Prof. Dr. J. Giesl
M. Brockschmidt, F. Emmes, C. Fuhs, C. Otto, T. Ströder

Vorname: ______________________________

Nachname: ______________________________

Matrikelnummer: ______________________________

Studiengang (bitte ankreuzen):

- ○ Informatik Bachelor
- ○ Informatik Master (Auflage)
- ○ Mathematik Bachelor
- ○ Technik-Kommunikation M.A.
- ○ Informatik Lehramt
- ○ Informatik Promotion (Auflage)
- ○ Technik-Kommunikation Bachelor
- ○ Sonstige: ______________

	Anzahl Punkte	**Erreichte Punkte**
Aufgabe 1	**4**	
Aufgabe 2	**5**	
Aufgabe 3	**4**	
Aufgabe 4	**4**	
Aufgabe 5	**4**	
Aufgabe 6	**10**	
Aufgabe 7	**8**	
Aufgabe 8	**6**	
Summe	**45**	

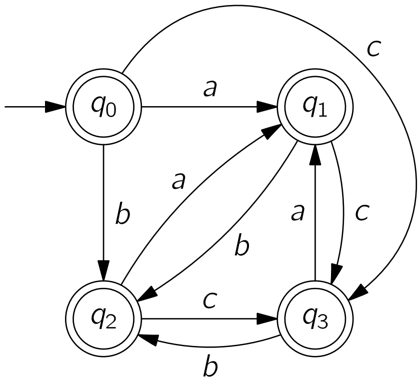
Hinweise:

- Geben Sie Ihre Antworten in lesbarer und verständlicher Form an.
- Schreiben Sie mit dokumentenechten Stiften, nicht mit roten Stiften oder mit Bleistiften.
- Bitte beantworten Sie die Aufgaben auf den Aufgabenblättern (benutzen Sie auch die Rückseiten).
- **Auf alle Blätter** (inklusive zusätzliche Blätter) müssen Sie **Ihren Namen und Ihre Matrikelnummer** schreiben.
- Was nicht bewertet werden soll, streichen Sie bitte durch.
- Werden **Täuschungsversuche** beobachtet, so wird die Präsenzübung mit **0 Punkten** bewertet.
- Geben Sie am Ende der Übung **alle Blätter zusammen mit den Aufgabenblättern ab**.

Matrikelnummer: **Name:**

Aufgabe 1 (Universalität): **(3 + 1 = 4 Punkte)**

Sei $\Sigma = \{a, b, c\}$ ein Alphabet. Betrachten Sie den folgenden NFA M.

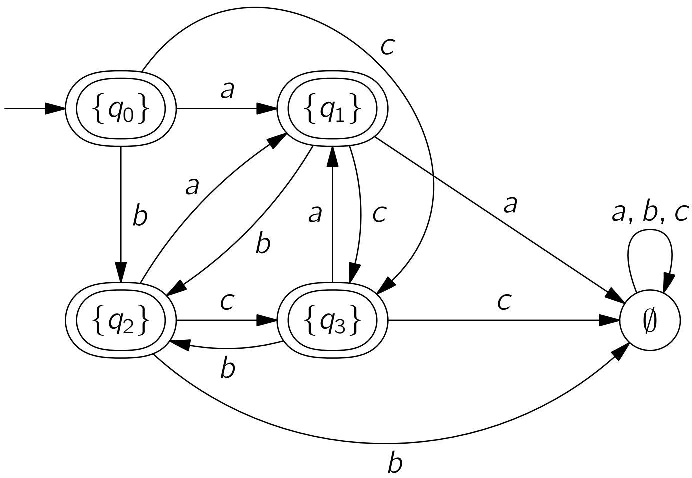


a) Überführen Sie den NFA M in einen DFA M' mit $L(M) = L(M')$.

b) Gilt $L(M) = \Sigma^*$? Geben Sie eine kurze Begründung für Ihre Aussage.

Lösung:

a) Wir wenden zuerst die Potenzmengenkonstruktion auf den NFA M an und erhalten den folgenden DFA M'.



b) Wir sehen, dass M' einen erreichbaren Zustand besitzt, der kein Endzustand ist. Daher gilt $L(M) = L(M') \neq \Sigma^*$.

