Klausur - Formale Systeme, Automaten und Prozesse

16.08.2016

Vorname :

Nachname :

Matrikelnummer :

Studiengang :
- ☐ Informatik Bachelor
- ☐ Informatik Master (Auflage)
- ☐ Technik-Kommunikation Bachelor
- ☐ Informatik Lehramt
- ☐ Mathematik Bachelor
- ☐ Technik-Kommunikation M.A.
- ☐ Sonstige:

	Punkte
1.	/16
2.	/18
3.	/10
4.	/10
5.	/12
6.	/10
7.	/8
8.	/8
9.	/8
$\sum$	/100

Allgemeine Hinweise:

- **Auf alle Blätter** (inklusive zusätzliche Blätter) müssen Sie **Ihren Vornamen, Ihren Nachnamen und Ihre Matrikelnummer** schreiben.
- 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).
- Geben Sie für jede Aufgabe **maximal eine** Lösung an. Streichen Sie alles andere durch. Andernfalls werden alle Lösungen der Aufgabe mit **0 Punkten** bewertet.
- Werden **Täuschungsversuche** beobachtet, so wird die Klausur mit **0 Punkten** bewertet.
- Die Bearbeitungszeit beträgt **120 Minuten**.
- Geben Sie am Ende der Klausur **alle Blätter zusammen mit den Aufgabenblättern ab**.

Name: Matrikelnummer:

Aufgabe 1 **2+2+2+2+2+2+2+2=16 Punkte**

Beantworten Sie die folgenden Fragen.

a) Gibt es für jeden PDA einen DFA, der dieselbe Sprache erkennt?

b) Seien r_1 und r_2 reguläre Ausdrücke. Gibt es immer einen regulären Ausdruck, der die Sprache $L(r_1) \cap L(r_2)$ beschreibt?

c) Wie viele Wörter gibt es in der Sprache $L((a + ab)(b + bb))$?

d) Ist für jede reguläre Sprache L die Sprache $L \setminus \{\varepsilon\}$ regulär?

e) Ist der Schnitt zweier kontextfreier Sprachen kontextfrei?

f) Gibt es in jedem Petrinetz eine Markierung, die eine Verklemmung ist? Begründen Sie Ihre Antwort kurz in 1-3 Sätzen.

g) Gibt es eine reguläre Sprache L, für die alle Äquivalenzklassen von $\sim_L$ endlich sind? Begründen Sie Ihre Antwort kurz in 1-3 Sätzen.

h) Sei $\mathcal{A} := (Q, \Sigma, \delta, q_0, F)$ ein minimaler DFA. Dann erkennt der DFA $\mathcal{A}' := (Q, \Sigma, \delta, q_0, Q \setminus F)$ das Komplement von $L(\mathcal{A})$. Ist $\mathcal{A}'$ unbedingt minimal? Begründen Sie Ihre Antwort kurz in 1-3 Sätzen.

Lösung:

Name: Matrikelnummer:

Aufgabe 2 **6+6+6=18 Punkte**

a) Geben Sie einen DFA an, der die Sprache

$$L_1 := \{w \in \{a,b\}^* \mid w \text{ hat nicht das Suffix } bba.\}$$

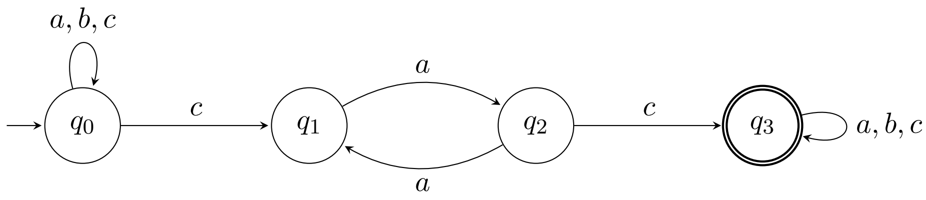
über dem Alphabet $\{a,b\}$ erkennt.

b) Geben Sie einen NFA an, der die Sprache

$$L_2 := \{w \in \{a,b,c\}^* \mid |w| \geq 3 \text{ oder } w \text{ beginnt mit } c.\}$$

über dem Alphabet $\{a,b,c\}$ erkennt.

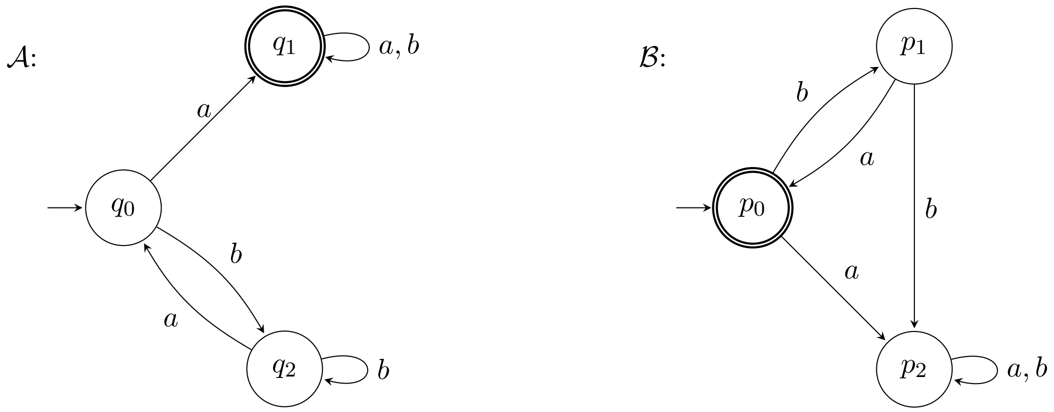
c) Geben Sie einen regulären Ausdruck an, der die Sprache beschreibt, die von dem folgenden NFA erkannt wird.



Lösung:

Aufgabe 3 **5+5=10 Punkte**

Betrachten Sie die folgenden DFAs.



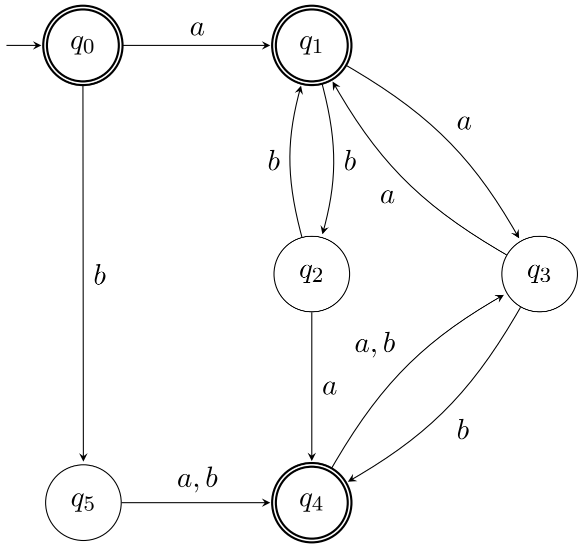
a) Geben Sie einen DFA ohne unerreichbare Zustände an, der die Sprache $L(\mathcal{A}) \cup L(\mathcal{B})$ akzeptiert.

b) Geben Sie einen DFA ohne unerreichbare Zustände an, der die Sprache $L(\mathcal{A}) \cap L(\mathcal{B})$ akzeptiert.

Lösung:

Aufgabe 4 **7+3=10 Punkte**

Wir betrachten den folgenden DFA $\mathcal{A}$ über dem Alphabet $\{a, b\}$.



a) Berechnen Sie den minimalen DFA für $L(\mathcal{A})$ mit dem Algorithmus aus der Vorlesung. Geben Sie dabei in jedem Schritt an,

- welche Klasse
- mit welchem Buchstaben und
- bezüglich welcher Klasse

Sie verfeinern.

b) Geben Sie die Myhill-Nerode Äquivalenzklassen für $L(\mathcal{A})$ an.

Lösung:

Aufgabe 5 **6+6=12 Punkte**

Sind die folgenden Sprachen über dem Alphabet $\{a, b, c\}$ kontextfrei? Beweisen Sie jeweils, dass die Sprache nicht kontextfrei ist oder geben Sie eine kontextfreie Grammatik an, die sie erzeugt.

a)

$$L := \big\{w \in \{a, b, c\}^* \mid |w|_a = |w|_b\big\}$$

b)

$$K := \big\{w \in \{a, b, c\}^* \mid |w|_a = |w|_b = |w|_c\big\}$$

Lösung:

Aufgabe 6 **8+2=10 Punkte**

Gegeben sei die kontextfreie Grammatik $\mathcal{G} = (\{S, A, B, C\}, \{a, b, c\}, P, S)$ wobei P die folgenden Regeln beinhaltet:

$$\begin{array}{llcll} S \to & SS \mid AC \mid DB \mid c & \qquad & C \to & SA \mid CS \\ A \to & AS \mid SA \mid a & & D \to & BS \\ B \to & BS \mid b & & & \end{array}$$

a) Bestimmen Sie mit Hilfe des CYK-Algorithmus, ob das Wort *abcba* in $L(\mathcal{G})$ liegt.

b) Geben Sie einen Ableitungsbaum für das Wort *bcb* in $\mathcal{G}$ an.

