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: ____________________

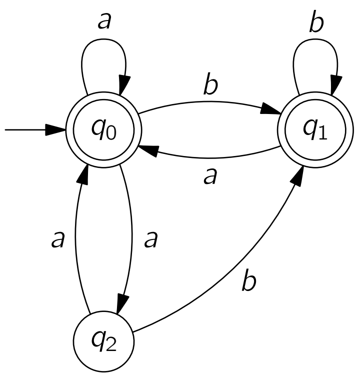
	Anzahl Punkte	**Erreichte Punkte**
Aufgabe 1	**5**	
Aufgabe 2	**5**	
Aufgabe 3	**4**	
Aufgabe 4	**4**	
Aufgabe 5	**4**	
Aufgabe 6	**8**	
Aufgabe 7	**3**	
Aufgabe 8	**4**	
Summe	**37**	

Hinweise:

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

Aufgabe 1 (Endliche Automaten): **(3 + 1 + 1 = 5 Punkte)**

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



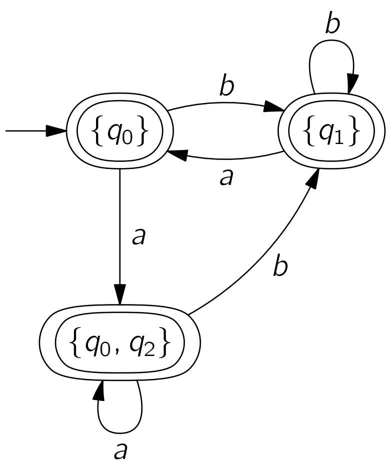
a) Überführen Sie den NFA M in einen DFA M' mit $L(M) = L(M')$, indem Sie den Potenzautomaten zu M bilden.

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

c) Geben Sie den minimalen DFA M_{min} mit $L(M) = L(M_{min})$ an. Sie müssen für diese Aufgabe **nicht** den Minimierungsalgorithmus aus der Vorlesung anwenden. Das Ergebnis aus Aufgabenteil b) könnte jedoch hilfreich sein.

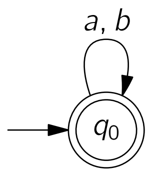
Lösung:

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



b) Wir sehen, dass der DFA M' ausschließlich Endzustände besitzt. Daher gilt $L(M) = L(M') = \Sigma^*$.

c) Da $L(M) = \Sigma^*$ gilt, können wir M_{min} direkt angeben.



Aufgabe 2 (Induktionsbeweis): (5 Punkte)

Sei Σ ein Alphabet und $\sigma : \Sigma \to \Sigma$ eine Permutation über Σ. Beispielsweise wäre für $\Sigma = \{a, b, c\}$ die Funktion σ mit $\sigma(a) = b$, $\sigma(b) = c$ und $\sigma(c) = a$ eine Permutation.
Wir erweitern σ zu einer Funktion $\overline{\sigma} : \Sigma^* \to \Sigma^*$ auf Wörtern, indem wir jedes Zeichen einzeln permutieren:

$$\overline{\sigma}(w) = \begin{cases} \overline{\sigma}(w') \cdot \sigma(a) & \text{, falls } w = w' \cdot a \text{ mit } a \in \Sigma, w' \in \Sigma^* \\ \epsilon & \text{, falls } w = \epsilon \end{cases}$$

Für σ wie oben gilt also $\overline{\sigma}(abc) = bca$.
Zu jedem DFA $M = (Q, \Sigma, \delta, q_0, F)$ erstellen wir nun einen DFA $\overline{M} = (Q, \Sigma, \delta', q_0, F)$, der die permutierten Wörter erkennen soll. Dafür definieren wir $\delta'(q, \sigma(a)) = \delta(q, a)$.
Um nachzuweisen, dass $L(\overline{M}) = \{\overline{\sigma}(w) \mid w \in L(M)\}$ gilt, beweisen Sie die allgemeinere Aussage

$$\hat{\delta}'(q, \overline{\sigma}(w)) = \hat{\delta}(q, w)$$

für alle $q \in Q$ und verwenden Sie dabei Induktion über die Wortlänge $|w|$.

Lösung:

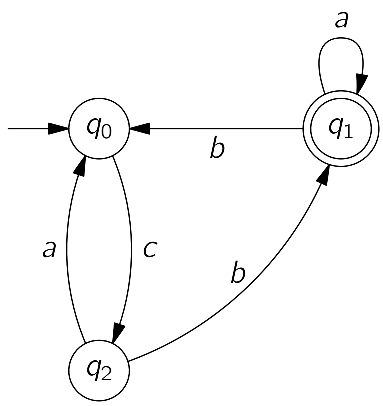
Im Induktionsanfang ist $w = \epsilon$ und es gilt $\hat{\delta}'(q, \overline{\sigma}(\epsilon)) = \hat{\delta}'(q, \epsilon) = q = \hat{\delta}(q, \epsilon)$.
Im Induktionsschritt betrachten wir $w = w' \cdot a$ für ein $a \in \Sigma$ und setzen voraus, dass die Aussage für w' bereits gilt.
Dann gilt:

$$\begin{aligned} \hat{\delta}'(q, \overline{\sigma}(w)) &\overset{w=w'\cdot a}{=} \hat{\delta}'(q, \overline{\sigma}(w' \cdot a)) \\ &\overset{\text{Def. } \overline{\sigma}}{=} \hat{\delta}'(q, \overline{\sigma}(w') \cdot \sigma(a)) \\ &\overset{\text{Def. } \hat{\delta}'}{=} \delta'(\hat{\delta}'(q, \overline{\sigma}(w')), \sigma(a)) \\ &\overset{\text{IH}}{=} \delta'(\hat{\delta}(q, w'), \sigma(a)) \\ &\overset{\text{Def. } \delta'}{=} \delta(\hat{\delta}(q, w'), a) \\ &\overset{\text{Def. } \hat{\delta}}{=} \hat{\delta}(q, w' \cdot a) \\ &\overset{w=w'\cdot a}{=} \hat{\delta}(q, w) \end{aligned}$$

Nach dem Induktionsprinzip ist die Aussage damit bewiesen.

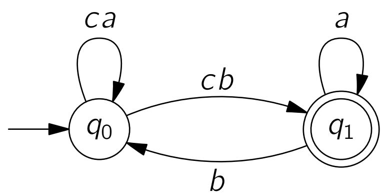
Aufgabe 3 (NFAs und reguläre Ausdrücke): **(4 Punkte)**

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



Lösung:

Im ersten Schritt entfernen wir den Zustand q_2.



Der resultierende Ausdruck ist:

$$(ca)^*cb(a + b(ca)^*cb)^*$$

Alternative Lösung:

$$(ca)^*cba^*(b(ca)^*cba^*)^*$$

Aufgabe 4 (Unproduktive Symbole): **(2 + 2 = 4 Punkte)**

Sei $G := (N, T, P, S)$ eine kontextfreie Grammatik mit $N := \{S, A, B, C\}, T := \{a, b, c\}$ und P wie folgt:

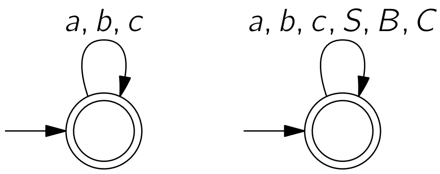
$$\begin{array}{lcl} S & \to & AB \mid a \\ A & \to & SA \\ B & \to & CA \mid b \\ C & \to & BB \mid c \end{array}$$

a) Ermitteln sie die Menge $\{Z \in N \mid \text{es gibt kein } w \in T^* \text{ mit } Z \Rightarrow^* w\}$ der unproduktiven Nonterminale mit Hilfe des in der Vorlesung vorgestellten Verfahrens. Geben Sie als Zwischenergebnis auch den NFA für die Sprache $pre^*_G(T^*)$ an.

b) Geben Sie eine zu G äquivalente Grammatik G' an, in der nur noch produktive, vom Startsymbol S aus erreichbare Nonterminale und Terminale existieren und aus der die unproduktiven Nonterminale entfernt wurden.

Lösung:

a) Zuerst berechnen wir den NFA zur Erkennung von T^* und daraus den NFA für $pre^*_G(T^*)$.



Die Menge der unproduktiven Symbolen ist nun $N \setminus pre^*_G(T^*) = \{A\}$.

b) Wir erzeugen eine Grammatik, in der die A-Regeln nicht mehr auftreten:

$$\begin{array}{lcl} S & \to & a \\ B & \to & b \\ C & \to & BB \mid c \end{array}$$

Da aber S nun nur noch zu dem Terminalsymbol a ableitet, sind die B- und C-Produktionen nicht mehr erreichbar und wir erhalten:

$$S \to a$$

Aufgabe 5 (CYK-Algorithmus): **(4 Punkte)**

Gegeben sei die folgende Grammatik G in Chomsky-Normalform.

$$\begin{array}{rcl} S & \to & AC \\ A & \to & R_aB \mid R_aR_b \\ B & \to & R_bC \\ C & \to & R_bR_c \mid CR_c \mid R_cR_c \\ R_a & \to & a \\ R_b & \to & b \\ R_c & \to & c \end{array}$$

Testen Sie mit dem CYK-Algorithmus, ob das Wort $w = abcc$ in $L(G)$ liegt.

Lösung:

$w =$	a	b	c	c
1	R_a	R_b	R_c	R_c
2	A	C	C	
3		B, C		
4	A, S			

$\Rightarrow w \in L(G)$

Aufgabe 6 (Pumping-Lemma): **(2 + 4 + 2 = 8 Punkte)**

a) In dieser Aufgabe geht es darum, zu beweisen, dass aus den Eigenschaften des regulären Pumping-Lemmas die Eigenschaften des kontextfreien Pumping-Lemmas folgen.

Sei L eine Sprache, für die folgendes gilt:

Es gibt ein $n \in \mathbb{N}_0$, so dass jedes Wort $z \in L$ mit $|z| \geq n$ in $z = rst$ zerlegt werden kann mit

- $|rs| \leq n$
- $|s| > 0$
- $rs^it \in L$ für alle $i \geq 0$

Zeigen Sie, dass L dann folgende Eigenschaften erfüllt.

