Informatik III - Tutorium IX & X (SR -107)

Tut Nr. 2 – Üb1, NEA $\rightarrow$ DEA, Minimierung, PL

David Münch

Universität Karlsruhe (TH)
Institut für Informatik
IAKS Beth

4. November 2008

Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

Inhaltsverzeichnis

Inhaltsverzeichnis

Inhaltsverzeichnis

Inhaltsverzeichnis

Organisatorisches

Email: muenchdavid@gmail.com

`https://www.stud.uni-karlsruhe.de/~uhbro/`

Tutorium 09: Mittwochs 8:00 Uhr - Raum -107
Tutorium 10: Mittwochs 9:45 Uhr - Raum -107

Übungsblattabgabe Donnerstags.

Organisatorisches

Deckblatt benutzen: `http://www.stud.uni-karlsruhe.de/~unbdh/deckblatt/index.php?course=6`

Gruppenarbeit erwünscht, aber jeder muss handschriftliche Lösung mit Namen aller Gruppenteilnehmer abgeben.

50 % der Punkte sind notwendig für den Schein.

WS 2008/09 letzte Möglichkeit für den Schein!

Was wollen wir heute erreichen?

Zu DEAs, regulären Ausdrücken, NEAs. Grammatiken wollen wir:

Was wollen wir heute erreichen?

Zu DEAs, regulären Ausdrücken, NEAs. Grammatiken wollen wir:

- Umformungsalgorithmen anwenden können.

Was wollen wir heute erreichen?

Zu DEAs, regulären Ausdrücken, NEAs. Grammatiken wollen wir:

- Umformungsalgorithmen anwenden können.
- Aufgaben lösen.

Was wollen wir heute erreichen?

Zu DEAs, regulären Ausdrücken, NEAs. Grammatiken wollen wir:

- Umformungsalgorithmen anwenden können.
- Aufgaben lösen.
- Zusammenhänge wiederholt verstehen.

Was wollen wir heute erreichen?

Zu DEAs, regulären Ausdrücken, NEAs. Grammatiken wollen wir:

- Umformungsalgorithmen anwenden können.
- Aufgaben lösen.
- Zusammenhänge wiederholt verstehen.
- Minimierungsalgorithmus anwenden.

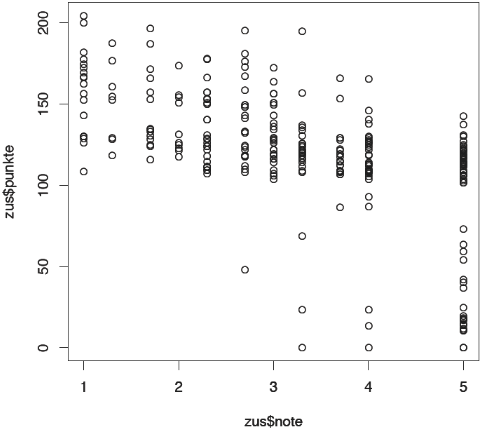
Was wollen wir heute erreichen?

Zu DEAs, regulären Ausdrücken, NEAs. Grammatiken wollen wir:

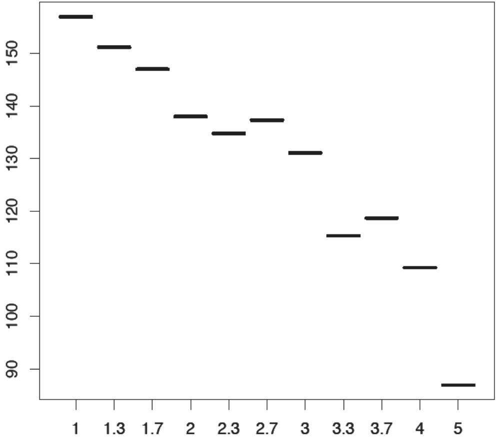
- Umformungsalgorithmen anwenden können.
- Aufgaben lösen.
- Zusammenhänge wiederholt verstehen.
- Minimierungsalgorithmus anwenden.
- Pumping-Lemma verstehen und anwenden.

Warum Übungsblätter?

Warum Übungsblätter?

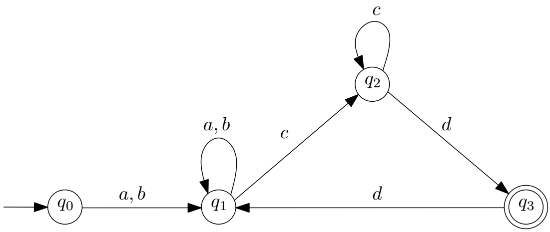


Deshalb!



Aufgabe 1a

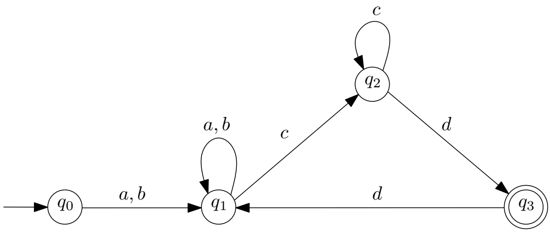
Gegeben sei der folgende endliche Akzeptor $\mathcal{M}$ mit dem Eingabealphabet $\mathcal{A} = \{a, b, c, d\}$:



i) Welche Sprache $\mathcal{L}(\mathcal{M})$ wird von dem Akzeptor $\mathcal{M}$ akzeptiert?

Aufgabe 1a

Gegeben sei der folgende endliche Akzeptor $\mathcal{M}$ mit dem Eingabealphabet $\mathcal{A} = \{a, b, c, d\}$:



i) Welche Sprache $\mathcal{L}(\mathcal{M})$ wird von dem Akzeptor $\mathcal{M}$ akzeptiert?
Lösung:
$L(\gamma)$ mit $\gamma = (a+b)^+c^+d(d(a+b)^*c^+d)^*$

Aus endlichem Automaten eine Typ 3 Grammatik konstruieren:

Aus endlichem Automaten eine Typ 3 Grammatik konstruieren:

Sei A_L ein (deterministischer) endlicher Automat, der die Sprache $L \subseteq \Sigma^*$ akzeptiert, $A_L = (Q, \Sigma, \delta, q_0, F)$. Dann kann man die zugehörige Grammatik G_L wie folgt konstruieren:

$V := Q$

$S := q_0$

R enthält die Regel $q \longrightarrow \varepsilon$ für alle $q \in F$ und die Regel $q \longrightarrow aq'$, falls $\delta(q, a) = q'$

Für $w = w_1 ... w_n \in L$ durchläuft A_L genau die Zustände $q_0, q_1, ..., q_n \in Q$ mit $q_n \in F$. Dann gilt:
$q_0 \longrightarrow w_1 q_1 \longrightarrow w_1 w_2 q_2 \longrightarrow ... \longrightarrow w q_n \longrightarrow w$

Aufgabe 1a

ii) Konstruieren Sie aus $\mathcal{M}$ eine rechtslineare Grammatik G, die $\mathcal{L}(\mathcal{M})$ erzeugt.

Aufgabe 1a

ii) Konstruieren Sie aus $\mathcal{M}$ eine rechtslineare Grammatik G, die $\mathcal{L}(\mathcal{M})$ erzeugt.

Lösung:

$G = (V, \Sigma, S, R)$ mit $V = \{Q0, Q1, Q2.Q3\}$

$S = Q0$

$\Sigma = \{a, b, c, d\}$

$R = \{$

$Q0 \rightarrow aQ1|bQ1$

$Q1 \rightarrow aQ1|bQ1|cQ2$

$Q2 \rightarrow cQ2|dQ3$

$Q3 \rightarrow dQ1|\varepsilon\}$

Aufgabe 1b

Die Sprache $\mathcal{L}$ sei durch den regulären Ausdruck $(aa^*b^*)^*cc^*$ definiert.

i) Geben Sie eine rechtslineare Grammatik $\mathcal{G}$ an, die $\mathcal{L}$ erzeugt.

Aufgabe 1b

Die Sprache $\mathcal{L}$ sei durch den regulären Ausdruck $(aa^*b^*)^*cc^*$ definiert.

i) Geben Sie eine rechtslineare Grammatik $\mathcal{G}$ an, die $\mathcal{L}$ erzeugt.

Lösung:

$\mathcal{L}$ wird von folgender Grammatik $\mathcal{G}$ (in Kurznotation) mit Terminalalphabet $\{a, b, c\}$ und Variablenalphabet $\mathcal{V} = \{S, A, B\}$ erzeugt:

$S \rightarrow aA|cB$

$A \rightarrow aA|bA|cB$

$B \rightarrow cB|\lambda$

Aus Typ 3 Grammatik einen nichtdeterministischen endlichen Automaten konstruieren.

Aus Typ 3 Grammatik einen nichtdeterministischen endlichen Automaten konstruieren.

Sei eine Chomsky-3-Grammatik G_L gegeben. Wir entwerfen einen NEA $A_L := (Q, \Sigma, \delta, q_0, F)$, der L akzeptiert. Und zwar auf folgende Art:

$$\begin{aligned}
Q &:= V \\
q_0 &:= S \\
F &:= \{A \in V | (A \to \varepsilon) \in R\} \\
\delta(A, a) &:= \{B | (A \to aB) \in R\}
\end{aligned}$$

Für $w = w_1 ... w_n \in L$ hat die Ableitung von w mittels G_L das Aussehen:

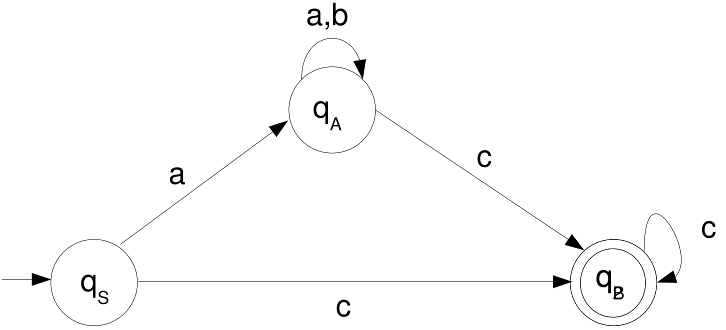
$S \to w_1 A_1 \to w_1 w_2 A_2 \to ... \to w A_n \to w$

Aufgabe 1b

ii) Konstruieren Sie aus $\mathcal{G}$ einen endlichen Akzeptor, der $\mathcal{L}$ akzeptiert.

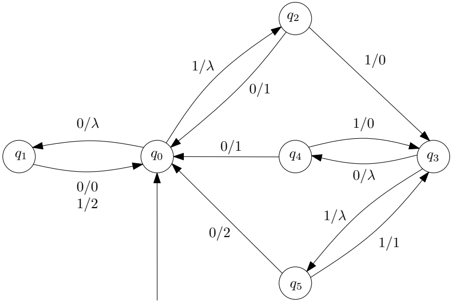
Aufgabe 1b

ii) Konstruieren Sie aus $\mathcal{G}$ einen endlichen Akzeptor, der $\mathcal{L}$ akzeptiert. Lösung:



Aufgabe 2a

Gegeben der Automat $T = (Q, \mathcal{A}, \Omega, \delta, \rho, q_0)$ mit den Zuständen $Q = \{q_0, \ldots, q_5\}$, wobei das Eingabealphabet durch $\mathcal{A} = \{0, 1\}$ und das Ausgabealphabet durch $\Omega = \{0, 1, 2\}$ gegeben sei. Der durch δ und ρ gegebene Zustandsübergangsgraph für den Automaten T sei



Aufgabe 2a

i) Berechnen sie die Ausgabe dieses Automaten für die Eingabe $w = 01011100 \in \mathcal{A}^*$. Lesen sie dabei die Zeichen von links nach rechts. Geben sie in ihrer Lösung auch die Zustandsfolge an, die der Automat durchläuft.

Aufgabe 2a

i) Berechnen sie die Ausgabe dieses Automaten für die Eingabe $w = 01011100 \in \mathcal{A}^*$. Lesen sie dabei die Zeichen von links nach rechts. Geben sie in ihrer Lösung auch die Zustandsfolge an, die der Automat durchläuft.

Lösung:

w	0	1	0	1	1	1	0	0
	λ	2	λ	2	λ	0	λ	1
	q_1	q_0	q_1	q_0	q_2	q_3	q_4	q_0

ii) Von welchem Typ ist dieser Automat (Mealy oder Moore Typ)?

Aufgabe 2a

i) Berechnen sie die Ausgabe dieses Automaten für die Eingabe $w = 01011100 \in \mathcal{A}^*$. Lesen sie dabei die Zeichen von links nach rechts. Geben sie in ihrer Lösung auch die Zustandsfolge an, die der Automat durchläuft.

Lösung:

w	0	1	0	1	1	1	0	0
	λ	2	λ	2	λ	0	λ	1
	q_1	q_0	q_1	q_0	q_2	q_3	q_4	q_0

ii) Von welchem Typ ist dieser Automat (Mealy oder Moore Typ)?

Lösung:

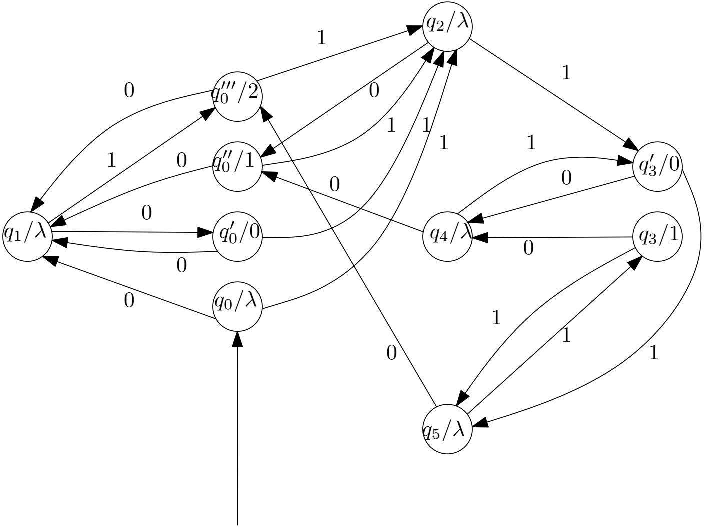
Dies ist ein Mealy Automat.

Aufgabe 2a

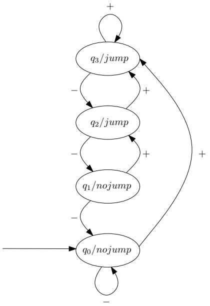
iii) Überführen sie die Darstellung dieses Automaten in die des anderen Typs (Mealy oder Moore).

Aufgabe 2a

iii) Überführen sie die Darstellung dieses Automaten in die des anderen Typs (Mealy oder Moore).
Lösung:
Zur Umwandlung in einen Moore-Automaten zählt man einfach wieviele mit verschiedenen Ausgabesymbolen beschrifteten Kanten in einen Zustand eingehen und vervielfacht den Zustand dementsprechend. Dann weist man den entsprechenden Zuständen die Ausgabezeichen zu. Zuletzt fügt man den neuen Zuständen wieder alle ausgehenden Kanten des Ursprungszustandes hinzu.

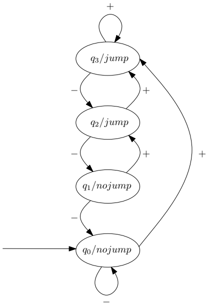

q_2/λ
$q_0'''/2$
$q_0''/1$
$q_0'/0$
q_0/λ
q_1/λ
$q_3'/0$
q_4/λ
$q_3/1$
q_5/λ
0
1

Aufgabe 2b



i) Von welchem Typ ist dieser Automat (Mealy oder Moore Typ)?

Aufgabe 2b



i) Von welchem Typ ist dieser Automat (Mealy oder Moore Typ)?
Lösung:
Dies ist ein Moore Automat.

Aufgabe 2b

ii) Berechnen sie die Ausgabe dieses Automaten für die Eingabe $w = - + - - - + - + - + - \in \mathcal{A}^*$. Geben sie in ihrer Lösung auch die Zustandsfolge an, die der Automat durchläuft. Wie oft hat sich der Automat mit seiner Vorhersage geirrt?

Aufgabe 2b

ii) Berechnen sie die Ausgabe dieses Automaten für die Eingabe $w = - + - - - + - + - + - \in \mathcal{A}^*$. Geben sie in ihrer Lösung auch die Zustandsfolge an, die der Automat durchläuft. Wie oft hat sich der Automat mit seiner Vorhersage geirrt?

Lösung:

w		$-$	$+$	$-$	$-$	$-$	$+$	$-$	$+$	$-$	$+$	$-$
	nj	nj	j	j	nj	nj	j	j	j	j	j	j
	q_0	q_0	q_3	q_2	q_1	q_0	q_3	q_2	q_3	q_2	q_3	q_2

Der Automat hat sich 7 mal mit seiner Vorhersage geirrt.

Wiederholung

Definition: ε-Übergänge beim NEA

Wiederholung

Definition: ε-Übergänge beim NEA

Für einen Zustand $q \in Q$ ist der ε-Abschluss $E(q)$ wie folgt definiert:
$E(q) := \{p \in Q | p$ ist von q durch eine Folge von ε-Übergängen erreichbar.$\}$
Beachte:
$E(q) \subseteq Q$, $E(q) \in 2^Q$, $q \in E(q)$

Potenzmengenkonstruktion

Gegeben sei ein NEA $A := (Q, \Sigma, \delta, s, F)$. Wir konstruieren daraus einen DEA $\tilde{A} = (\tilde{Q}, \Sigma, \tilde{\delta}, \tilde{s}, \tilde{F})$

Potenzmengenkonstruktion

Gegeben sei ein NEA $A := (Q, \Sigma, \delta, s, F)$. Wir konstruieren daraus einen DEA $\tilde{A} = (\tilde{Q}, \Sigma, \tilde{\delta}, \tilde{s}, \tilde{F})$

- $\tilde{Q} = 2^Q$, d.h. die Zustände des DEA sind Mengen von Zuständen des NEA.

Potenzmengenkonstruktion

Gegeben sei ein NEA $A := (Q, \Sigma, \delta, s, F)$. Wir konstruieren daraus einen DEA $\tilde{A} = (\tilde{Q}, \Sigma, \tilde{\delta}, \tilde{s}, \tilde{F})$

- $\tilde{Q} = 2^Q$, d.h. die Zustände des DEA sind Mengen von Zuständen des NEA.
- $\tilde{\delta} : \tilde{Q} \times \Sigma \rightarrow \tilde{Q}$ # interessante Teil

Potenzmengenkonstruktion

Gegeben sei ein NEA $A := (Q, \Sigma, \delta, s, F)$. Wir konstruieren daraus einen DEA $\tilde{A} = (\tilde{Q}, \Sigma, \tilde{\delta}, \tilde{s}, \tilde{F})$

- $\tilde{Q} = 2^Q$, d.h. die Zustände des DEA sind Mengen von Zuständen des NEA.
- $\tilde{\delta} : \tilde{Q} \times \Sigma \to \tilde{Q}$ # interessante Teil
- $\tilde{s} := E(s)$

Potenzmengenkonstruktion

Gegeben sei ein NEA $A := (Q, \Sigma, \delta, s, F)$. Wir konstruieren daraus einen DEA $\tilde{A} = (\tilde{Q}, \Sigma, \tilde{\delta}, \tilde{s}, \tilde{F})$

- $\tilde{Q} = 2^Q$, d.h. die Zustände des DEA sind Mengen von Zuständen des NEA.
- $\tilde{\delta} : \tilde{Q} \times \Sigma \rightarrow \tilde{Q}$ # interessante Teil
- $\tilde{s} := E(s)$
- $\tilde{F} := \{\tilde{q} \in \tilde{Q} | \tilde{q} \cap F \neq \emptyset\}$

Aufgabe 1

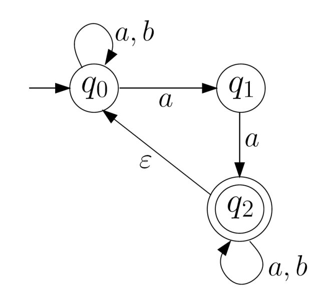
Finde mit Potenzmengenkonstruktion einen zum NEA $M1$ äquivalenten DEA.

Aufgabe 2

Finde mit Potenzmengenkonstruktion einen zum NEA $M2$ äquivalenten DEA.

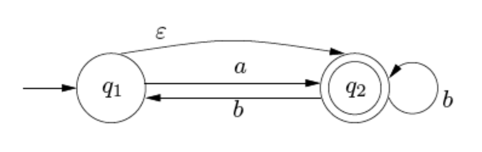
Aufgabe 4

Finde mit Potenzmengenkonstruktion einen zu folgendem NEA äquivalenten DEA.



Aufgabe 5

Finde mit Potenzmengenkonstruktion einen zu folgendem NEA äquivalenten DEA.

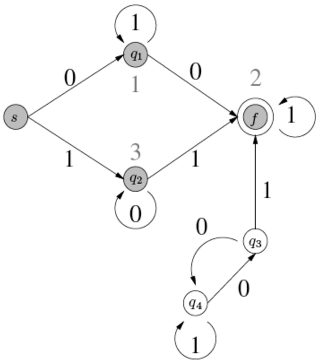


Definition: überflüssige Zustände

Zustände eines (deterministischen) endlichen Automaten, die vom Anfangszustand aus nicht erreichbar sind, heissen überflüssig.

Definition: überflüssige Zustände

Zustände eines (deterministischen) endlichen Automaten, die vom Anfangszustand aus nicht erreichbar sind, heissen überflüssig.



Definition:

Zwei Zustände p und q eines deterministischen endlichen Automaten heissen äquivalent $(p \equiv q)$, wenn für alle Wörter $w \in \Sigma^*$ gilt:

Definition:

Zwei Zustände p und q eines deterministischen endlichen Automaten heissen äquivalent $(p \equiv q)$, wenn für alle Wörter $w \in \Sigma^*$ gilt:

$$\delta(p, w) \in F \Leftrightarrow \delta(q, w) \in F.$$

Definition:

Zwei Zustände p und q eines deterministischen endlichen Automaten heissen äquivalent $(p \equiv q)$, wenn für alle Wörter $w \in \Sigma^*$ gilt:

$$\delta(p, w) \in F \Leftrightarrow \delta(q, w) \in F.$$

Offensichtlich ist $\equiv$ eine Äquivalenzrelation. Mit $[p]$ bezeichnen wir die Äquivalenzklasse der zu p äquivalenten Zustände.

Defiition: Äquivalenzklassenautomat

Zu einem DEA $A = (Q, \Sigma, \delta, s, F)$ definieren wir den Äquivalenzklassenautomaten $A^{\equiv} = (Q^{\equiv}, \Sigma^{\equiv}, \delta^{\equiv}, s^{\equiv}, F^{\equiv})$ durch:

Defiition: Äquivalenzklassenautomat

Zu einem DEA $A = (Q, \Sigma, \delta, s, F)$ definieren wir den Äquivalenzklassenautomaten $A^{\equiv} = (Q^{\equiv}, \Sigma^{\equiv}, \delta^{\equiv}, s^{\equiv}, F^{\equiv})$ durch:

$$Q^{\equiv} := \{[q] | q \in Q\}$$

Defiition: Äquivalenzklassenautomat

Zu einem DEA $A = (Q, \Sigma, \delta, s, F)$ definieren wir den Äquivalenzklassenautomaten $A^{\equiv} = (Q^{\equiv}, \Sigma^{\equiv}, \delta^{\equiv}, s^{\equiv}, F^{\equiv})$ durch:

$$Q^{\equiv} := \{[q] | q \in Q\}$$
$$\Sigma^{\equiv} := \Sigma$$

Defiition: Äquivalenzklassenautomat

Zu einem DEA $A = (Q, \Sigma, \delta, s, F)$ definieren wir den Äquivalenzklassenautomaten $A^{\equiv} = (Q^{\equiv}, \Sigma^{\equiv}, \delta^{\equiv}, s^{\equiv}, F^{\equiv})$ durch:

$$Q^{\equiv} := \{[q] | q \in Q\}$$

$$\Sigma^{\equiv} := \Sigma$$

$$\delta^{\equiv}([q], a) := [\delta(q, a)]$$

Defiition: Äquivalenzklassenautomat

Zu einem DEA $A = (Q, \Sigma, \delta, s, F)$ definieren wir den Äquivalenzklassenautomaten $A^{\equiv} = (Q^{\equiv}, \Sigma^{\equiv}, \delta^{\equiv}, s^{\equiv}, F^{\equiv})$ durch:

$$Q^{\equiv} := \{[q] | q \in Q\}$$
$$\Sigma^{\equiv} := \Sigma$$
$$\delta^{\equiv}([q], a) := [\delta(q, a)]$$
$$s^{\equiv} := [s]$$

Defiition: Äquivalenzklassenautomat

Zu einem DEA $A = (Q, \Sigma, \delta, s, F)$ definieren wir den Äquivalenzklassenautomaten $A^{\equiv} = (Q^{\equiv}, \Sigma^{\equiv}, \delta^{\equiv}, s^{\equiv}, F^{\equiv})$ durch:

$$
\begin{aligned}
Q^{\equiv} &:= \{[q] | q \in Q\} \\
\Sigma^{\equiv} &:= \Sigma \\
\delta^{\equiv}([q], a) &:= [\delta(q, a)] \\
s^{\equiv} &:= [s] \\
F^{\equiv} &:= \{[f] | f \in F\}
\end{aligned}
$$

Satz

Der Äquivalenzklassenautomat $A^{\equiv}$ zu A akzeptiert dieselbe Sprache wie A.

Satz

Der Äquivalenzklassenautomat $A^{\equiv}$ zu A akzeptiert dieselbe Sprache wie A.

Satz

Der Äquivalenzklassenautomat $A^{\equiv}$ zu einem DEA A ohne überflüssige Zustände ist minimal.

Minimierungsalgorithmus:

1. Entferne alle überflüssigen Zustände.

Minimierungsalgorithmus:

1. Entferne alle überflüssigen Zustände.
2. Stelle eine Tabelle aller Zustandspaare $\{p, q\}$ mit $p \neq q$ auf.

Minimierungsalgorithmus:

1. Entferne alle überflüssigen Zustände.
2. Stelle eine Tabelle aller Zustandspaare $\{p, q\}$ mit $p \neq q$ auf.
3. Markiere alle Paare $\{p, q\}$ mit entweder $p \in F$ oder $q \in F$.

Minimierungsalgorithmus:

1. Entferne alle überflüssigen Zustände.
2. Stelle eine Tabelle aller Zustandspaare $\{p, q\}$ mit $p \neq q$ auf.
3. Markiere alle Paare $\{p, q\}$ mit entweder $p \in F$ oder $q \in F$.
4. Für jedes noch unmarkierte Paar $\{p, q\}$ und jedes $a \in \Sigma$ teste, ob $\{\delta(p, a), \delta(q, a)\}$ bereits markiert ist. Wenn ja, dann markiere auch $\{p, q\}$.

Minimierungsalgorithmus:

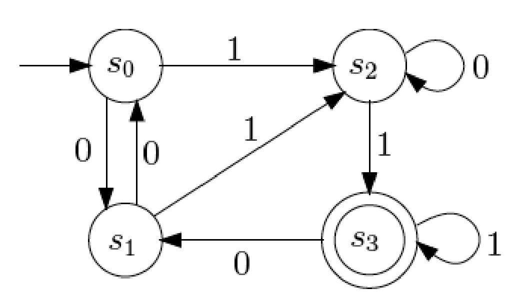
1. Entferne alle überflüssigen Zustände.
2. Stelle eine Tabelle aller Zustandspaare $\{p, q\}$ mit $p \neq q$ auf.
3. Markiere alle Paare $\{p, q\}$ mit entweder $p \in F$ oder $q \in F$.
4. Für jedes noch unmarkierte Paar $\{p, q\}$ und jedes $a \in \Sigma$ teste, ob $\{\delta(p, a), \delta(q, a)\}$ bereits markiert ist. Wenn ja, dann markiere auch $\{p, q\}$.
5. Wiederhole Schritt 4 bis sich keine Änderung in der Tabelle mehr ergibt.

Minimierungsalgorithmus:

1. Entferne alle überflüssigen Zustände.
2. Stelle eine Tabelle aller Zustandspaare $\{p, q\}$ mit $p \neq q$ auf.
3. Markiere alle Paare $\{p, q\}$ mit entweder $p \in F$ oder $q \in F$.
4. Für jedes noch unmarkierte Paar $\{p, q\}$ und jedes $a \in \Sigma$ teste, ob $\{\delta(p, a), \delta(q, a)\}$ bereits markiert ist. Wenn ja, dann markiere auch $\{p, q\}$.
5. Wiederhole Schritt 4 bis sich keine Änderung in der Tabelle mehr ergibt.
6. Verschmelze alle jetzt noch unmarkierten Paare zu jeweils einem Zustand.

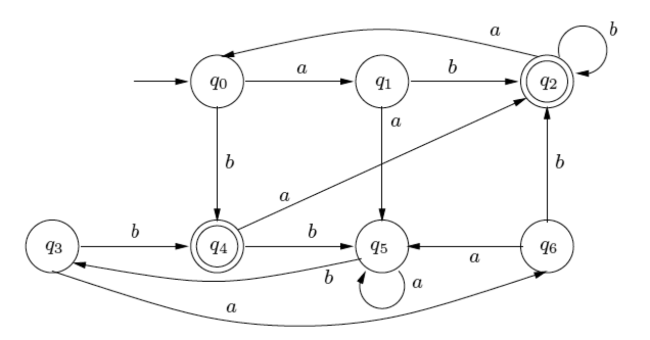
Aufgabe 1

Minimiere den folgenden Automaten mittels Konstruktion des Äquivalenzklassenautomatens. Gib den Minimalautomaten graphisch an.



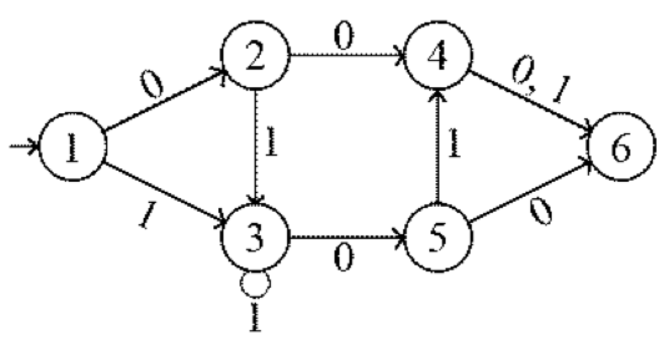
Aufgabe 2

Minimiere den folgenden Automaten mittels Konstruktion des Äquivalenzklassenautomatens. Gib den Minimalautomaten graphisch an.



Aufgabe 3

Minimiere den folgenden Automaten mittels Konstruktion des Äquivalenzklassenautomatens. Gib den Minimalautomaten graphisch an. $F = 6$



Satz: Pumping-Lemma

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, sodass für jedes Wort $w \in L$ mit $|w| > n$ eine Darstellung

$$w = uvx \text{ mit } |uv| \leq n, v \neq \epsilon,$$

existiert, bei der auch $uv^ix \in L$ ist für alle $i \in \mathbb{N}_0$.

Satz: Pumping-Lemma

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, sodass für jedes Wort $w \in L$ mit $|w| > n$ eine Darstellung

$$w = uvx \text{ mit } |uv| \leq n, v \neq \epsilon,$$

existiert, bei der auch $uv^ix \in L$ ist für alle $i \in \mathbb{N}_0$.

Hinweis:
Man verwendet das Pumping-Lemma um zu zeigen, dass eine Sprache nicht regulär ist.

Beispiel:
$L = \{a^n b^n | n > 0, n \in \mathbb{N}\}$

Um zu zeigen, dass L nicht regulär ist, verwenden wir das Pumping-Lemma.

Beispiel:
$L = \{a^n b^n | n > 0, n \in \mathbb{N}\}$

Um zu zeigen, dass L nicht regulär ist, verwenden wir das Pumping-Lemma.

Angenommen L sei regulär. Dann $\exists n$, sodass sich alle Wörter $w \in L$ der Länge $\geq n$ wie im Pumping Lemma beschrieben zerlegen lassen. Betrachten wir speziell das Wort $a^n b^n$ der Länge $2n$. Die entsprechende Zerlegung uvx dieses Wortes ist aufgrund $v \neq \epsilon$ so, dass v nicht leer ist, und aufgrund $|uv| \leq n$ kann v nur aus a's bestehen. Aufgrund von $uv^i x \in L$ wäre dann das Wort $ux = a^{n-|v|} b^n$ in der Sprache, was im Widerspruch zur Definition von L steht. Deshalb ist L nicht regulär.

Aufgabe 4

$L = \{0^m | m$ ist Quadratzahl$\}$ Zeige, dass L nicht regulär ist.

Aufgabe 4

$L = \{0^m | m \text{ ist Quadratzahl}\}$ Zeige, dass L nicht regulär ist.

Aufgabe 5

$L = \{0^p | p \text{ ist Primzahl}\}$ Zeige, dass L nicht regulär ist.

Aufgabe 6

$L = \{a^n b a^m b a^{n+m} | n, m \geq 1\}$ Zeige, dass L nicht regulär ist.

Aufgabe 6

$L = \{a^n b a^m b a^{n+m} | n, m \geq 1\}$ Zeige, dass L nicht regulär ist.

Aufgabe 7

Zeige für die Sprache $L = \{01^k 0^l 1 | k, l \geq 0\}$ explizit, dass die im Pumping-Lemma formulierte notwendige Bedingung für Regularität erfüllt ist. (Gib also eine konkrete Belegung für die existenzquantifizierten Ausdrücke an und begründe.)

Aufgabe 8

$L = \emptyset$

Ist für L das Pumping-Lemma anwendbar?

Aufgabe 8

$L = \emptyset$

Ist für L das Pumping-Lemma anwendbar?

Aufgabe 9

$L = \{00, 11\}$

Ist für L das Pumping-Lemma anwendbar?

Reflexion

Was haben wir heute gelernt?

Reflexion

Was haben wir heute gelernt?

- Übungsblatt immer, sauber und formal (einigermassen) korrekt bearbeiten.

Reflexion

Was haben wir heute gelernt?

- Übungsblatt immer, sauber und formal (einigermassen) korrekt bearbeiten.
- NEA $\rightarrow$ DEA (Potenzmengenkonstruktion)

Reflexion

Was haben wir heute gelernt?

- Übungsblatt immer, sauber und formal (einigermassen) korrekt bearbeiten.
- NEA $\rightarrow$ DEA (Potenzmengenkonstruktion)
- Automatenminimierung

Reflexion

Was haben wir heute gelernt?

- Übungsblatt immer, sauber und formal (einigermassen) korrekt bearbeiten.
- NEA $\longrightarrow$ DEA (Potenzmengenkonstruktion)
- Automatenminimierung
- Pumping-Lemma

Noch Fragen?

Vorschau

Vorschau

- unbekannt

Bis zum nächsten Mal