Lösung:

a					
	b				
		c			
			b		
				a	

Aufgabe 7 **8 Punkte**

Konstruieren Sie einen PDA, der mit **leerem Keller** akzeptiert und die folgende Sprache über dem Alphabet $\{a, b\}$ erkennt.

$$L = \{uv \mid |u|_a = |v|_b\}$$

Hierfür reicht es aus den PDA graphisch anzugeben. Beschreiben Sie außerdem **kurz** (1-2 Sätze) die Funktionsweise des von Ihnen konstruierten PDAs.

Lösung:

Aufgabe 8 **8 Punkte**

Für zwei Wörter v und w sagen wir, dass v ein *Dropwort* von w ist, wenn v aus w durch Weglassen von Buchstaben entsteht. Zum Beispiel sind $abc, bc, ac, ab, a, b, c, \varepsilon$ Dropwörter von abc. Das Wort ba is aber kein Dropwort von abc.

Formal sagen wir, dass v ein *Dropwort* von einem Wort $w = a_1 a_2 \ldots a_k$ ist, wenn es eine Zahl l, mit $0 \leq l \leq k$, und Indizes $i_1 < i_2 < \ldots < i_l$ aus $\{1, \ldots, k\}$ gibt, so dass $v = a_{i_1} a_{i_2} \ldots a_{i_l}$.

Sei Σ ein Alphabet und $L \subseteq \Sigma^*$ eine reguläre Sprache. Zeigen Sie, dass dann die Sprache

$$L_{\text{Drop}} := \{v \in \Sigma^* \mid \ v \text{ ist Dropwort eines Wortes } w \in L\}$$

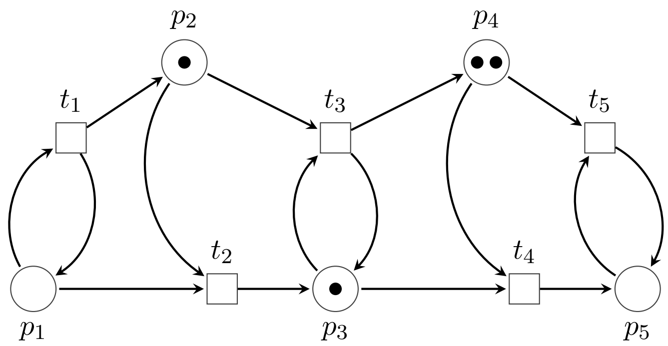
auch regulär ist.

Es reicht, wenn Sie ausgehend von einem NFA, der L erkennt, einen endlichen Automaten konstruieren, der L_{Drop} erkennt. Geben Sie Ihre Konstruktion formal an, und erklären Sie kurz die Idee dahinter.

Lösung:

Aufgabe 9 **2+2+2+2=8 Punkte**

Betrachten Sie das folgende Petrinetz $\mathcal{N}$ mit der Markierung $M = (0, 1, 1, 2, 0)$:



Beantworten Sie folgende Fragen und geben Sie jeweils eine kurze Begründung an.

a) Ist in $(\mathcal{N}, M)$ eine Markierung M' erreichbar, so dass $(\mathcal{N}, M')$ verklemmt ist?

b) Ist $(\mathcal{N}, M)$ beschränkt?

c) Geben Sie die Transitionsmatrix D von $\mathcal{N}$ an.

d) Geben Sie einen Lauf von $(1, 0, 0, 0, 0)$ nach $(0, 0, 0, 0, 1)$ an.

Lösung:

Klausur - Formale Systeme, Automaten und Prozesse
mit Lösungen

16.08.2016

Name: Matrikelnummer:

Aufgabe 1 **2+2+2+2+2+2+2+2=16 Punkte**

Beantworten Sie die folgenden Fragen.

a) Gibt es für jeden PDA einen DFA, der dieselbe Sprache erkennt?

b) Seien r_1 und r_2 reguläre Ausdrücke. Gibt es immer einen regulären Ausdruck, der die Sprache $L(r_1) \cap L(r_2)$ beschreibt?

c) Wie viele Wörter gibt es in der Sprache $L((a + ab)(b + bb))$?

d) Ist für jede reguläre Sprache L die Sprache $L \setminus \{\varepsilon\}$ regulär?

e) Ist der Schnitt zweier kontextfreier Sprachen kontextfrei?

f) Gibt es in jedem Petrinetz eine Markierung, die eine Verklemmung ist? Begründen Sie Ihre Antwort kurz in 1-3 Sätzen.

g) Gibt es eine reguläre Sprache L, für die alle Äquivalenzklassen von $\sim_L$ endlich sind? Begründen Sie Ihre Antwort kurz in 1-3 Sätzen.

h) Sei $\mathcal{A} := (Q, \Sigma, \delta, q_0, F)$ ein minimaler DFA. Dann erkennt der DFA $\mathcal{A}' := (Q, \Sigma, \delta, q_0, Q \setminus F)$ das Komplement von $L(\mathcal{A})$. Ist $\mathcal{A}'$ unbedingt minimal? Begründen Sie Ihre Antwort kurz in 1-3 Sätzen.

Lösung:

a) Nein

b) Ja

c) 3 *(Die Wörter ab, abb und abbb sind in der Sprache)*

d) Ja

e) Nein

f) Nein. Wenn der Vorbereich einer Transition leer ist, so kann diese immer schalten. Das Petrinetz ist also nie verklemmt.

g) Nein. Reguläre Sprachen L haben endlich viele $\sim_L$ Äquivalenzklassen. Es können also nicht alle endlich sein, da Σ^* unendlich ist.

h) Ja. Wenn ein kleinerer DFA für das Komplement existieren würde, erhielte man durch Anwendern dieser Konstruktion einen kleineren DFA für $L(\mathcal{A})$. Dies ist im Widerspruch zur Minimalität von $\mathcal{A}$.

Name: Matrikelnummer:

Aufgabe 2 **6+6+6=18 Punkte**

a) Geben Sie einen DFA an, der die Sprache

$$L_1 := \{w \in \{a,b\}^* \mid w \text{ hat nicht das Suffix } bba.\}$$

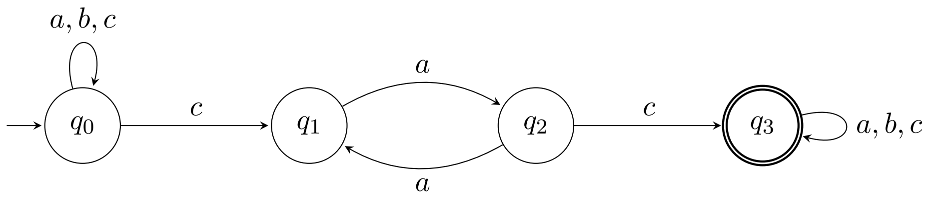
über dem Alphabet $\{a,b\}$ erkennt.

b) Geben Sie einen NFA an, der die Sprache

$$L_2 := \{w \in \{a,b,c\}^* \mid |w| \geq 3 \text{ oder } w \text{ beginnt mit } c.\}$$

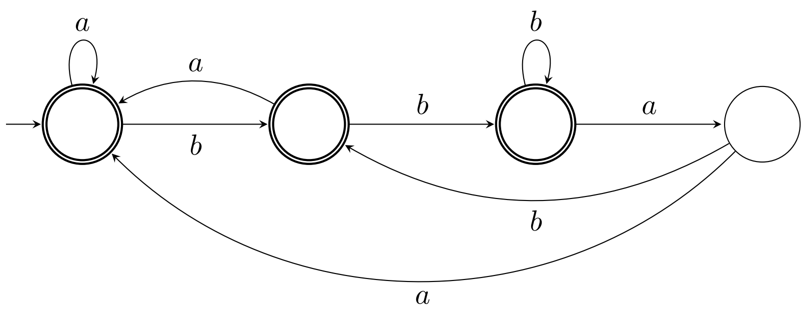
über dem Alphabet $\{a,b,c\}$ erkennt.

c) Geben Sie einen regulären Ausdruck an, der die Sprache beschreibt, die von dem folgenden NFA erkannt wird.