Aufgabe 2 (Induktionsbeweis): **(5 Punkte)**

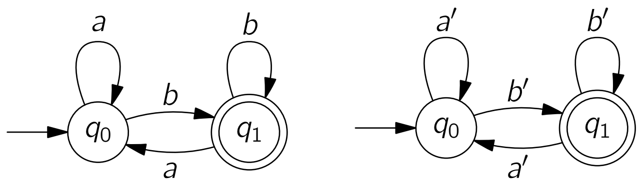
Seien $\Sigma = \{0, 1\}$ und $\Sigma' = \{0', 1'\}$ Alphabete. Wir definieren die folgende Funktion *prime* : $\Sigma^* \to \Sigma'^*$ mit

$$\begin{aligned} \mathit{prime}(\epsilon) &= \epsilon \\ \mathit{prime}(a \cdot w) &= a' \cdot \mathit{prime}(w) \text{ für } w \in \Sigma^*, a \in \Sigma \end{aligned}$$

Es gilt also zum Beispiel $\mathit{prime}(011) = 0'1'1'$.
Betrachten wir nun einen DFA $M = (Q, \Sigma, \delta, q_0, F)$.
Wir definieren dazu den Prime-Automaten $M' = (Q, \Sigma', \delta', q_0, F)$ mit

$$\delta'(q, a') = \delta(q, a) \text{ für alle } q \in Q, a' \in \Sigma'.$$

Diese Konstruktion ist untenstehend für einen DFA beispielhaft illustriert.



M M'

Beweisen Sie nun folgende Aussage für alle $w \in \Sigma^*$ per Induktion über die Wortlänge $|w|$:

$$w \in L(M) \text{ genau dann, wenn } \mathit{prime}(w) \in L(M')$$

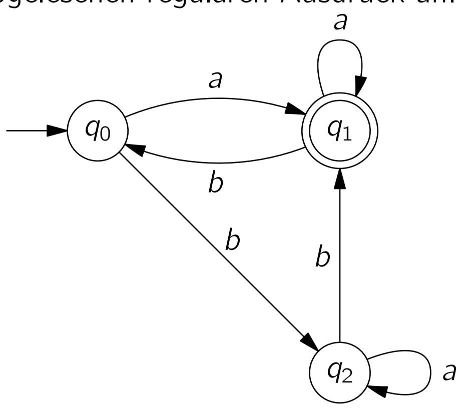
Lösung:

Es gilt $w \in L(M) \Leftrightarrow \mathit{prime}(w) \in L(M')$ gdw. $\hat{\delta}(q_0, w) \in F \Leftrightarrow \hat{\delta}'(q_0, \mathit{prime}(w)) \in F$. Wir zeigen die stärkere Aussage $\hat{\delta}(q, w) = \hat{\delta}'(q, \mathit{prime}(w))$ für alle $w \in \Sigma^*, q \in Q$.
Im Induktionsanfang ist $w = \epsilon$. Dann haben wir $\hat{\delta}(q, \epsilon) = q = \hat{\delta}'(q, \epsilon) = \hat{\delta}'(q, \mathit{prime}(\epsilon))$.
Als Induktionshypothese nehmen wir an, dass die Aussage für jedes v mit $|v| < |w|$ gilt.
Im Induktionsschritt ist $w = a \cdot v$. Nun gilt:

$$\begin{aligned} \hat{\delta}(q, a \cdot v) &\overset{\text{Def. } \hat{\delta}}{=} \hat{\delta}(\delta(q, a), v) \\ &\overset{\text{Def. } \delta'}{=} \hat{\delta}(\delta'(q, a'), v) \\ &\overset{\text{IH}}{=} \hat{\delta}'(\delta'(q, a'), \mathit{prime}(v)) \\ &\overset{\text{Def. } \hat{\delta}}{=} \hat{\delta}'(q, a' \cdot \mathit{prime}(v)) \\ &\overset{\text{Def. } \mathit{prime}}{=} \hat{\delta}'(q, \mathit{prime}(a \cdot v)) \end{aligned}$$

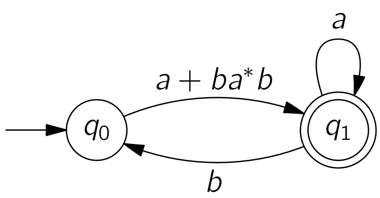
Aufgabe 3 (DFAs und reguläre Ausdrücke): **(4 Punkte)**

Sei $\Sigma = \{a, b\}$ ein Alphabet. Wandeln Sie folgenden DFA über Σ in einen regulären Ausdruck um, indem Sie Zustände schrittweise entfernen und die betroffenen Kanten durch reguläre Ausdrücke ersetzen. Geben Sie den resultierenden Automaten nach Entfernung von q_2 an und geben Sie den zum Schluss abgelesenen regulären Ausdruck an.



