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

Hinweise:

- Die **Hausaufgaben** sollen in Gruppen von je 2 Studierenden aus dem gleichen Tutorium bearbeitet werden.
- Die Lösungen der Hausaufgaben müssen bis Fr., 16.07.2010, 11:45 in der Globalübung abgegeben werden. Alternativ ist es möglich, die Lösungen in den Tutorien am Mittwoch abzugeben oder bis Fr., 16.07.2010 11 Uhr im Abgabekasten im Flur des LuFG I2 einzuwerfen (Ahornstr. 55, E1, 2. Etage).
- Namen und Matrikelnummern der Studierenden sowie **die Nummer der Übungsgruppe** sind auf jedes Blatt der Abgabe zu schreiben. **Heften bzw. tackern Sie die Blätter**!
- Die **Tutoraufgaben** werden in den jeweiligen Tutorien gemeinsam besprochen und bearbeitet.
- Am **14.07.2010** finden die **letzten Tutorien** zur Vorlesung Formale Systeme, Sprachen, Prozesse statt. Dort werden Sie die Möglichkeit haben, unter Anleitung des Tutors verschiedene klausurrelevante Aufgaben aus dem Semester zu wiederholen.
- Am **8.7.2010** findet die **letzte Vorlesung** Formale Systeme, Sprachen, Prozesse im Sommersemester 2010 statt.

Hausaufgabe 1 (Induktion): (5 Punkte)

Gegeben sei ein NFA $M = (Q, \Sigma, \delta, q_0, F)$. Wir konstruieren damit die linkslineare Grammatik $G = (Q, \Sigma, P, q_o)$ mit den Produktionen $P = \{q \to ap \mid p \in \delta(q, a), q \in Q, a \in \Sigma\} \cup \{q \to \epsilon \mid q \in F\}$.
Beweisen Sie, dass für alle $p, q \in Q$, $w \in \Sigma^*$ gilt: $p \in \hat{\delta}(q, w) \Leftrightarrow q \Rightarrow_G^* wp$.[1]
Hinweis: Beweisen Sie beide Richtungen getrennt. Verwenden Sie für $\Rightarrow$ eine Induktion über die Wortlänge und für $\Leftarrow$ eine Induktion über die Ableitungslänge.

Lösung:

Wir zeigen zuerst $p \in \hat{\delta}(q, w) \Rightarrow q \Rightarrow_G^* wp$ ($\dagger$) per Induktion über die Länge des Wortes w.
Induktionsanfang: Sei $|w| = 0$, d.h. $w = \epsilon$. Sei $p \in \hat{\delta}(q, \epsilon)$. Dann gilt $p = q$ und die Ableitung $q \Rightarrow_G^0 q = p$ existiert.
Wir nehmen nun als Induktionshypothese an, dass für beliebige $n \in \mathbb{N}_0$ die Aussage ($\dagger$) für Wörter der Länge n bereits gilt.
Induktionsschritt: Sei $|w| = n + 1$ mit $w = w' \cdot a$, $w' \in \Sigma^n, a \in \Sigma$. Dann gibt es ein q' mit $p \in \delta(q', a)$ und $q' \in \hat{\delta}(q, w')$. Nach Induktionshypothese gilt $q \Rightarrow_G^* w'q'$. Nach Konstruktion gilt außerdem $q' \to ap \in P$. Damit können wir die Ableitung $q \Rightarrow_G^* w'q' \Rightarrow_{q' \to ap} w'ap = wp$ bilden.

Wir zeigen nun die Rückrichtung $q \Rightarrow_G^* wp \Rightarrow p \in \hat{\delta}(q, w)$ ($\ddagger$) per Induktion über die Länge n der Ableitung.
Induktionsanfang: Sei $n = 0$, d.h. $q \Rightarrow_g^0 q = \epsilon q$. Es gilt offensichtlich $q \in \hat{\delta}(q, \epsilon)$.
Wir nehmen nun als Induktionshypothese an, dass für beliebige $n \in \mathbb{N}_0$ die Aussage ($\ddagger$) für Ableitungen der Länge n bereits gilt.
Induktionsschritt: Nach Konstruktion von G gibt es nur linkslineare Regeln. Es gibt also ein $q' \in Q$ mit einer Regel $q' \to ap$ mit $q \Rightarrow_G^n w'q' \Rightarrow_{q' \to ap} w'ap = wp$. Nach Induktionshypothese gilt $q' \in \hat{\delta}(q, w')$ und nach Konstruktion der Regel $q' \to ap$ gilt auch $p \in \delta(q', a)$. Damit gilt dann direkt $p \in \hat{\delta}(q, w)$.

[1] Dies ist Lemma 4.0.4 aus der Vorlesung.