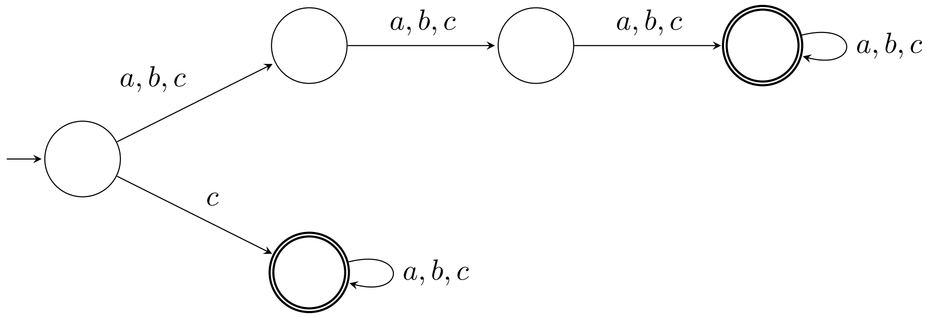


Lösung:

a)



b)

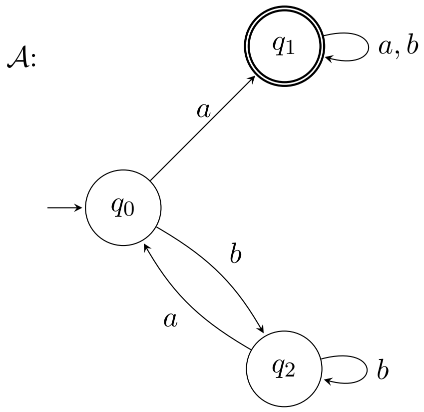


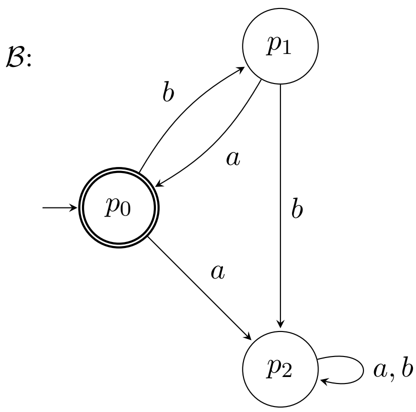
c) Im folgenden Ausdruck steht Σ für $(a + b + c)$:

$$\Sigma^* ca(aa)^* c\Sigma^*$$

Aufgabe 3 **5+5=10 Punkte**

Betrachten Sie die folgenden DFAs.



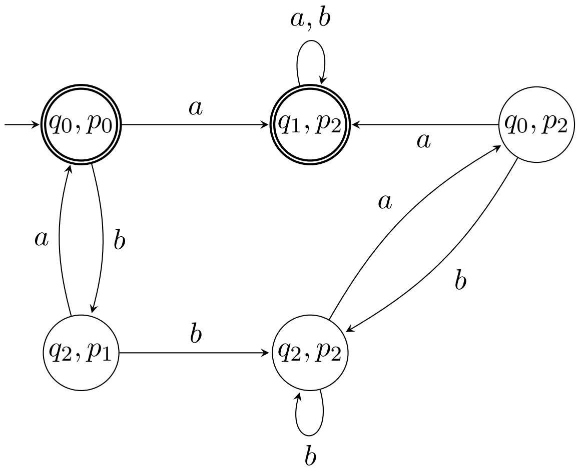


a) Geben Sie einen DFA ohne unerreichbare Zustände an, der die Sprache $L(\mathcal{A}) \cup L(\mathcal{B})$ akzeptiert.

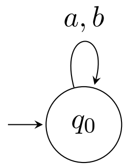
b) Geben Sie einen DFA ohne unerreichbare Zustände an, der die Sprache $L(\mathcal{A}) \cap L(\mathcal{B})$ akzeptiert.

Lösung:

a)

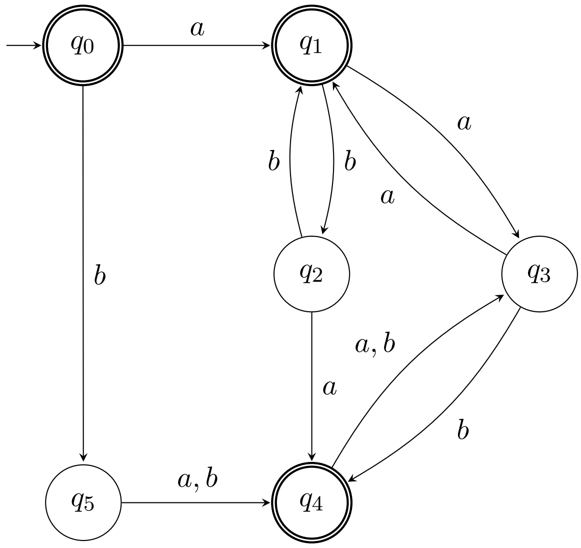


b) Wähle $\emptyset$ als Endzustandsmenge für den Automaten aus Teilaufgabe **a)**.
Alternative Lösung: $\mathcal{B}$ erkennt die Sprache $(ba)^*$. $\mathcal{A}$ akzeptiert kein Wort der Sprache $(ba)^*$, da jeder Lauf auf einem solchen Wort in Zustand q_0 endet. Folglich ist $L(\mathcal{A}) \cap L(\mathcal{B}) = \emptyset$. Der entsprechende DFA ist:


a, b
q0

Aufgabe 4 **7+3=10 Punkte**

Wir betrachten den folgenden DFA $\mathcal{A}$ über dem Alphabet $\{a, b\}$.



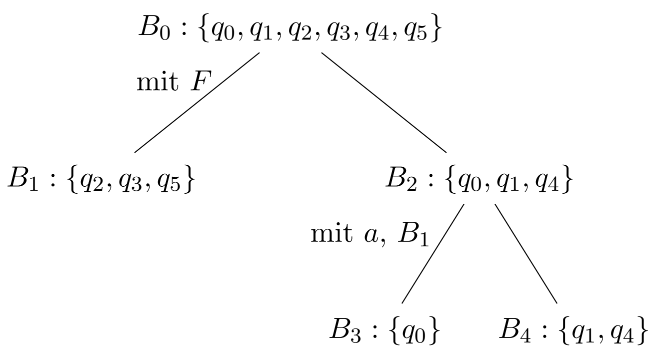
a) Berechnen Sie den minimalen DFA für $L(\mathcal{A})$ mit dem Algorithmus aus der Vorlesung. Geben Sie dabei in jedem Schritt an,
- welche Klasse
- mit welchem Buchstaben und
- bezüglich welcher Klasse

Sie verfeinern.

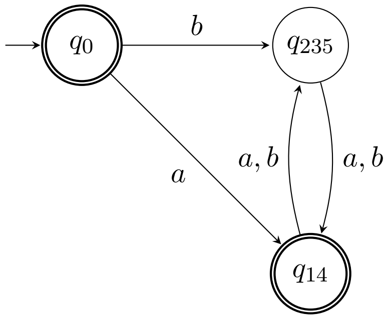
b) Geben Sie die Myhill-Nerode Äquivalenzklassen für $L(\mathcal{A})$ an.

Lösung:

a) Der Verfeinerungsalgorithmus für $\mathcal{A}$ kann dann wie folgt dargestellt werden:



Wir erhalten damit den folgenden minimalen Automaten:



b) Sei $L = L(\mathcal{A})$.

$$\begin{aligned}
\varepsilon/_L &= \{\varepsilon\} \\
a/_L &= \{aw \mid |w| \text{ ist gerade}\} \cup \{bw \mid |w| \text{ ist ungerade}\} \\
b/_L &= \{aw \mid |w| \text{ ist ungerade}\} \cup \{bw \mid |w| \text{ ist gerade}\}
\end{aligned}$$

Aufgabe 5 **6+6=12 Punkte**

Sind die folgenden Sprachen über dem Alphabet $\{a, b, c\}$ kontextfrei? Beweisen Sie jeweils, dass die Sprache nicht kontextfrei ist oder geben Sie eine kontextfreie Grammatik an, die sie erzeugt.

a)

$$L := \{w \in \{a, b, c\}^* \mid |w|_a = |w|_b\}$$

b)

$$K := \{w \in \{a, b, c\}^* \mid |w|_a = |w|_b = |w|_c\}$$

Lösung:

- L ist kontextfrei (sogar regulär):

  $\mathcal{G} = (\{S\}, \{a, b, c\}, P, S)$ erzeugt L, wobei P die folgenden Regeln beinhaltet:

  $$S \to aSbS \mid bSaS \mid cS \mid \varepsilon$$

- Angenommen: K ist kontextfrei.

  Dann existiert nach dem Pumping Lemma für kontextfreie Sprachen eine Zahl $n \geq 1$, sodass jedes Wort $z \in K$ mit $|z| \geq n$ sich zerlegen lässt als $z = uvwxy$ mit

  (i) $vx \neq \varepsilon$,

  (ii) $|vwx| \leq n$,

  (iii) $uv^kwx^ky \in K$ für alle $k \in \mathbb{N}$.

  Betrachte das Wort $z := a^n b^n c^n \in K$. Es hat die Länge $\geq n$.

  Sei $uvwxy$ eine Zerlegung von z mit den Eigenschaften (i), (ii) und (iii).

  Wegen der Eigenschaft 2 ist vwx ein Infix von $a^n b^n$ oder ein Infix von $b^n c^n$.

  - 1. Fall: vwx ist ein Infix $a^n b^n$.

    Wegen Eigensachaft 1 gilt dann $|uwy|_a < |uwy|_c$ oder $|uwy|_b < |uwy|_c$.

    Damit gilt $uwy \notin K$.

  - 2. Fall: vwx ist ein Infix $b^n c^n$.

    Wegen Eigensachaft 1 gilt dann $|uwy|_b < |uwy|_a$ oder $|uwy|_c < |uwy|_a$.

    Damit gilt $uwy \notin K$.

  Insgesamt erhalten wir einen Widerspruch zur Eigenschaft 3.

Aufgabe 6 **8+2=10 Punkte**

Gegeben sei die kontextfreie Grammatik $\mathcal{G} = (\{S, A, B, C\}, \{a, b, c\}, P, S)$ wobei P die folgenden Regeln beinhaltet:

$$\begin{array}{llcll} S \to & SS \mid AC \mid DB \mid c & \quad & C \to & SA \mid CS \\ A \to & AS \mid SA \mid a & & D \to & BS \\ B \to & BS \mid b & & & \end{array}$$

a) Bestimmen Sie mit Hilfe des CYK-Algorithmus, ob das Wort *abcba* in $L(\mathcal{G})$ liegt.

b) Geben Sie einen Ableitungsbaum für das Wort *bcb* in $\mathcal{G}$ an.

Lösung:

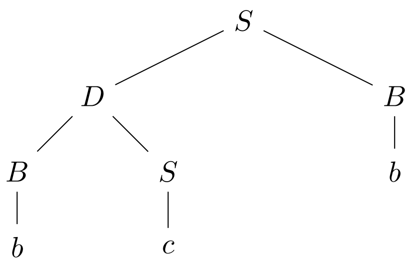
a					
	b				
		c			
			b		
				a	

a)

a	A	$\emptyset$	$\emptyset$	A	S
	b	B	B, D	S	A, C
		c	S	$\emptyset$	$\emptyset$
			b	B	$\emptyset$
				a	A

Da $S \in N_1, 5$ enthalten ist, gilt $abcba \in L(\mathcal{G})$.

b) Ein Ableitungsbaum für *bcb* in $\mathcal{G}$ ist:



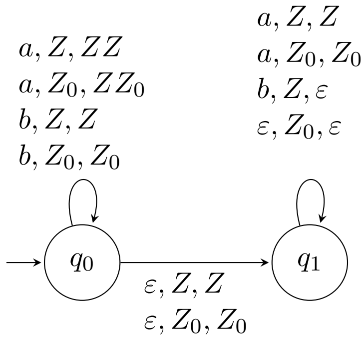
Aufgabe 7 **8 Punkte**

Konstruieren Sie einen PDA, der mit **leerem Keller** akzeptiert und die folgende Sprache über dem Alphabet $\{a, b\}$ erkennt.

$$L = \{uv \mid |u|_a = |v|_b\}$$

Hierfür reicht es aus den PDA graphisch anzugeben. Beschreiben Sie außerdem **kurz** (1-2 Sätze) die Funktionsweise des von Ihnen konstruierten PDAs.

Lösung:



In q_0 legt der Automat für jedes gelesene a ein zusätzliches Z auf den Keller und entscheidet sich irgendwann nichtdeterministisch, den zweiten Teil des Wortes zu lesen. (Übergang von q_0 nach q_1 ohne den Keller zu verändern.) Dann wird für jedes gelesene b ein Z entfernt. Wenn schließlich der Keller leer ist (nur noch Z_0 auf dem Keller), kann der PDA das Kellerbodensymbol entfernen und somit akzeptieren.

Aufgabe 8 **8 Punkte**

Für zwei Wörter v und w sagen wir, dass v ein *Dropwort* von w ist, wenn v aus w durch Weglassen von Buchstaben entsteht. Zum Beispiel sind $abc, bc, ac, ab, a, b, c, \varepsilon$ Dropwörter von abc. Das Wort ba is aber kein Dropwort von abc.

Formal sagen wir, dass v ein *Dropwort* von einem Wort $w = a_1 a_2 \dots a_k$ ist, wenn es eine Zahl l, mit $0 \leq l \leq k$, und Indizes $i_1 < i_2 < \dots < i_l$ aus $\{1, \dots, k\}$ gibt, so dass $v = a_{i_1} a_{i_2} \dots a_{i_l}$.

Sei Σ ein Alphabet und $L \subseteq \Sigma^*$ eine reguläre Sprache. Zeigen Sie, dass dann die Sprache

$$L_{\text{Drop}} := \{v \in \Sigma^* \mid \ v \text{ ist Dropwort eines Wortes } w \in L\}$$

auch regulär ist.

Es reicht, wenn Sie ausgehend von einem NFA, der L erkennt, einen endlichen Automaten konstruieren, der L_{Drop} erkennt. Geben Sie Ihre Konstruktion formal an, und erklären Sie kurz die Idee dahinter.

Lösung:

Sei L eine reguläre Sprache und $\mathcal{A} := (Q, \Sigma, \Delta, q_0, F)$ ein NFA, der L erkennt.

Wir definieren den ε-NFA $\mathcal{A}' := (Q, \Sigma, \Delta', q_0, F)$ durch

$$\Delta' := \Delta \cup \Delta_\varepsilon$$

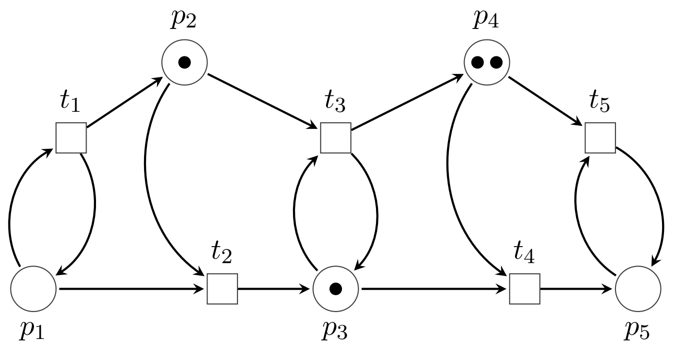
wobei

$$\Delta_\varepsilon := \big\{(p, \varepsilon, q) \mid \text{ Es gibt ein } a \in \Sigma \text{ mit } (p, a, q) \in \Delta \ \big\}.$$

Irgendeine vernünftige Erklärung reicht hier aus.

Aufgabe 9 **2+2+2+2=8 Punkte**

Betrachten Sie das folgende Petrinetz $\mathcal{N}$ mit der Markierung $M = (0, 1, 1, 2, 0)$:



Beantworten Sie folgende Fragen und geben Sie jeweils eine kurze Begründung an.

a) Ist in $(\mathcal{N}, M)$ eine Markierung M' erreichbar, so dass $(\mathcal{N}, M')$ verklemmt ist?

b) Ist $(\mathcal{N}, M)$ beschränkt?

c) Geben Sie die Transitionsmatrix D von $\mathcal{N}$ an.

d) Geben Sie einen Lauf von $(1, 0, 0, 0, 0)$ nach $(0, 0, 0, 0, 1)$ an.

Lösung:

a) Ja. Durch Schalten von t_4 und danach t_5 wird die Markierung $(0, 1, 0, 0, 1)$ erreicht. Diese Markierung ist verklemmt, da keine Transition schalten kann.

b) Ja, die einzigen erreichbaren Markierungen sind:

$(0, 1, 1, 2, 0)$, $(0, 1, 0, 1, 1)$, $(0, 1, 0, 0, 1)$, $(0, 0, 1, 3, 0)$, $(0, 0, 0, 2, 1)$, $(0, 0, 0, 1, 1)$, $(0, 0, 0, 0, 1)$.

Eine andere Begründung ist: t_1 wird von M aus nie feuern können, da p_1 nie gefüllt wird. Alle anderen Transitionen haben mindestens so viele Plätze im Vorbereich wie im Nachbereich. Also können nie mehr Marken erzeugt werden und somit ist $\mathcal{N}$ von M aus beschränkt.

c)

$$D := \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & -1 & 1 \\ 0 & 0 & 0 & -1 & 0 \end{pmatrix}$$

d) t_1, t_1, t_2, t_3, t_4