Technische Universität München
Prof. J. Esparza / Dr. M. Luttenberger, S. Sickert

Sommer 2016
18. Mai 2016

HA-Lösung

TA-Lösung

Einführung in die theoretische Informatik – Aufgabenblatt 4

Beachten Sie: Soweit nicht explizit angegeben, sind Ergebnisse stets zu begründen!

Hausaufgaben: Abgabe bis zum 11.05.2016 (Mittwoch) um 12:00

Aufgabe 4.1 **Umkehrung (Spiegelung)** **1P+1P+3P**

In den Folien finden Sie einen Algorithmus mit dem man aus einem DFA M einen ε-NFA N konstruieren kann, so dass $L(M)^R = L(N)$ gilt.

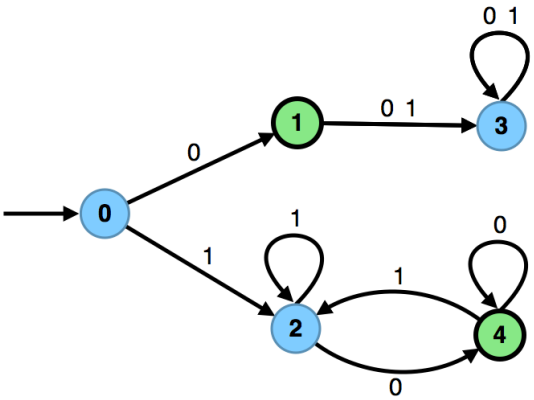
(a) Konstruieren Sie für den regulären Ausdruck $r = 0 \mid 1(0|1)^*0$ einen DFA und wenden Sie das Spiegelungsverfahren an.

(b) Geben Sie einen regulären Ausdruck r' mit $L(r) = L(r')^R$ an.

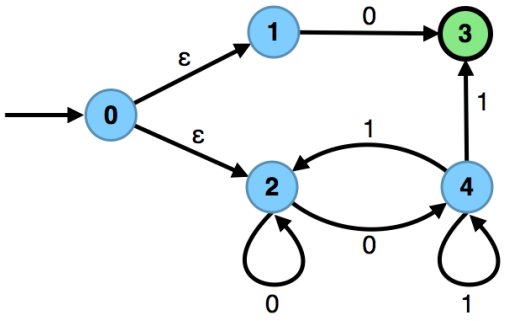
(c) Geben Sie nun, entsprechend TA 3.3(c) ein Verfahren an, dass rekursiv einen regulären Ausdruck r in einen regulären Ausdruck r' umschreibt, so dass $L(r)^R = L(r')$ gilt. Zeigen Sie die Korrektheit Ihres Verfahrens mittels struktureller Induktion und seien Sie besonders ausführlich in den Fällen $\alpha\beta$, $\alpha \mid \beta$ und α^*.

Lösung:

(a) Siehe Abbildung.



(a) DFA für $L(r)$



(b) ε-NFA für $L(r)^R$

(b) $r' = 0 \mid 0(0|1)^*1$

(c) **Konstruktion:** Wir definieren folgende rekursive Prozedur:

$$\begin{aligned} rev(\emptyset) &= \emptyset & rev(\alpha\beta) &= rev(\beta)rev(\alpha) \\ rev(\epsilon) &= \epsilon & rev(\alpha \mid \beta) &= rev(\alpha) \mid rev(\beta) \\ rev(a) &= a & rev(\alpha^*) &= rev(\alpha)^* \end{aligned}$$

Korrektheit: Da für alle $u, v \in \Sigma^*$ gilt: $(uv)^R = (v_m \dots v_1 u_n \dots u_1) = v^R u^R$, folgt für beliebige Sprachen A und B: $(AB)^R = B^R A^R$ und $(A^R)^* = (A^*)^R$. Dann können wir die Korrektheit $(L(rev(\gamma)) = L(\gamma)^R)$ mit struktureller Induktion über γ zeigen:

- $\gamma = \emptyset$: $L(rev(\emptyset)) = \emptyset = \emptyset^R = L(\emptyset)^R$
- $\gamma = \epsilon$: $L(rev(\epsilon)) = \{\epsilon\} = \{\epsilon\}^R = L(\epsilon)^R$
- $\gamma = a$: $L(rev(a)) = \{a\} = \{a\}^R = L(a)^R$
- $\gamma = \alpha\beta$: $L(rev(\alpha\beta)) = L(rev(\beta))L(rev(\alpha)) \overset{\text{IH}}{=} L(\beta)^R L(\alpha)^R = L(\alpha\beta)^R$
- $\gamma = \alpha \mid \beta$: $L(rev(\alpha \mid \beta)) = L(rev(\alpha)) \cup L(rev(\beta)) \overset{\text{IH}}{=} L(\alpha)^R \cup L(\beta)^R = L(\alpha \mid \beta)^R$
- $\gamma = \alpha^*$: $L(rev(\alpha^*)) = L(rev(\alpha))^* \overset{\text{IH}}{=} (L(\alpha)^R)^* = L(\alpha^*)^R$

Aufgabe 4.2 Pumping-Lemma **2P+2P**

Zeigen Sie mithilfe des Pumping-Lemmas, dass folgende Sprachen nicht regulär sind:

(a) $L_a = \{a^{6i}b^{6i} \mid i \geq 0\}$

(b) $L_b = \{a^{2^i} \mid i \geq 0\}$

Lösung:

(a) Angenommen: L_a wäre regulär. Dann können wir das Pumping-Lemma anwenden.

Sei $n \geq 1$ eine Pumping-Lemma-Zahl. Dann gilt $a^{6n}b^{6n} \in L_a$ und $\left|a^{6n}b^{6n}\right| \geq n$. Es gibt also für $z = a^{6n}b^{6n}$ eine Zerlegung $z = uvw$ mit $v \neq \epsilon$ und $|uv| \leq n$, so dass $(*)$ $uv^iw \in L_a$ für alle $i \in \mathbb{N}_0$. Deshalb $v = a^j$ für ein $0 < j \leq n$.

Dann ist aber $uv^0w = a^{6n-j}b^{6n} \notin L_a$, denn $6n - j < 6n$. Dies steht im Widerspruch zu $(*)$, d.h. unsere ursprüngliche Annahme, dass L_a regulär ist, ist falsch.

(b) Angenommen: L_b wäre regulär. Dann können wir das Pumping-Lemma anwenden.

Sei $n \geq 1$ eine Pumping-Lemma-Zahl. Dann gilt $a^{2^n} \in L_k$ und $\left|a^{2^n}\right| \geq n$. Es gibt also für $z = a^{2^n}$ eine Zerlegung $z = uvw$ mit $v \neq \epsilon$ und $|uv| \leq n$, so dass $(*)$ $uv^iw \in L_b$ für alle $i \in \mathbb{N}_0$. Deshalb $v = a^j$ für ein $0 < j \leq n$.

Dann ist aber $uv^2w = a^{2^n+j} \notin L_b$, denn

$$2^n < 2^n + j \leq 2^n + n < 2^n + 2^n = 2^{n+1}$$

Dies steht im Widerspruch zu $(*)$, d.h. unsere ursprüngliche Annahme, dass L_b regulär ist, ist falsch.

Aufgabe 4.3 Homomorphismen auf regulären Sprachen **3P+1P**

Sei Σ ein Alphabet und $L \subseteq \Sigma^*$ eine reguläre Sprache. Weiter sei $h\colon \Sigma \to \Sigma^*$ eine Abbildung, die jedem Zeichen $a \in \Sigma$ ein Wort $h(a) \in \Sigma^*$ zuordnet. h erweitert man kanonisch auf Σ^* mittels $h(\varepsilon) = \varepsilon$ und $h(xy) = h(x)h(y)$ für alle $x, y \in \Sigma^*$. Schließlich sei $h(L) = \{h(w) \mid w \in L\}$.

(a) Zeigen Sie: Ist L regulär, dann ist auch $h(L)$ regulär.

(b) Verwenden Sie das Resultat aus (a), um zu zeigen, dass $L' = \{ab^{3i}cd^{2i}e \mid i \in \mathbb{N}_0\}$ nicht regulär ist. Hierbei dürfen Sie das Ergebnis aus HA 4.2 verwenden.

Lösung:

(a) **Konstruktion:** Wir liften h auf reguläre Ausdrücke und erhalten h_{RE}:

$$\begin{aligned} h_{RE}(\emptyset) &= \emptyset & h_{RE}(\alpha\beta) &= h_{RE}(\beta)h_{RE}(\alpha) \\ h_{RE}(\epsilon) &= \epsilon & h_{RE}(\alpha \mid \beta) &= h_{RE}(\alpha) \mid h_{RE}(\beta) \\ h_{RE}(a) &= h(a) & h_{RE}(\alpha^*) &= h_{RE}(\alpha)^* \end{aligned}$$

Korrektheit: Wir zeigen $L(h_{RE}(\gamma)) = h(L(\gamma))$ mit struktureller Induktion über γ. Dabei verwenden wir das h strukturerhaltend ist: $h(A^n) = (h(A))^n$, sowie $h(A \cup B) = h(A) \cup h(B)$.

- $\gamma = \emptyset$: $L(h_{RE}(\emptyset)) = \emptyset = h(\emptyset)$
- $\gamma = \epsilon$: $L(h_{RE}(\epsilon)) = \{\epsilon\} = h(\{\epsilon\})$
- $\gamma = a$: $L(h_{RE}(a)) = \{h(a)\} = h(\{a\})$.
- $\gamma = \alpha\beta$: $L(h_{RE}(\alpha\beta)) = L(h_{RE}(\alpha))L(h_{RE}(\beta)) \overset{\text{IH}}{=} h(L(\alpha))h(L(\beta)) = h(L(\alpha)L(\beta)) = h(L(\alpha\beta))$

- $\gamma = \alpha \mid \beta$: $L(h_{RE}(\alpha \mid \beta)) = L(h_{RE}(\alpha)) \cup L(h_{RE}(\beta)) \overset{\text{IH}}{=} h(L(\alpha)) \cup h(L(\beta)) = h(L(\alpha) \cup L(\beta)) = h(L(\alpha \mid \beta))$
- $\gamma = \alpha^*$: $L(h_{RE}(\alpha^*)) = L(h_{RE}(\alpha))^* \overset{\text{IH}}{=} h(L(\alpha))^* = \bigcup_{i \geq 0} (h(L(\alpha)))^i = \bigcup_{i \geq 0} h(L(\alpha)^i) = h(L(\alpha^*))$

Sei r ein regulärer Ausdruck für L. Dann

$$h(L) = h(L(r)) = L(h_{RE}(r))$$

und somit $h(L)$ regulär.

Alternativ kann der Beweis anhand eines Automaten geführt werden. Dabei werden die Kantenbeschriftungen mit h umgeschrieben.

(b) Wir definieren eine Abbildung $h\colon \Sigma \to \Sigma^*$:

$$h(a) = \epsilon \qquad h(b) = aa \qquad h(c) = \epsilon \qquad h(d) = bbb \qquad h(e) = \epsilon$$

Wir nehmen an das L' regulär ist. Dann ist laut (a) auch

$$h(\{ab^{3i}cd^{2i}e \mid i \in \mathbb{N}_0\}) = \{a^{6i}b^{6i} \mid i \in \mathbb{N}_0\}$$

regulär. Widerspruch zu HA 4.3. L' ist somit nicht regulär.

Aufgabe 4.4 Produktkonstruktion **2P+1P**

Wir definieren den *ITE*-Operator (*if-then-else*) für Mengen:

$$ITE(A, B, C) = \{w \in \Sigma^* \mid (w \in A \wedge w \in B) \vee (w \notin A \wedge w \in C)\}$$

Seien jetzt $A, B, C \subseteq \Sigma^*$ reguläre Sprachen, die von den DFAs M_A, M_B und M_C erkannt werden.

(a) Erweitern Sie die Produktkonstruktion aus der Vorlesung, so dass aus den Automaten M_A, M_B und M_C ein DFA $M_{ITE(A,B,C)}$ mit $L(M_{ITE(A,B,C)}) = ITE(A, B, C)$ konstruiert wird. Beweisen Sie auch die Korrektheit dieser Konstruktion!

(b) Sei nun $\Sigma = \{a, b, c\}$. Wenden Sie diese Konstruktion auf die Sprachen $A = \{w \in \Sigma^* \mid |w| \text{ ist gerade}\}$, $B = \{a^n b^m \mid n, m \geq 0\}$ und $C = \{c^n b^m \mid n, m \geq 0\}$.

Lösung:

(a) **Konstruktion:**

$$\begin{aligned} M_{ITE(A,B,C)} &:= (Q^A \times Q^B \times Q^C, \Sigma, \delta_\times, q_{0\times}, F_\times) \\ \delta_\times((q, q', q''), a) &:= (\delta^A(q, a), \delta^B(q', a), \delta^C(q'', a)) \\ q_{0\times} &:= (q_0^A, q_0^B, q_0^C) \\ F_\times &:= \{(q^A, q^B, q^C) \mid (q^A \in F^A \wedge q^B \in F^B) \vee (q^A \notin F^A \wedge q^C \in F^C)\} \end{aligned}$$

Korrektheit: Mit Induktion über $w \in \Sigma^*$ erhalten wir für alle q, q', q'':

$$\hat{\delta}_\times((q, q', q''), w) = (\hat{\delta}^A(q, w), \hat{\delta}^B(q', w), \hat{\delta}^C(q'', w))$$

Damit gilt:

$$\begin{aligned} & w \in L(M_{ITE(A,B,C)}) \\ \Leftrightarrow\ & \hat{\delta}_\times(q_{0\times}, w) \in F_\times \\ \Leftrightarrow\ & (\hat{\delta}^A(q_0^A, w) \in F^A \wedge \hat{\delta}^B(q_0^B, w) \in F^B) \vee (\hat{\delta}^A(q_0^A, w) \notin F^A \wedge \hat{\delta}^C(q_0^C, w) \in F^C) \\ \Leftrightarrow\ & (w \in A \wedge w \in B) \vee (w \notin A \wedge w \in C) \\ \Leftrightarrow\ & w \in ITE(A, B, C) \end{aligned}$$

(b) Lösung:

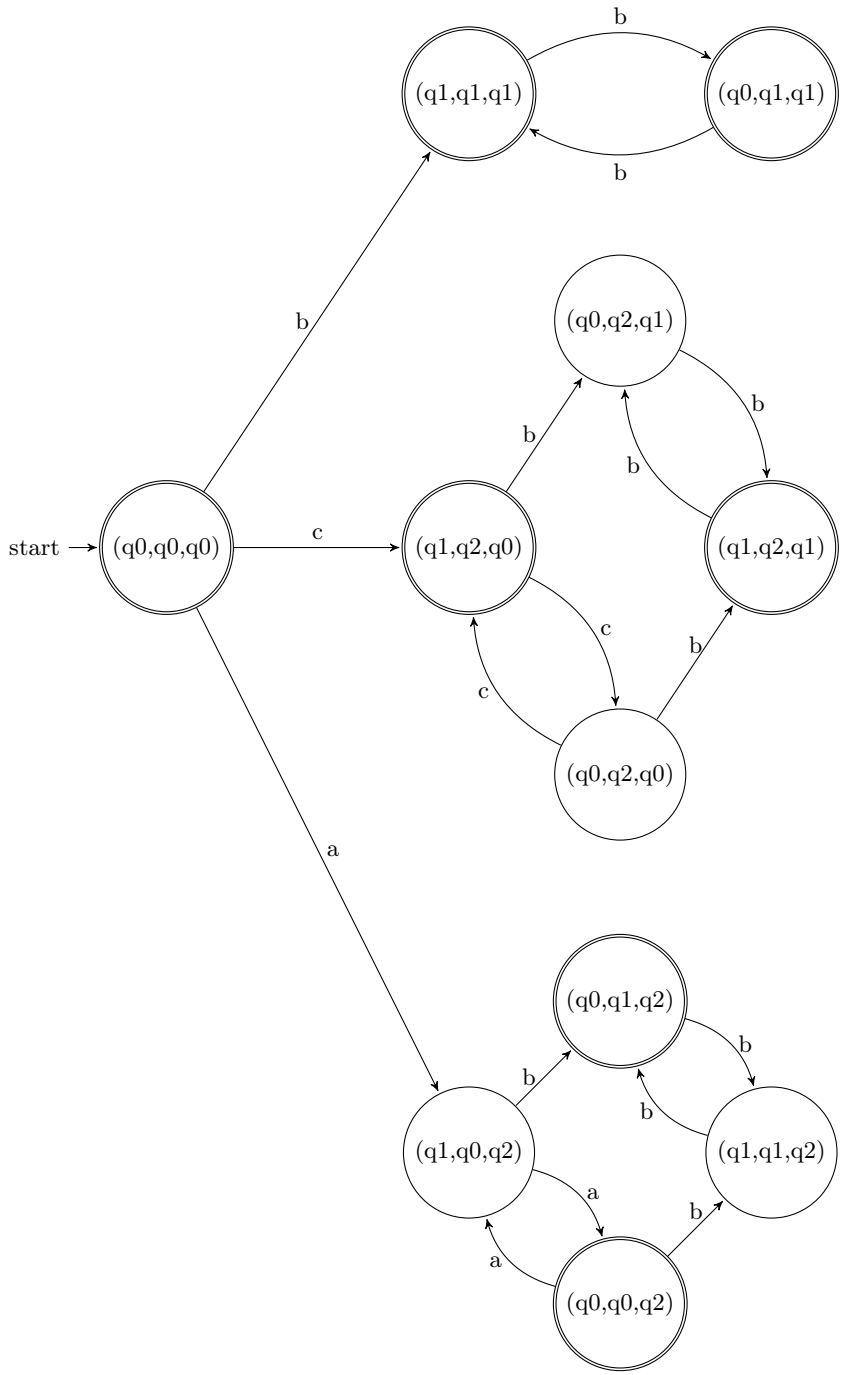


Abbildung 2: Produktautomat. Fangzustände sind nicht eingezeichnet.

Tutoraufgaben: Besprechung in KW19

Bitte beachten Sie die Regelung zu den Pfingstferien auf der Webseite.

Aufgabe 4.1

Geben Sie einen möglichst effizienten Algorithmus an, der für einen gegebenen NFA $N = (Q, \Sigma, \delta, q_0, F)$ den Wert $\min\{|w| \mid w \in L(N)\}$ berechnet.

Lösung:

```
def BFSReach(Q,Σ,δ,q0,F):
  if q0 in F:
    return 0
  if len(F) == 0:
    return None
  n = 0
  todo = set([q0])
  visited = set([q0])
  while len(todo) > 0:
    n += 1
```

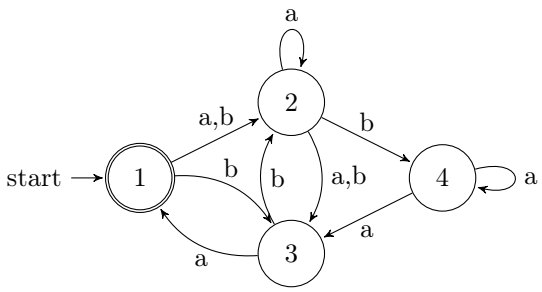
```
    nxt_todo = set()
    for q in todo:
      for a in Σ:
        for q' in δ(q,a):
          if q' not in visited:
            nxt_todo.add(q')
            visited.add(q')
            if q' in F:
              return n
    todo = nxt_todo
  return None
```

Erklärung: BFS auf Automat beginnend im Startzustand (für mehrere Startzustände entsprechend). Jede Transition wird höchstens einmal getestet.

Aufgabe 4.2

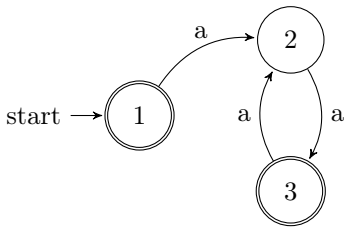
(a) Passen Sie den Minimierungsalgorithmus so an, dass er für jedes Paar von inäquivalenten Zuständen q, q' eines DFAs ein Wort w mit $\delta(q, w) \in F \Leftrightarrow \delta(q', w) \notin F$ berechnet.

(b) Determinisieren und dann minimieren Sie folgenden NFA. Verwenden Sie für die Minimierung den Algorithmus aus (a).



Lösung:

(a) Für alle Paare $\{p, q\} \in U$ kann man $W(\{p, q\}) := \varepsilon$ verwenden. Fügt man dann in der Schleife $\{p, q\}$ neu zu U hinzu, da $\{\delta(p, a), \delta(q, a)\} \in U$, so kann man $W(\{p, q\}) = aW(\{\delta(p, a), \delta(q, a)\})$ setzen. Hinweis für Tutoren: Wiederholen Sie den Algorithmus für:



(b) Der NFA muss erst zu einem DFA determinisiert werden:

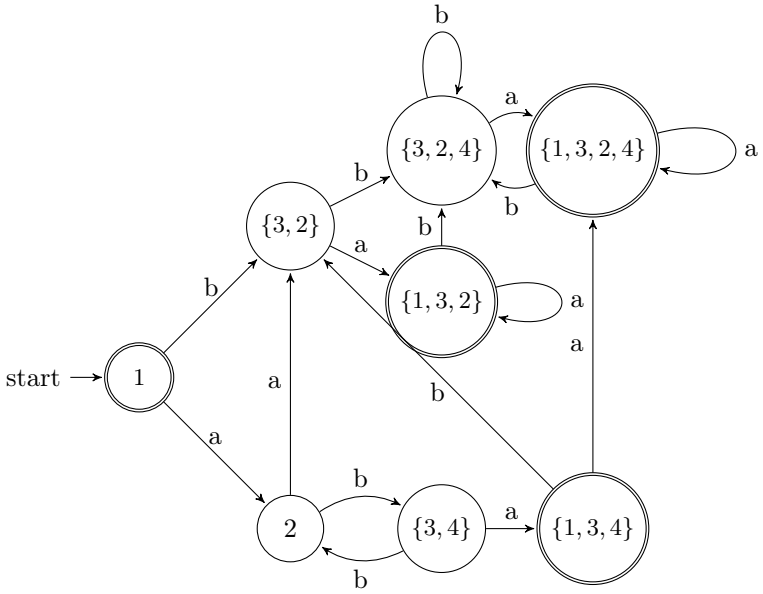


Abbildung 3: DFA nach Potenzmengenkonstruktion.

	{1,4,3}	{1,4,2,3}	{4,3}	{4,2,3}	{1,2,3}	{2,3}	{1}	{2}
{1,4,3}	–	–	–	–	–	–	–	–
{1,4,2,3}	=	–	–	–	–	–	–	–
{4,3}	ε	ε	–	–	–	–	–	–
{4,2,3}	ε	ε	ba	–	–	–	–	–
{1,2,3}	=	=	ε	ε	–	–	–	–
{2,3}	ε	ε	ba	=	ε	–	–	–
{1}	a	a	ε	ε	a	ε	–	–
{2}	ε	ε	a	a	ε	a	ε	–

Abbildung 4: Tabelle mit unterscheidenden Wörtern

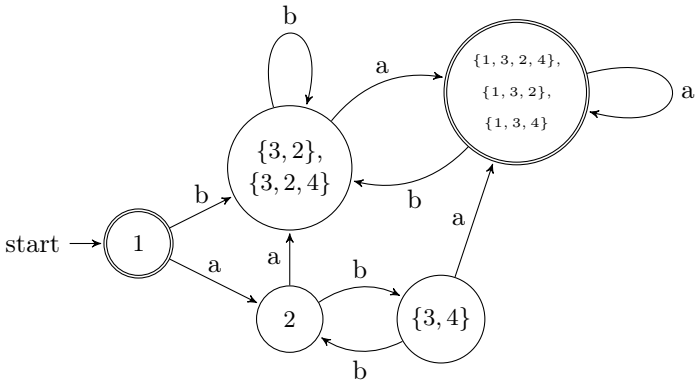


Abbildung 5: Minimierter DFA

Aufgabe 4.3

Bestimmen Sie die MyHill-Nerode-Relation für jede der folgenden Sprachen über dem Alphabet $\Sigma = \{a, b, c\}$ und skizzieren Sie jeweils den Quotientenautomaten $(\Sigma^*/\equiv_L, \Sigma, \delta_L, [\varepsilon]_{\equiv_L}, F_L)$.

(a) $L_1 = L((ab \mid ba)^*)$

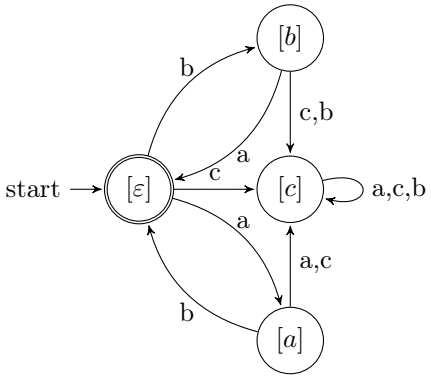
(b) $L_2 = L((aa)^*)$

(c) $L_3 = \{a^i b^i c^i \mid i \in \mathbb{N}_0\}$

Lösung:

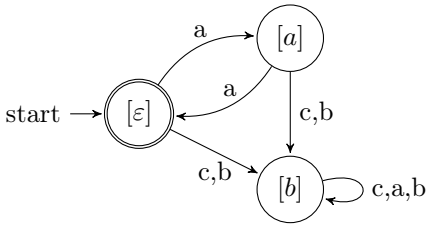
(a) Ablesen aus minimalem DFA:

$[\varepsilon]_{L_1} = L((ab \mid ab)^*) \quad [b]_{L_1} = L((ab \mid ba)^* b) \quad [a]_{L_1} = L((ab \mid ba)^* a) \quad [c]_{L_1} = L((ab \mid ba)^*(a(a \mid c) \mid b(b \mid c) \mid c)(a \mid b \mid c)^*)$

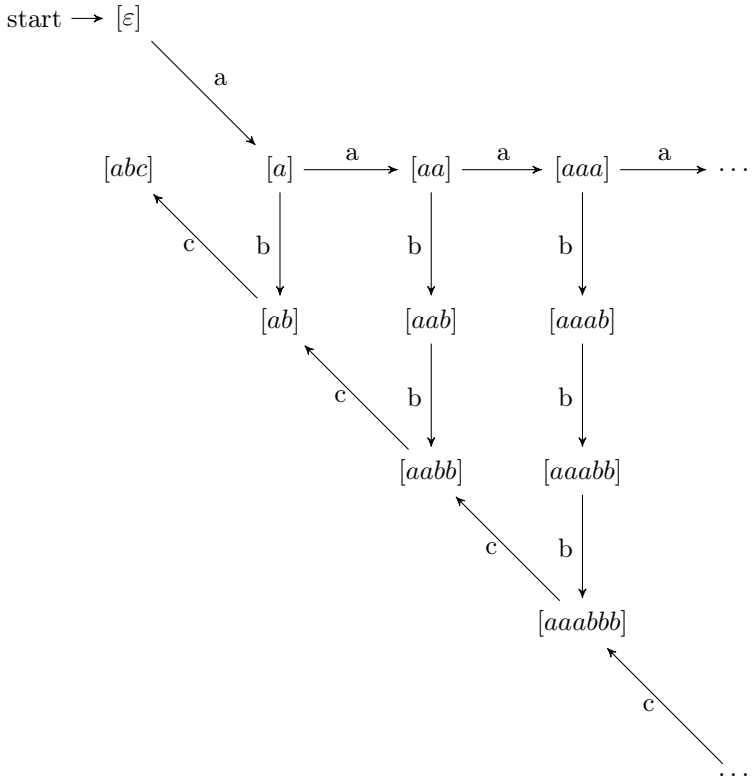


(b) Ablesen aus minimalem DFA:

$[\varepsilon]_{L_2} = L((aa)^*) \qquad [a]_{L_2} = L(a(aa)^*) \qquad [b]_{L_2} = L(a^*(b \mid c)(a \mid b \mid c)^*)$



(c) Ablehnender Zustand/Äquivalenzklasse und entsprechende Transitionen wurden nicht gezeichnet:



Knobelaufgabe

Freiwillige Abgabe per E-Mail an theo-uebungsleitung@in.tum.de. Der Preis diese Woche für die ersten drei richtigen Gesamtlösungen ist jeweils ein Chari Tea Red. Alle richtigen Einsendungen bis zum 11.05.2016 um 12:00 Uhr erhalten die Bonuspunkte. Wir bewerten nur die erste Einsendung und behalten uns das Recht vor unleserliche oder unverständliche Abgaben nicht zu korrigieren.

Aufgabe 4.1 **Bonus: 2P**

Eine Arena $(V, \Sigma, E, \sigma, v_0, Z)$ besteht aus einer endlichen Menge V von Knoten, einem endlichen Alphabet Σ, Σ-beschrifteten Kanten $E \subseteq V \times \Sigma \times V$, einer Abbildung $\sigma\colon V \to \{\top, \bot\}$, die jedem Knoten einen Spieler zuordnet, einem Startknoten v_0 und einer Menge Z von Zielknoten.

Gegeben ein Wort $w = a_0 a_1 \dots a_l \in \Sigma^*$ $(a_i \in \Sigma)$ sieht ein *Spielverlauf* wie folgt aus: beginnend in v_0 wählt stets Spieler $\sigma(v_i)$ den nächsten Knoten v_{i+1} aus allen über eine mit a_i beschriftete Kante erreichbaren Knoten, d.h. es muss $(v_i, a_i, v_{i+1}) \in E$ gelten. Spieler $\top$ gewinnt den Spielverlauf $v_0 v_1 \dots v_l v_{l+1}$, falls am Ende $v_{l+1} \in Z$ gilt, ansonsten gewinnt Spieler $\bot$.

Spieler $\top$ gewinnt das Wort w, falls er seine Züge so wählen kann, dass er jeden dann noch möglichen Spielverlauf gewinnt.

Zeigen Sie: Die Sprache der Wörter, die $\top$ gewinnt, ist regulär.