Lösung:

Im ersten Schritt entfernen wir den Zustand q_2.



Der resultierende Ausdruck ist:

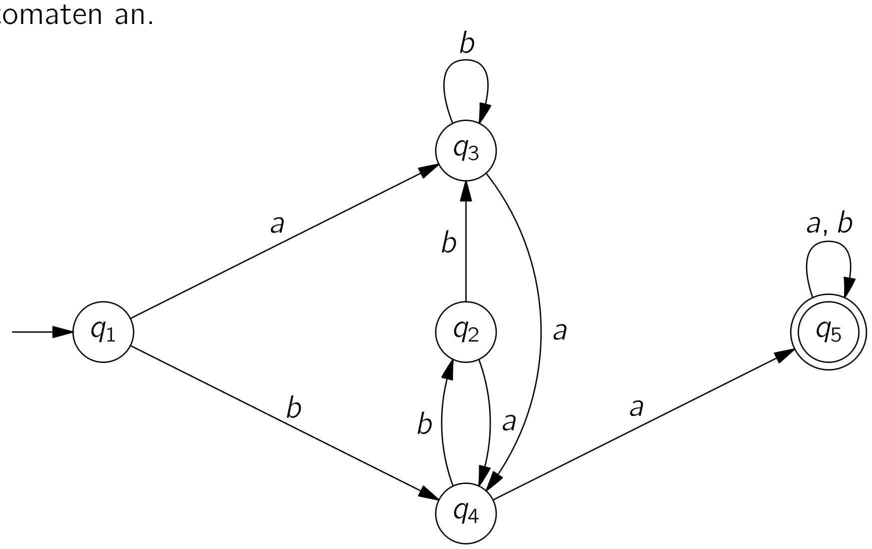
$$(a + ba^*b)(a + b(a + ba^*b))^*$$

Eine alternative Lösung ist:

$$(a + ba^*b)a^*(b(a + ba^*b)a^*)^*$$

Aufgabe 4 (DFA minimieren): (4 Punkte)

Verwenden Sie den **schnellen** Markierungsalgorithmus aus der Vorlesung, um folgenden DFA zu minimieren. Geben Sie die beim Anwenden des Algorithmus entstehende Tabelle und den minimalen Automaten an.

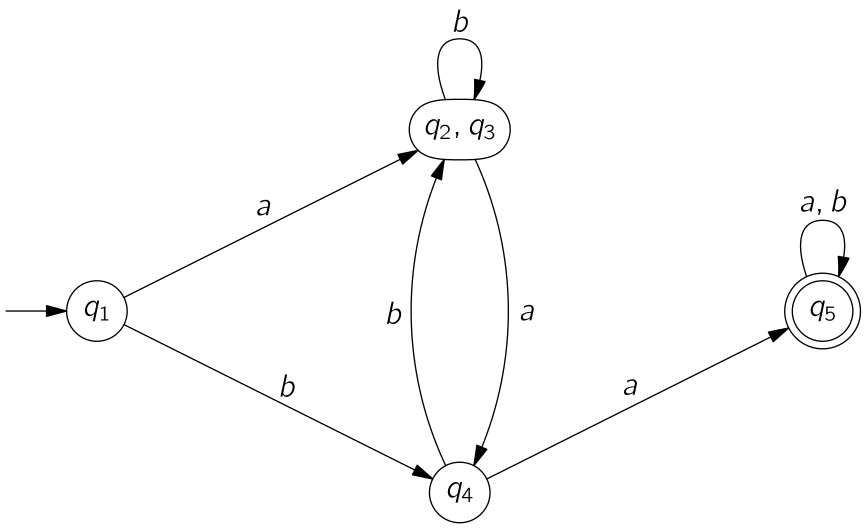


Lösung:

Tabelle des Markierungsalgorithmus:

2	×			
3	×			
4	×	×	(1, 2), (1, 3), ×	
5	×	×	×	×
	1	2	3	4

Minimierter Automat:



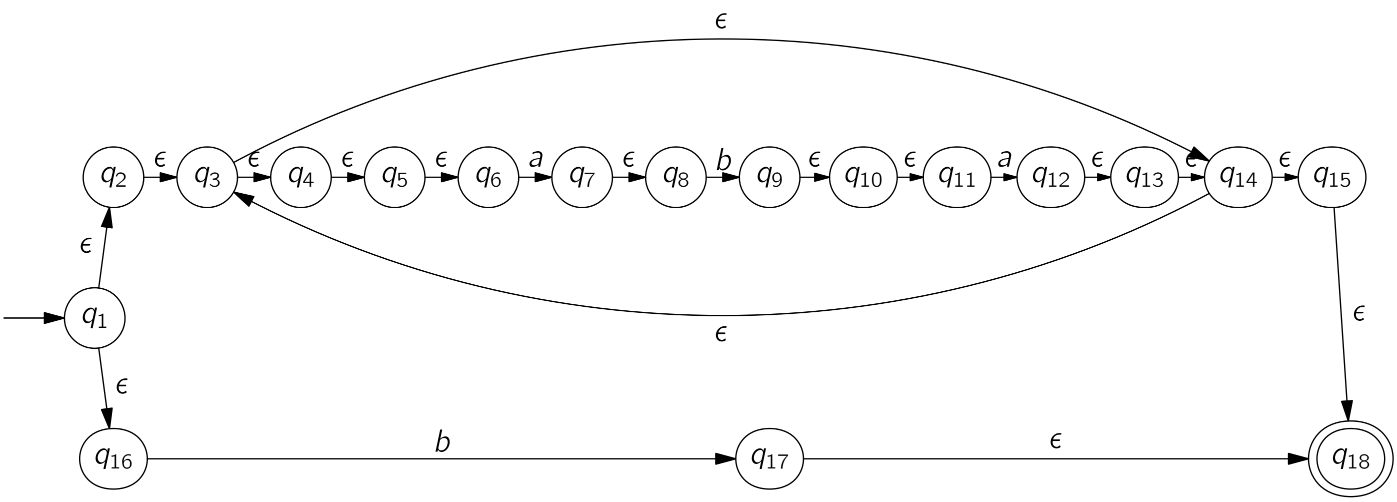
____.

Matrikelnummer: **Name:**

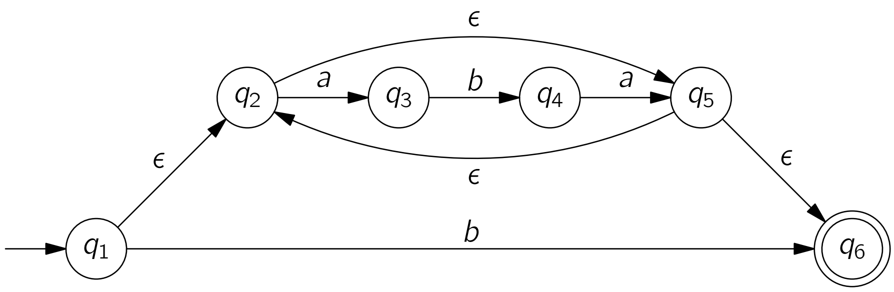
Aufgabe 5 (Thompson-Konstruktion): (4 Punkte)

Geben Sie einen ϵ-NFA an, der die Sprache $(aba)^* + b$ akzeptiert.

Lösung:



Alternative Lösung:



Matrikelnummer: **Name:**

Aufgabe 6 (Pumping Lemma): **(2 + 8 = 10 Punkte)**

a) Geben Sie das Pumping Lemma für reguläre Sprachen an.

b) Führen Sie für alle $i \in \{1, 2, 3\}$ die folgende Aufgabe für die Sprache L_i über dem Alphabet Σ_i durch:

- Falls L_i regulär ist, dann geben Sie einen regulären Ausdruck r_i mit $L(r_i) = L_i$ an.
- Falls L_i nicht regulär ist, dann beweisen Sie dies mit Hilfe des Pumping Lemmas.

Wir erinnern an die Funktion $rev : \Sigma^* \to \Sigma^*$ aus den Übungen:

$$\begin{aligned} rev(\epsilon) &:= \epsilon \\ rev(w \cdot a) &:= a \cdot rev(w) \qquad\qquad a \in \Sigma, w \in \Sigma^* \end{aligned}$$

- $L_1 := \{w \cdot rev(w) \mid w \in \Sigma_1^*\}$ über $\Sigma_1 := \{a\}$.
- $L_2 := \{0^k(10)^k \mid k \in \mathbb{N}_0\}$ über $\Sigma_2 := \{0, 1\}$.
- $L_3 := \{(0^k 1^{k+2})^\ell \mid k, \ell \in \mathbb{N}_0, k < 2\}$ über $\Sigma_3 := \{0, 1\}$.

Lösung:

a) Sei L eine reguläre Sprache. Dann gibt es $n \in \mathbb{N}_0$, so dass jedes Wort $w \in L$ mit $|w| \geq n$ in $w = xyz$ zerlegt werden kann mit

- $|xy| \leq n$
- $|y| > 0$
- $xy^iz \in L$ für alle $i \geq 0$

b)
- Da das Alphabet nur ein Symbol a enthält, gilt $rev(w) = w$ für alle $w \in \Sigma_1^*$. Jedes Wort in L_1 ist also von der Gestalt a^{2k} für ein $k \in \mathbb{N}_0$. Daher können wir die Sprache L_1 mit dem regulären Ausdruck $(aa)^*$ beschreiben.
- Die Sprache L_2 ist nicht regulär.

  Beweis: Angenommen, sie ist regulär. Dann gibt es nach dem Pumping Lemma eine Länge $n \in \mathbb{N}_{>0}$, so dass jedes Wort w in xyz zerlegt werden kann, so dass xy^iz für alle $i \in \mathbb{N}_0$ in L_2 liegt.

  Wir wählen das Wort $w = 0^n(10)^n \in L_2$. Nach Pumping Lemma gibt es eine Zerlegung $w = xyz$, wobei nach Konstruktion $y = 0^\ell$ für ein $\ell \in \mathbb{N}$ mit $1 \leq \ell \leq n$ ist. Dann muss aber auch $xy^0z = 0^{n-\ell}(10)^n \in L_2$ liegen. Da aber $\ell > 0$ gilt, ist dies nicht der Fall. Dies ist ein Widerspruch zur Annahme und wir können schließen, dass L_2 nicht regulär ist.
- Da k beschränkt ist, ist die Sprache regulär. Es gilt $L_3 = \{(11)^\ell \mid \ell \in \mathbb{N}_0\} \cup \{(0111)^\ell \mid \ell \in \mathbb{N}_0\}$. Wir können daher L_3 mit dem regulären Ausdruck $(11)^* + (0111)^*$ beschreiben.

Matrikelnummer: **Name:**

Aufgabe 7 (Myhill-Nerode-Äquivalenzklassen): (5 + 3 = 8 Punkte)

Führen Sie für alle $i \in \{1, 2\}$ die folgende Aufgabe für die Sprache L_i über dem Alphabet $\Sigma := \{0, 1\}$ durch:

- Falls L_i regulär ist, geben Sie den minimalen Myhill-Nerode-Automaten an.
- Falls L_i nicht regulär ist, dann begründen Sie, warum der Index von $\equiv_{L_i}$ unendlich ist. Zeigen Sie hierzu, wie man unendlich viele Wörter $w_0, w_1, \dots$ erhalten kann, so dass $w_k \not\equiv_{L_i} w_\ell$ für alle $k \neq \ell$ ist.

Wir führen zunächst die Funktionen $\sharp_0, \sharp_1 : \Sigma^* \to \mathbb{N}_0$ ein, die die Vorkommen von 0 bzw. 1 in einem Wort w zählen. Formal ist $\sharp_0$ wie folgt definiert:

$$\begin{aligned} \sharp_0(\epsilon) &:= 0 \\ \sharp_0(w \cdot 0) &:= \sharp_0(w) + 1 && w \in \Sigma^* \\ \sharp_0(w \cdot 1) &:= \sharp_0(w) && w \in \Sigma^* \end{aligned}$$

$\sharp_1$ ist analog definiert.

a) $L_1 := \{w \in \Sigma^* \mid \sharp_0(w) = \sharp_1(w)\}$

b) $L_2 := \{w \in \Sigma^* \mid \sharp_0(w) \text{ ist durch 3 teilbar}\}$

Lösung:

a) L_1 ist nicht regulär.

Als Folge von Wörtern w_i ist z.B. $0, 00, 000, \dots$, also $w_i = 0^i$ geeignet. Damit gilt für $k \neq \ell$, dass $w_k \cdot 1^k \in L_1$ und $w_\ell \cdot 1^k \notin L_1$ ist.

Alternativ definieren wir $C_k := \{w \in \{0, 1\}^* \mid \sharp_0(w) - \sharp_1(w) = k\}$, d.h. C_k enthält genau die Worte, in denen 0 k mal öfter auftritt als 1. Für $k = 0$ treten beide also gleich oft auf, für $k < 0$ kommt 1 $-k$ mal häufiger vor als 0.

Die Menge der Äquivalenzklassen $\{0, 1\}^* / \equiv_{L_1}$ ist nun $\{C_k \mid k \in \mathbb{Z}\}$.

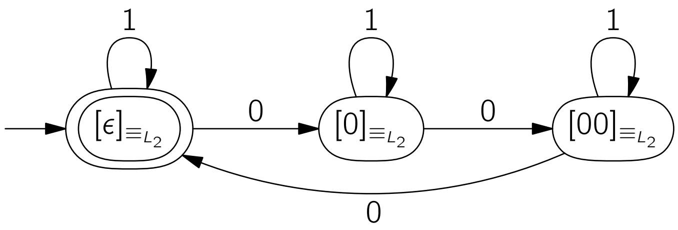
Seien nun $k \neq \ell$ und damit $C_k \neq C_\ell$ zwei verschiedene Äquivalenzklassen. Seien $w_k \in C_k$ und $w_l \in C_\ell$ Vetreter dieser Klassen. Wir suchen nun ein trennendes Wort u, so dass $w_k \cdot u \in L_1$ und $w_\ell \cdot u \notin L_1$ gilt.

Offensichtlich gilt, dass $\sharp_a$ ein Monoid-Homomorphismus von $(\Sigma, \cdot)$ nach $(\mathbb{N}, +)$ ist. Nach Konstruktion der Äquivalenzklassen gibt es außerdem natürliche Zahlen t, t', so dass $\sharp_0(w_k) = t + k$ und $\sharp_1(w_k) = t$ bzw. $\sharp_0(w_\ell) = t' + \ell$ und $\sharp_1(w_\ell) = t'$ gelten.

Fall $k \geq 0$: Wir wählen $u = 1^k$. Es gilt $\sharp_0(u) = 0$ und $\sharp_1(u) = k$. Dann ist nach Vorüberlegung $\sharp_0(w_k \cdot u) = t + k + 0 = t + k = \sharp_1(w_k \cdot u)$. Es gilt also $w_k \cdot u \in L_1$. Hingegen gilt $\sharp_0(w_\ell \cdot u) = t' + \ell \neq t' + k = \sharp_1(w_\ell \cdot u)$ nach Voraussetzung $k \neq \ell$.

Fall $k < 0$: Analog mit $u = 0^{-k}$.

b) Die Menge der Äquivalenzklassen ist $\{\{w \in \{0,1\}^* \mid \sharp_0(w)\ mod\ 3 = k\} \mid k \in \{0,1,2\}\}$.
Der minimale DFA für die Sprache L_2 sieht daher wie folgt aus:



Matrikelnummer: **Name:**

Aufgabe 8 (Reguläre und Nicht-Reguläre Sprachen): (1,5 + 1,5 + 1,5 + 1,5 = 6 Punkte)

Sei $\Sigma = \{a, b\}$ ein Alphabet. Beantworten Sie folgende Fragen mit einer kurzen Begründung.

a) Gibt es eine reguläre Sprache, die Teilmenge einer nicht-regulären Sprache ist?

b) Gibt es eine nicht-reguläre Sprache, die Teilmenge einer regulären Sprache ist?

c) Gibt es eine reguläre Sprache L_1 und eine nicht-reguläre Sprache L_2 über Σ mit $L_1 \cap L_2 = \emptyset$?

d) Gibt es eine reguläre Sprache L_1 und eine nicht-reguläre Sprache L_2 über Σ mit $L_1 \cup L_2 = \Sigma^*$?

Lösung:

Bekannt aus der Vorlesung ist, dass $L = \{a^n b^n \mid n \geq 0\}$ nicht regulär ist.

a) Ja. Beispiel: $\emptyset$ ist regulär und Teilmenge von L.

b) Ja. Beispiel: L ist nicht-regulär und Teilmenge von Σ^*.

c) Ja. Beispiel: $L_1 = \emptyset$ und $L_2 = L$.

d) Ja. Beispiel: $L_1 = \Sigma^*$ und $L_2 = L$.