaten $A/{\equiv_A}$ für die gleiche Sprache $L(A)$ haben die gleiche Struktur, d.h. sie unterscheiden sich nur durch eine Umbenennung der Zustände.

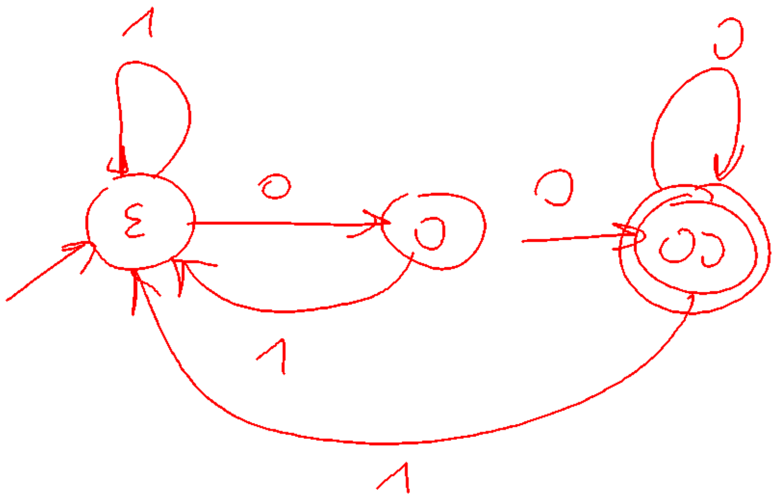
Daher beschriften wir die Zustände des kanonischen Minimalautomaten für eine Sprache L mit $\equiv_L$ Äquivalenzklassen.

Beispiel 2.68
Sei $L := \{w \in \{0,1\}^* \mid w \text{ endet mit } 00\}$.
Die einzigen drei $\equiv_L$ Äquivalenzklassen sind:

$$[\epsilon]_{\equiv_L} = \{w \mid w \text{ endet nicht mit } 0\}$$

$$[0]_{\equiv_L} = \{w \mid w \text{ endet mit 0, aber nicht mit } 00\}$$

$$[00]_{\equiv_L} = \{w \mid w \text{ endet mit } 00\}$$



Definition 2.69 (Kanonischer Minimalautomat)

$$M_L \ :=$$

Definition 2.69 (Kanonischer Minimalautomat)

$$M_L \ := \ (\Sigma^*/{\equiv_L},$$

Definition 2.69 (Kanonischer Minimalautomat)

$$M_L \ := \ (\Sigma^*/{\equiv_L},\ \Sigma,\ \delta_L,\ [\epsilon]_{\equiv_L},\ F_L)$$

mit $\delta_L([w]_{\equiv_L}, a) := [wa]_{\equiv_L}$

Definition 2.69 (Kanonischer Minimalautomat)

$$M_L \;:=\; (\Sigma^*/{\equiv_L},\ \Sigma,\ \delta_L,\ [\epsilon]_{\equiv_L},\ F_L)$$

mit $\delta_L([w]_{\equiv_L}, a) := [wa]_{\equiv_L}$ und $F_L := \{[w]_{\equiv_L} \mid w \in L\}$.

Definition 2.69 (Kanonischer Minimalautomat)

$$M_L \;:=\; (\Sigma^*/{\equiv_L},\ \Sigma,\ \delta_L,\ [\epsilon]_{\equiv_L},\ F_L)$$

mit $\delta_L([w]_{\equiv_L}, a) := [wa]_{\equiv_L}$ und $F_L := \{[w]_{\equiv_L} \mid w \in L\}$.

Man sieht leicht: δ_L ist wohldefiniert

Definition 2.69 (Kanonischer Minimalautomat)

$$M_L \;:=\; (\Sigma^*/{\equiv_L},\ \Sigma,\ \delta_L,\ [\epsilon]_{\equiv_L},\ F_L)$$

mit $\delta_L([w]_{\equiv_L}, a) := [wa]_{\equiv_L}$ und $F_L := \{[w]_{\equiv_L} \mid w \in L\}$.

Man sieht leicht: δ_L ist wohldefiniert und $\hat{\delta}_L([\epsilon]_{\equiv_L}, w) = [w]_{\equiv_L}$.
Dann gilt offensichtlich $L(M_L) = L$.

Definition 2.69 (Kanonischer Minimalautomat)

$$\boxed{M_L} \;:=\; (\boxed{\Sigma^*/{\equiv_L},}\ \Sigma,\ \delta_L,\ [\epsilon]_{\equiv_L},\ F_L)$$

mit $\delta_L([w]_{\equiv_L}, a) := [wa]_{\equiv_L}$ und $F_L := \{[w]_{\equiv_L} \mid w \in L\}$.

Man sieht leicht: δ_L ist wohldefiniert und $\hat{\delta}_L([\epsilon]_{\equiv_L}, w) = [w]_{\equiv_L}$.
Dann gilt offensichtlich $L(M_L) = L$.

Satz 2.70 (Myhill-Nerode)

Eine Sprache $L \subseteq \Sigma^*$ ist genau dann regulär, wenn $\boxed{\equiv_L}$ endlich viele Äquivalenzklassen hat.

Definition 2.69 (Kanonischer Minimalautomat)

$$M_L \;:=\; (\Sigma^*/\!\equiv_L,\ \Sigma,\ \delta_L,\ [\epsilon]_{\equiv_L},\ F_L)$$

mit $\delta_L([w]_{\equiv_L}, a) := [wa]_{\equiv_L}$ und $F_L := \{[w]_{\equiv_L} \mid w \in L\}$.

Man sieht leicht: δ_L ist wohldefiniert und $\hat{\delta}_L([\epsilon]_{\equiv_L}, w) = [w]_{\equiv_L}$. Dann gilt offensichtlich $L(M_L) = L$.

Satz 2.70 (Myhill-Nerode)

Eine Sprache $L \subseteq \Sigma^*$ ist genau dann regulär, wenn $\equiv_L$ endlich viele Äquivalenzklassen hat.

Beweis:
„$\Longrightarrow$": Ist L regulär, so wird L von einem DFA A akzeptiert. Daher hat $\equiv_L$ so viele Äquivalenzklassen wie $A/\!\equiv_A$ Zustände.
„$\Longleftarrow$": Hat $\equiv_L$ endlich viele Äquivalenzklasse, so ist M_L ein DFA mit $L(M_L) = L$. $\square$

Definition 2.69 (Kanonischer Minimalautomat)

$$M_L \;:=\; (\Sigma^*/\!\equiv_L,\ \Sigma,\ \delta_L,\ [\epsilon]_{\equiv_L},\ F_L)$$

mit $\delta_L([w]_{\equiv_L}, a) := [wa]_{\equiv_L}$ und $F_L := \{[w]_{\equiv_L} \mid w \in L\}$.

Man sieht leicht: δ_L ist wohldefiniert und $\hat{\delta}_L([\epsilon]_{\equiv_L}, w) = [w]_{\equiv_L}$. Dann gilt offensichtlich $L(M_L) = L$.

Satz 2.70 (Myhill-Nerode)

Eine Sprache $L \subseteq \Sigma^*$ ist genau dann regulär, wenn $\equiv_L$ endlich viele Äquivalenzklassen hat.

Beweis:
„$\Longrightarrow$": Ist L regulär, so wird L von einem DFA A akzeptiert. Daher hat $\equiv_L$ so viele Äquivalenzklassen wie $A/\!\equiv_A$ Zustände.
„$\Longleftarrow$": Hat $\equiv_L$ endlich viele Äquivalenzklasse, so ist M_L ein DFA mit $L(M_L) = L$. $\square$

Wie sieht M_L aus wenn L *nicht* regulär ist?

Definition 2.69 (Kanonischer Minimalautomat)

$$M_L \;:=\; (\Sigma^*/\!\equiv_L,\ \Sigma,\ \delta_L,\ [\epsilon]_{\equiv_L},\ F_L)$$

mit $\delta_L([w]_{\equiv_L}, a) := [wa]_{\equiv_L}$ und $F_L := \{[w]_{\equiv_L} \mid w \in L\}$.

Man sieht leicht: δ_L ist wohldefiniert und $\hat{\delta}_L([\epsilon]_{\equiv_L}, w) = [w]_{\equiv_L}$. Dann gilt offensichtlich $L(M_L) = L$.

Satz 2.70 (Myhill-Nerode)

Eine Sprache $L \subseteq \Sigma^*$ ist genau dann regulär, wenn $\equiv_L$ endlich viele Äquivalenzklassen hat.

Beweis:
„$\Longrightarrow$": Ist L regulär, so wird L von einem DFA A akzeptiert. Daher hat $\equiv_L$ so viele Äquivalenzklassen wie $A/\!\equiv_A$ Zustände.
„$\Longleftarrow$": Hat $\equiv_L$ endlich viele Äquivalenzklasse, so ist M_L ein DFA mit $L(M_L) = L$. $\square$

Wie sieht M_L aus wenn L *nicht* regulär ist?

Beispiel 2.71

$L = \{0^i 1^i \mid i \in \mathbb{N}\}$

Wie sieht M_L aus wenn L *nicht* regulär ist?

Beispiel 2.71

$L = \{0^i 1^i \mid i \in \mathbb{N}\}$

$\to [\epsilon]$

Wie sieht M_L aus wenn L *nicht* regulär ist?

Beispiel 2.71

$L = \{0^i 1^i \mid i \in \mathbb{N}\}$

$\to [\epsilon] \xrightarrow{0} [0]$

Wie sieht M_L aus wenn L *nicht* regulär ist?

Beispiel 2.71

$L = \{0^i 1^i \mid i \in \mathbb{N}\}$

$\to [\epsilon] \xrightarrow{0} [0] \xrightarrow{0} [0^2]$

Wie sieht M_L aus wenn L *nicht* regulär ist?

Beispiel 2.71

$L = \{0^i 1^i \mid i \in \mathbb{N}\}$

$$\to [\epsilon] \xrightarrow{0} [0] \xrightarrow{0} [0^2] \xrightarrow{0} \cdots \xrightarrow{0} [0^i]$$

Wie sieht M_L aus wenn L *nicht* regulär ist?

Beispiel 2.71

$L = \{0^i 1^i \mid i \in \mathbb{N}\}$

$$\begin{tikzcd} \arrow[r] & {[\epsilon]} \arrow[r, "0"] & {[0]} \arrow[r, "0"] \arrow[d, "1"] & {[0^2]} \arrow[r, "0"] & \cdots \arrow[r, "0"] & {[0^i]} \arrow[r, "0"] & \cdots \\ & & {[01]} & & & & \end{tikzcd}$$

Wie sieht M_L aus wenn L *nicht* regulär ist?

Beispiel 2.71

$L = \{0^i 1^i \mid i \in \mathbb{N}\}$

$$\begin{tikzcd} \arrow[r] & {[\epsilon]} \arrow[r, "0"] & {[0]} \arrow[r, "0"] \arrow[d, "1"] & {[0^2]} \arrow[r, "0"] \arrow[d, "1"] & \cdots \arrow[r, "0"] & {[0^i]} \arrow[r, "0"] & \cdots \\ & & {[01]} & {[0^21]} & & & \end{tikzcd}$$

Wie sieht M_L aus wenn L *nicht* regulär ist?

Beispiel 2.71

$L = \{0^i 1^i \mid i \in \mathbb{N}\}$

$$\begin{tikzcd} \arrow[r] & {[\epsilon]} \arrow[r, "0"] & {[0]} \arrow[r, "0"] \arrow[d, "1"] & {[0^2]} \arrow[r, "0"] \arrow[d, "1"] & \cdots \arrow[r, "0"] & {[0^i]} \arrow[r, "0"] \arrow[d, "1"] & \cdots \\ & & {[01]} & {[0^21]} & & {[0^i1]} & \end{tikzcd}$$

Wie sieht M_L aus wenn L *nicht* regulär ist?

Beispiel 2.71

$L = \{0^i 1^i \mid i \in \mathbb{N}\}$

$$\begin{array}{ccccccccc} \to [\epsilon] & \xrightarrow{0} & [0] & \xrightarrow{0} & [0^2] & \xrightarrow{0} \cdots \xrightarrow{0} & [0^i] & \xrightarrow{0} & \cdots \\ & & \downarrow 1 & & \downarrow 1 & & \downarrow 1 & & \\ & & [01] & \xleftarrow{1} & [0^2 1] & & [0^i 1] & & \end{array}$$

Wie sieht M_L aus wenn L *nicht* regulär ist?

Beispiel 2.71

$L = \{0^i 1^i \mid i \in \mathbb{N}\}$

$$\begin{array}{ccccccccc} \to [\epsilon] & \xrightarrow{0} & [0] & \xrightarrow{0} & [0^2] & \xrightarrow{0} \cdots \xrightarrow{0} & [0^i] & \xrightarrow{0} & \cdots \\ & & \downarrow 1 & & \downarrow 1 & & \downarrow 1 & & \\ & & [01] & \xleftarrow{1} & [0^2 1] & \xleftarrow{1} \cdots \xleftarrow{1} & [0^i 1] & & \end{array}$$

Wie sieht M_L aus wenn L *nicht* regulär ist?

Beispiel 2.71

$L = \{0^i 1^i \mid i \in \mathbb{N}\}$

$$\begin{array}{ccccccccc} \to [\epsilon] & \xrightarrow{0} & [0] & \xrightarrow{0} & [0^2] & \xrightarrow{0} \cdots \xrightarrow{0} & [0^i] & \xrightarrow{0} & \cdots \\ & & \downarrow 1 & & \downarrow 1 & & \downarrow 1 & & \\ & & [01] & \xleftarrow{1} & [0^2 1] & \xleftarrow{1} \cdots \xleftarrow{1} & [0^i 1] & \xleftarrow{1} & \cdots \end{array}$$

Warum $0^i \not\equiv_L 0^j$ falls $i \neq j$?

Was fehlt noch?

Wie sieht M_L aus wenn L *nicht* regulär ist?

Beispiel 2.71

$L = \{0^i 1^i \mid i \in \mathbb{N}\}$

$$\begin{array}{ccccccccc} \to [\epsilon] & \xrightarrow{0} & [0] & \xrightarrow{0} & [0^2] & \xrightarrow{0} \cdots \xrightarrow{0} & [0^i] & \xrightarrow{0} & \cdots \\ & & \downarrow 1 & & \downarrow 1 & & \downarrow 1 & & \\ & & [01] & \xleftarrow{1} & [0^2 1] & \xleftarrow{1} \cdots \xleftarrow{1} & [0^i 1] & \xleftarrow{1} & \cdots \end{array}$$

Warum $0^i \not\equiv_L 0^j$ falls $i \neq j$?

Was fehlt noch?

Vollständige Methode um Nichtregularität von L zu zeigen:

Wie sieht M_L aus wenn L *nicht* regulär ist?

Beispiel 2.71

$L = \{0^i 1^i \mid i \in \mathbb{N}\}$

$$\begin{array}{ccccccccc}
\to [\epsilon] & \xrightarrow{0} & [0] & \xrightarrow{0} & [0^2] & \xrightarrow{0} \cdots \xrightarrow{0} & [0^i] & \xrightarrow{0} & \cdots \\
 & & \downarrow 1 & & \downarrow 1 & & \downarrow 1 & & \\
 & & [01] & \xleftarrow{1} & [0^2 1] & \xleftarrow{1} \cdots \xleftarrow{1} & [0^i 1] & \xleftarrow{1} & \cdots
\end{array}$$

Warum $0^i \not\equiv_L 0^j$ falls $i \neq j$?
Was fehlt noch?

Vollständige Methode um Nichtregularität von L zu zeigen:

Gib unendliche Menge $w_1, w_2, \ldots$ an mit $w_i \not\equiv_L w_j$ falls $i \neq j$.

Bemerkung

Eindeutigkeit des minimalen Automaten (modulo Umbenennung der Zustände) gilt nur bei DFAs, nicht bei NFAs!

Knobelaufgabe:

Nach Def. gilt $p \not\equiv_A q$ gdw $\exists w.\ \hat{\delta}(p, w) \in F \not\Leftrightarrow \hat{\delta}(q, w) \in F$

Knobelaufgabe:

Nach Def. gilt $p \not\equiv_A q$ gdw $\exists w.\ \hat{\delta}(p, w) \in F \not\Leftrightarrow \hat{\delta}(q, w) \in F$

Man zeige: Falls $p \not\equiv_A q$, dann gibt es ein w der Länge $< |Q|$, das p und q unterscheidet, d.h. $\hat{\delta}(p, w) \in F \not\Leftrightarrow \hat{\delta}(q, w) \in F$.

Rückblick reguläre Sprachen

- $\boxed{\text{DFA}}$ $\boxed{\text{NFA, } \epsilon\text{-NFA}}$ und $\boxed{\text{RE}}$ definieren die gleiche Sprachklasse.

Rückblick reguläre Sprachen

- DFA, NFA, ϵ-NFA und RE definieren die gleiche Sprachklasse.
  - Potenzmengenkonstruktion (exponentiell)

Rückblick reguläre Sprachen

- DFA, NFA, ϵ-NFA und RE definieren die gleiche Sprachklasse.
  - Potenzmengenkonstruktion (exponentiell)
  - Strukturelle Übersetzung von RE nach ϵ-NFA
  - Übersetzung NFA nach RE, zB über Gleichungssysteme (exponentiell)

Rückblick reguläre Sprachen

- DFA, NFA, ϵ-NFA und RE definieren die gleiche Sprachklasse.
  - Potenzmengenkonstruktion (exponentiell)
  - Strukturelle Übersetzung von RE nach ϵ-NFA
  - Übersetzung NFA nach RE, zB über Gleichungssysteme (exponentiell)
- Regularitätserhaltende Konstruktionen auf Sprachen bzw Automaten:

Rückblick reguläre Sprachen

- DFA, NFA, ϵ-NFA und RE definieren die gleiche Sprachklasse.
  - Potenzmengenkonstruktion (exponentiell)
  - Strukturelle Übersetzung von RE nach ϵ-NFA
  - Übersetzung NFA nach RE, zB über Gleichungssysteme (exponentiell)
- Regularitätserhaltende Konstruktionen auf Sprachen bzw Automaten:
  - $\cup$, $\cap$, $\ldots$, ^{R}, $\ldots$
  - Für welche Darstellung wie teuer? (Komplement, Produkt, $\ldots$)

Rückblick reguläre Sprachen

- DFA, NFA, ϵ-NFA und RE definieren die gleiche Sprachklasse.
  - Potenzmengenkonstruktion (exponentiell)
  - Strukturelle Übersetzung von RE nach ϵ-NFA
  - Übersetzung NFA nach RE, zB über Gleichungssysteme (exponentiell)
- Regularitätserhaltende Konstruktionen auf Sprachen bzw Automaten:
  - $\cup$, $\cap$, $\ldots$, ^{R}, $\ldots$
  - Für welche Darstellung wie teuer? (Komplement, Produkt, $\ldots$)
  - NFA kompakt aber oft teuer; DFA oft billiger

Rückblick reguläre Sprachen

- DFA, NFA, ϵ-NFA und RE definieren die gleiche Sprachklasse.
  - Potenzmengenkonstruktion (exponentiell)
  - Strukturelle Übersetzung von RE nach ϵ-NFA
  - Übersetzung NFA nach RE, zB über Gleichungssysteme (exponentiell)
- Regularitätserhaltende Konstruktionen auf Sprachen bzw Automaten:
  - $\cup$, $\cap$, $\ldots$, ^{R}, $\ldots$
  - Für welche Darstellung wie teuer? (Komplement, Produkt, $\ldots$)
  - NFA kompakt aber oft teuer; DFA oft billiger
- Entscheidungsprobleme auf Automaten und REs:

Rückblick reguläre Sprachen

- DFA, NFA, ϵ-NFA und RE definieren die gleiche Sprachklasse.
  - Potenzmengenkonstruktion (exponentiell)
  - Strukturelle Übersetzung von RE nach ϵ-NFA
  - Übersetzung NFA nach RE, zB über Gleichungssysteme (exponentiell)
- Regularitätserhaltende Konstruktionen auf Sprachen bzw Automaten:
  - $\cup$, $\cap$, $\ldots$, ^{R}, $\ldots$
  - Für welche Darstellung wie teuer? (Komplement, Produkt, $\ldots$)
  - NFA kompakt aber oft teuer; DFA oft billiger
- Entscheidungsprobleme auf Automaten und REs:
  - Direkt entscheidbar? Auf Automat? (Oft mit Erreichbarkeit) Auf RE?

Rückblick reguläre Sprachen

- DFA, NFA, ϵ-NFA und RE definieren die gleiche Sprachklasse.
  - Potenzmengenkonstruktion (exponentiell)
  - Strukturelle Übersetzung von RE nach ϵ-NFA
  - Übersetzung NFA nach RE, zB über Gleichungssysteme (exponentiell)
- Regularitätserhaltende Konstruktionen auf Sprachen bzw Automaten:
  - $\cup$, $\cap$, $\dots$, ^{R}, $\dots$
  - Für welche Darstellung wie teuer? (Komplement, Produkt, $\dots$)
  - NFA kompakt aber oft teuer; DFA oft billiger
- Entscheidungsprobleme auf Automaten und REs:
  - Direkt entscheidbar?
    Auf Automat? (Oft mit Erreichbarkeit) Auf RE?
  - Reduzierbar auf anderes Problem?
    ZB $L(A) \subseteq L(B)$ auf $L(A) \cap \overline{L(B)} = \emptyset$

Rückblick reguläre Sprachen

- DFA, NFA, ϵ-NFA und RE definieren die gleiche Sprachklasse.
  - Potenzmengenkonstruktion (exponentiell)
  - Strukturelle Übersetzung von RE nach ϵ-NFA
  - Übersetzung NFA nach RE, zB über Gleichungssysteme (exponentiell)
- Regularitätserhaltende Konstruktionen auf Sprachen bzw Automaten:
  - $\cup$, $\cap$, $\dots$, ^{R}, $\dots$
  - Für welche Darstellung wie teuer? (Komplement, Produkt, $\dots$)
  - NFA kompakt aber oft teuer; DFA oft billiger
- Entscheidungsprobleme auf Automaten und REs:
  - Direkt entscheidbar?
    Auf Automat? (Oft mit Erreichbarkeit) Auf RE?
  - Reduzierbar auf anderes Problem?
    ZB $L(A) \subseteq L(B)$ auf $L(A) \cap \overline{L(B)} = \emptyset$
  - Für welche Darstellung wie teuer?

- Pumping-Lemma
- Äquivalenzen $\equiv$ zw REs:
  - Standardregeln: Kommutativität etc, Beweis mit $L(.)$
  - $\alpha \equiv \beta$ entscheidbar da $L(\alpha) = L(\beta)$ über Automaten entscheidbar
  - Auch für REs mit Variablen entscheidbar: betrachte Variablen als Konstanten

---

- Pumping-Lemma
- Äquivalenzen $\equiv$ zw REs:
  - Standardregeln: Kommutativität etc, Beweis mit $L(.)$
  - $\alpha \equiv \beta$ entscheidbar da $L(\alpha) = L(\beta)$ über Automaten entscheidbar
  - Auch für REs mit Variablen entscheidbar: betrachte Variablen als Konstanten
  - Gleichungssysteme lösbar
    durch Variablenelimination mit Ardens Lemma

- Pumping-Lemma
- Äquivalenzen $\equiv$ zw REs:
  - Standardregeln: Kommutativität etc, Beweis mit $L(.)$
  - $\alpha \equiv \beta$ entscheidbar da $L(\alpha) = L(\beta)$ über Automaten entscheidbar
  - Auch für REs mit Variablen entscheidbar: betrachte Variablen als Konstanten
  - Gleichungssysteme lösbar
    durch Variablenelimination mit Ardens Lemma
- Minimierung