Es gibt ein $n \in \mathbb{N}_0$, so dass jedes Wort $z \in L$ mit $|z| \geq n$ in $z = uvwxy$ zerlegt werden kann mit

- $|vwx| \leq n$
- $|vx| > 0$
- $uv^iwx^iy \in L$ für alle $i \geq 0$

b) Betrachten Sie folgende Sprache $L_1 = \{a^k ba^k ba^k \mid k \geq 0\}$.

Beweisen Sie mithilfe des Pumping-Lemmas für kontextfreie Sprachen, dass L_1 nicht kontextfrei ist.

c) Sei L eine Sprache, die folgende Eigenschaften **nicht** erfüllt:

Es gibt ein $n \in \mathbb{N}_0$, so dass jedes Wort $z \in L$ mit $|z| \geq n$ in $z = uvwxy$ zerlegt werden kann mit

- $|vwx| \leq n$
- $|vx| > 0$
- $uv^iwx^iy \in L$ für alle $i \geq 0$

Zeigen Sie, dass L dann unendlich viele Äquivalenzklassen bzgl. der Myhill-Nerode-Relation $\equiv_L$ hat.

Sie dürfen selbstverständlich Sätze aus der Vorlesung in ihrem Beweis verwenden.

Lösung:

a) Wähle $u = v = \epsilon$, $w = r$, $x = s$ und $y = t$. Dann gilt $|vwx| = |rs| \leq n$, $|vx| = |s| > 0$ und $uv^iwx^iy = rs^it \in L$ für alle $i \in \mathbb{N}$. Damit ist die Aussage bewiesen.

b) Sei $n \in \mathbb{N}_0$. Wir wählen das Wort $z = a^nba^nba^n \in L_1$ mit $|z| > n$. Sei $z = uvwxy$ mit $|vwx| \leq n$ und $|vx| > 0$. Wir wählen $i = 0$, d.h. wir wollen zeigen, dass $uwy \notin L_1$ gilt. Wegen $|vwx| \leq n$ kann vwx nicht zweimal das Zeichen b enthalten. Falls vx ein b enthält, gilt sofort $uwy \notin L_1$. Andernfalls enthält vx mindestens ein a. Gleichzeitig kann aber vx nicht gleich viele a Zeichen im Bereich vor, nach und zwischen den beiden b Zeichen enthalten, da schon vwx nicht beide b Zeichen enthält. Damit gilt wiederum $uwy \notin L_1$ und nach dem Pumping-Lemma für kontextfreie Sprachen ist damit L_1 nicht kontextfrei.

c) Da L das Pumping-Lemma für kontextfreie Sprachen nicht erfüllt, ist L nicht kontextfrei. Damit ist L insbesondere auch nicht regulär. Nach dem Satz von Myhill-Nerode ist dies äquivalent dazu, dass L unendlich viele Äquivalenzklassen bzgl. $\equiv_L$ hat.

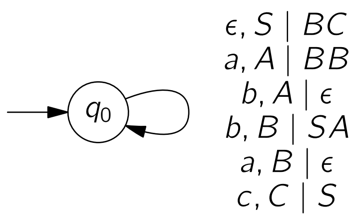
Aufgabe 7 (Kellerautomaten): (3 Punkte)

Gegeben sei die folgende Grammatik G:

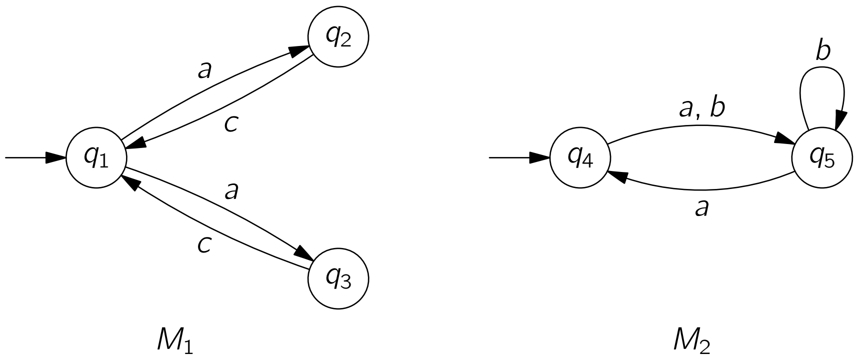
$$\begin{array}{lcl} S & \to & BC \\ A & \to & aBB \mid b \\ B & \to & bSA \mid a \\ C & \to & cS \end{array}$$

Geben Sie einen Kellerautomaten M an, so dass $N(M) = L(G)$ gilt. Ihr Automat sollte dabei nicht mehr als 6 Transitionen enthalten.

Lösung:



Aufgabe 8 (Synchronisiertes Produkt): **(4 Punkte)**



Gegeben seien die NFAs M_1 über dem Alphabet $\{a, c\}$ und M_2 über dem Alphabet $\{a, b\}$. Berechnen Sie das synchronisierte Produkt $M_1 \circ M_2$.

Lösung: