Theoretische Grundlagen der Informatik
Tutorium XII & XIII (AVG, SR -109)
Tut Nr. 2 – Üb1, NEA $\rightarrow$ DEA, Minimierung, PL

David Münch

Karlsruher Institut für Technologie
Institut für Informatik
IKS Müller-Quade

5. November 2009

Inhaltsverzeichnis

Inhaltsverzeichnis

Inhaltsverzeichnis

Inhaltsverzeichnis

Organisatorisches

Email: muenchdavid@gmail.com

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

Tutorium 12: Donnerstags 8:00 Uhr - AVG, Raum -109
Tutorium 13: Donnerstags 9:45 Uhr - AVG, Raum -109

Übungsblattabgabe Mittwochs 12:00 Uhr.

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.

Raumänderung

Ab Donnerstag 12.11.2009 findet dieses Tutorium immer in SR 301 in Gebäude 50.34 statt.

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.

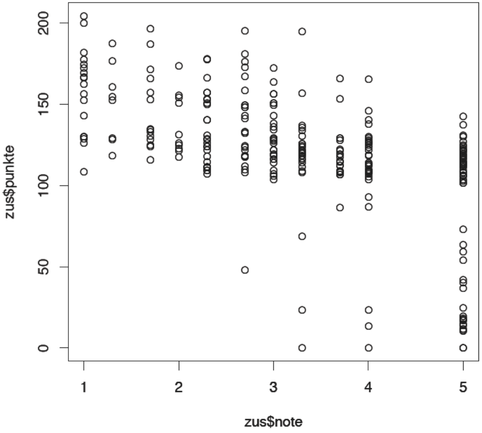
Was wollen wir heute erreichen?

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

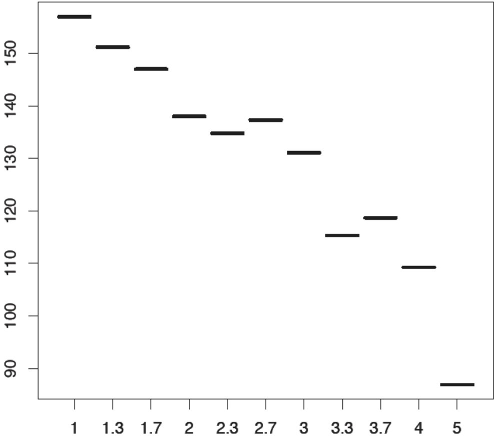
- Umformungsalgorithmen anwenden können.
- Aufgaben lösen.
- Zusammenhänge wiederholt verstehen.
- Minimierungsalgorithmus anwenden.
- Pumping-Lemma verstehen und anwenden.

Warum Übungsblätter?

Warum Übungsblätter?

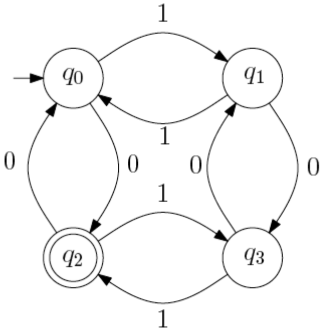


Deshalb!



Aufgabe 1

Gegeben sei der folgende deterministische endliche Automat $\mathcal{M} = (\mathcal{Q}, \Sigma, \delta, q_0, \mathcal{F})$ mit $\mathcal{Q} = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{0, 1\}$ und $\mathcal{F} = \{q_2\}$. Die Zustandsübergangsfunktion δ sei gegeben durch folgenden Graphen:



a) Simulieren Sie die Berechnung von $\mathcal{M}$ bei Eingabe $w_1 = 000011$. Akzeptiert $\mathcal{M}$ das Wort w_1? (1P)

a) Simulieren Sie die Berechnung von $\mathcal{M}$ bei Eingabe $w_1 = 000011$. Akzeptiert $\mathcal{M}$ das Wort w_1? (1P)

Lösung:

$$(q_0)0|(q_2)0|(q_0)0|(q_2)0|(q_0)1|(q_1)1|(q_0)$$

$\mathcal{M}$ terminiert nach Lesen von w_1 in $q_0 \notin \mathcal{F}$. Deshalb akzeptiert $\mathcal{M}$ das Wort w_1 nicht.

a) Simulieren Sie die Berechnung von $\mathcal{M}$ bei Eingabe $w_1 = 000011$. Akzeptiert $\mathcal{M}$ das Wort w_1? (1P)

Lösung:

$$(q_0)0|(q_2)0|(q_0)0|(q_2)0|(q_0)1|(q_1)1|(q_0)$$

$\mathcal{M}$ terminiert nach Lesen von w_1 in $q_0 \notin \mathcal{F}$. Deshalb akzeptiert $\mathcal{M}$ das Wort w_1 nicht.

b) Simulieren Sie die Berechnung von $\mathcal{M}$ bei Eingabe $w_2 = 0101011$. Akzeptiert $\mathcal{M}$ das Wort w_2? (1P)

a) Simulieren Sie die Berechnung von $\mathcal{M}$ bei Eingabe $w_1 = 000011$. Akzeptiert $\mathcal{M}$ das Wort w_1? (1P)
Lösung:

$$(q_0)0|(q_2)0|(q_0)0|(q_2)0|(q_0)1|(q_1)1|(q_0)$$

$\mathcal{M}$ terminiert nach Lesen von w_1 in $q_0 \notin \mathcal{F}$. Deshalb akzeptiert $\mathcal{M}$ das Wort w_1 nicht.

b) Simulieren Sie die Berechnung von $\mathcal{M}$ bei Eingabe $w_2 = 0101011$. Akzeptiert $\mathcal{M}$ das Wort w_2? (1P)
Lösung:

$$(q_0)0|(q_2)1|(q_3)0|(q_1)1|(q_0)0|(q_2)1|(q_3)1|(q_2)$$

$\mathcal{M}$ terminiert nach Lesen von w_1 in $q_2 \in \mathcal{F}$. Deshalb akzeptiert $\mathcal{M}$ das Wort w_1.

c) Welche Sprache $\mathcal{L}$ erkennt $\mathcal{M}$? (1P)

c) Welche Sprache $\mathcal{L}$ erkennt $\mathcal{M}$? (1P)

Lösung:

$\mathcal{M}$ akzeptiert ein Wort w genau dann wenn w eine ungerade Zahl Vorkommen von 0 und eine gerade Zahl Vorkommen von 1 hat, also $\mathcal{L}(\mathcal{M}) = \{w \in \Sigma^* | \ 2 \text{ teilt } \#_1 w \text{ und } 2 \text{ teilt nicht } \#_0 w\}$.

Aus endlichem Automaten eine Typ 3 Grammatik konstruieren:

Aus endlichem Automaten eine Typ 3 Grammatik konstruieren:

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

$V := Q$

$S := q_0$

P 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 M_L genau die Zustände $q_0, q_1, ..., q_n \in Q$ mit $q_n \in F$. Dann gilt:
$q_0 \longrightarrow w_1 q_1 \longrightarrow w_1 w_2 q_2 \longrightarrow ... \longrightarrow w q_n \longrightarrow w$

d) Geben Sie eine rechtslineare Grammatik $\mathcal{G}$ an, die $\mathcal{L}$ erzeugt. (1P)

d) Geben Sie eine rechtslineare Grammatik $\mathcal{G}$ an, die $\mathcal{L}$ erzeugt. (1P)

Lösung:

Sei $\mathcal{G} = (\{0,1\}, \{S,A,B,C\}, S, \mathcal{P})$ mit den Produktionen

$$\begin{aligned} \mathcal{P} = \{\ S &\rightarrow 1A|0B \\ A &\rightarrow 1S|0C \\ B &\rightarrow 0S|1C|\varepsilon \\ C &\rightarrow 1B|0A\}. \end{aligned}$$

Aufgabe 2

Gegeben sei die Grammatik $\mathcal{G} = (\mathcal{T}, \mathcal{V}, S, \mathcal{P})$ mit $\mathcal{T} = \{a, b, c\}$, $\mathcal{V} = \{S, A, B, C\}$ und den Produktionen $\mathcal{P}$ gegeben durch

$$\begin{aligned} \mathcal{P} : S &\longrightarrow aS|cS|aA|cB \\ A &\longrightarrow bC|b \\ B &\longrightarrow aC|a \\ C &\longrightarrow bC|b \end{aligned}$$

a) Geben Sie einen regulären Ausdruckt für die von $\mathcal{G}$ erzeugte Sprache $\mathcal{L}$ an. (2P)

Aufgabe 2

Gegeben sei die Grammatik $\mathcal{G} = (\mathcal{T}, \mathcal{V}, S, \mathcal{P})$ mit $\mathcal{T} = \{a, b, c\}$, $\mathcal{V} = \{S, A, B, C\}$ und den Produktionen $\mathcal{P}$ gegeben durch

$$\begin{aligned} \mathcal{P} : S &\longrightarrow aS|cS|aA|cB \\ A &\longrightarrow bC|b \\ B &\longrightarrow aC|a \\ C &\longrightarrow bC|b \end{aligned}$$

a) Geben Sie einen regulären Ausdruckt für die von $\mathcal{G}$ erzeugte Sprache $\mathcal{L}$ an. (2P)

Lösung:

$\mathcal{L}(\mathcal{G}) = (a + c)^*(ab + ca)b^*$

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 $M_L := (Q, \Sigma, \delta, q_0, F)$, der L akzeptiert. Und zwar auf folgende Art:

$$Q := V$$
$$q_0 := S$$
$$F := \{A \in V | (A \to \varepsilon) \in P\}$$
$$\delta(A, a) := \{B | (A \to aB) \in P\}$$

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

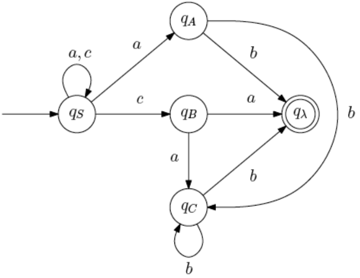
$S \to w_1 A_1 \to w_1 w_2 A_2 \to ... \to w A_n \to w$

b) Geben sie einen endlichen Akzeptor $\mathcal{M}$ an der $\mathcal{L}$ erkennt. (2P)

b) Geben sie einen endlichen Akzeptor $\mathcal{M}$ an der $\mathcal{L}$ erkennt. (2P)

Lösung:

Wir wenden die Konstruktion aus der Vorlesung an und konstruieren einen nichtdeterministischen endlichen Akzeptor. Wir führen die Zustände $\mathcal{Q} = \{q_\lambda, q_A, q_B, q_C, q_S\}$ ein. Startzustand ist q_S, Endzustandsmenge ist $\mathcal{F} = \{q_\lambda\}$. Wir geben die Zustandsübegangsrelation δ durch folgenden Graphen an.



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:
$E(q) \subseteq Q$, $E(q) \in 2^Q$, $q \in E(q)$

Hinweis: Alternativ auch λ statt ε möglich.

Potenzmengenkonstruktion

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

Potenzmengenkonstruktion

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

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

Potenzmengenkonstruktion

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

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

Potenzmengenkonstruktion

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

- $\tilde{Q} = 2^Q$, 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 $M := (Q, \Sigma, \delta, s, F)$. Wir konstruieren daraus einen DEA $\tilde{M} = (\tilde{Q}, \Sigma, \tilde{\delta}, \tilde{s}, \tilde{F})$

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

Aufgabe

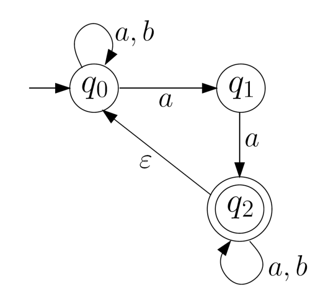
Finde mit Potenzmengenkonstruktion einen zum NEA $M1$ äquivalenten DEA.

Aufgabe

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

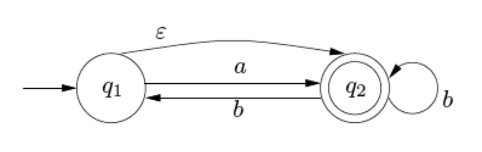
Aufgabe

Finde mit Potenzmengenkonstruktion einen zu folgendem NEA äquivalenten DEA.



Aufgabe

Finde mit Potenzmengenkonstruktion einen zu folgendem NEA äquivalenten DEA.

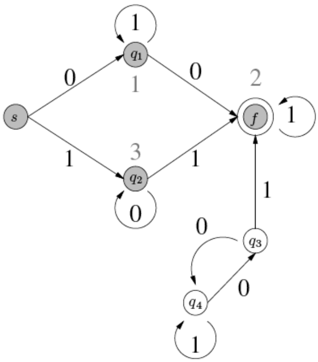


Definition: überflüssige Zustände

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

Defiition: Äquivalenzklassenautomat

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

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

Defiition: Äquivalenzklassenautomat

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

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

Defiition: Äquivalenzklassenautomat

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

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

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

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

Defiition: Äquivalenzklassenautomat

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

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

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:

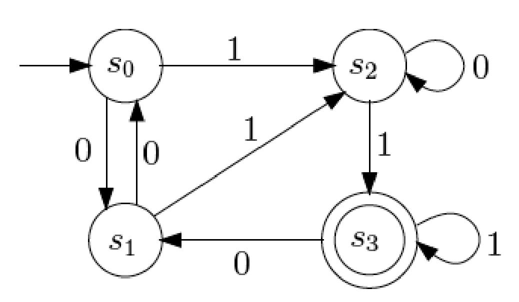
1. Entferne alle überflüssigen Zustände.
2. Stelle eine Tabelle aller Zustandspaare $\{p, q\}$ mit $p \neq q$ auf.
3. Markiere alle Paare $\{p, q\}$ mit entweder $p \in F$ **oder** $q \in F$.
4. Für jedes noch unmarkierte Paar $\{p, q\}$ und jedes $a \in \Sigma$ teste, ob $\{\delta(p, a), \delta(q, a)\}$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

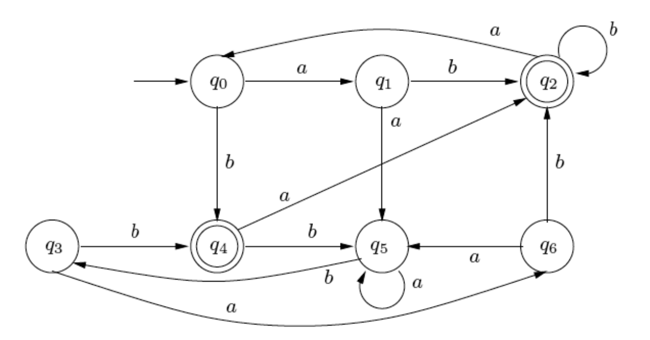
Aufgabe 1

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



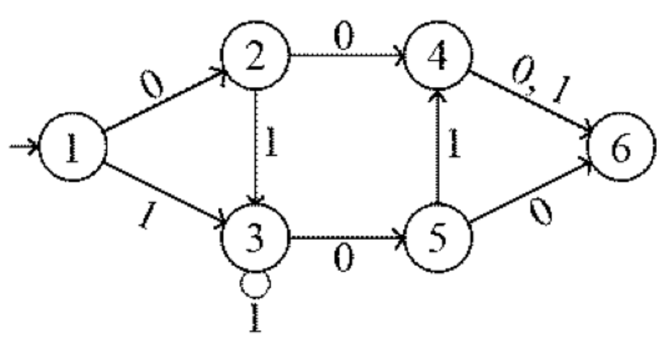
Aufgabe 2

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

- Pumping-Lemma

Vorschau

- Pumping-Lemma
- Kontextfreie Sprachen

Vorschau

- Pumping-Lemma
- Kontextfreie Sprachen
- Raumänderung: SR 301 in 50.34!

Bis zum nächsten Mal