Tutoraufgabe 2 (Kontextsensitive Sprachen):

Um zu zeigen, dass L_i für $i \in \{1, 2\}$ eine kontextsensitive Sprache ist, geben Sie jeweils eine monotone Grammatik G_i an, so dass $L(G_i) = L_i$ gilt. Geben Sie außerdem eine Ableitung für das Wort w_i mit Ihrer Grammatik an.

a) $L_1 = \{a^n b^n c^n \mid n \in \mathbb{N}_0\}$ und $w_1 = aabbcc$.

b) $L_2 = \{a^n b^m c^n d^m \mid n, m \in \mathbb{N}_{>0}\}$ und $w_2 = aabbbccddd$.

Lösung:

a)

$$
\begin{aligned}
S &\to \epsilon && (1)\\
S &\to T && (2)\\
T &\to aTBC && (3)\\
T &\to aBC && (4)\\
CB &\to BC && (5)\\
aB &\to ab && (6)\\
bB &\to bb && (7)\\
bC &\to bc && (8)\\
cC &\to cc && (9)
\end{aligned}
$$

Hier werden mit Regeln (3) und (4) für jedes Terminalsymbol a jeweils gleich viele B und C erzeugt. Diese sind noch nicht notwendigerweise in der richtigen Ordnung, können aber mit Regel (5) Schritt für Schritt so geordnet werden, dass kein B mehr rechts von einem C steht. Nun kann man von den a-Symbolen auf der linken Seite ausgehend jeweils das erste Nonterminal in ein Terminal umwandeln.

Beispielableitung für $aabbcc$ (abgeleitete Teile sind jeweils unterstrichen):

$$
\begin{aligned}
&\underline{S}\\
\Rightarrow^{(2)}\, &\underline{T}\\
\Rightarrow^{(3)}\, &a\underline{T}BC\\
\Rightarrow^{(4)}\, &aaB\underline{CB}C\\
\Rightarrow^{(5)}\, &a\underline{aB}BCC\\
\Rightarrow^{(6)}\, &aa\underline{bB}CC\\
\Rightarrow^{(7)}\, &aab\underline{bC}C\\
\Rightarrow^{(8)}\, &aabb\underline{cC}\\
\Rightarrow^{(9)}\, &aabbcc
\end{aligned}
$$

b)

$$\begin{aligned} S &\to aNM && (1)\\ S &\to aM && (2)\\ N &\to aNY && (3)\\ N &\to aY && (4)\\ M &\to bMd && (5)\\ M &\to bXd && (6)\\ Yb &\to bY && (7)\\ YX &\to Xc && (8)\\ X &\to c && (9) \end{aligned}$$

Hier werden die Teillängen n, m durch die Anzahl der N- bzw. M-Produktionen festgelegt. Während die a-, b- und c-Terminale bereits beim Verwenden der N und M-Produktionen aufgebaut werden, wird das Nonterminal Y verwendet, um die Anzahl der noch zu erzeugenden c-Terminale zu zählen. Das Nonterminal X wird genau einmal erzeugt und markiert die Stelle, an der die c-Terminale eingefügt werden müssen.

Mit Hilfe von Produktion (7) werden dann die vor den bereits erzeugten b stehenden Y im Wort nach rechts verschoben, bis sie X erreichen. Dort werden sie dann durch ein c ersetzt. Zuletzt kann X durch ein einziges c ersetzt werden.

Beispielableitung für $aabbbccddd$ (abgeleitete Teile sind jeweils unterstrichen):

$$\begin{aligned} &\underline{S}\\ \Rightarrow^{(1)}\ &a\underline{N}M\\ \Rightarrow^{(4)}\ &aaY\underline{M}\\ \Rightarrow^{(5)}\ &aaYb\underline{M}d\\ \Rightarrow^{(5)}\ &aaYbb\underline{M}dd\\ \Rightarrow^{(6)}\ &aa\underline{Yb}bbXddd\\ \Rightarrow^{(7)}\ &aab\underline{Yb}bXddd\\ \Rightarrow^{(7)}\ &aabb\underline{Yb}Xddd\\ \Rightarrow^{(7)}\ &aabbb\underline{YX}ddd\\ \Rightarrow^{(8)}\ &aabbb\underline{X}cddd\\ \Rightarrow^{(9)}\ &aabbbccddd \end{aligned}$$

Hausaufgabe 3 (Kontextsensitive Sprachen): (3 + 3 + 3 = 9 Punkte)

Um zu zeigen, dass L_i für $i \in \{3, 4, 5\}$ eine kontextsensitive Sprache ist, geben Sie jeweils eine monotone Grammatik G_i an, so dass $L(G_i) = L_i$ gilt. Geben Sie außerdem eine Ableitung für das Wort w_i mit Ihrer Grammatik an.

a) $L_3 = \{w \in \{a, b, c\}^* \mid \sharp_a(w) = \sharp_b(w) = \sharp_c(w)\}$ und $w_3 = caabcb$.

b) $L_4 = \{(ab)^n(cb)^n(ac)^n \mid n \in \mathbb{N}_0\}$ und $w_4 = ababcbcbacac$.

c) $L_5 = \{a^n b^m c^k \mid n, m, k \in \mathbb{N}_{>0} \wedge n \leq m \leq k\}$ und $w_5 = aabbccc$.

Lösung:

a)

$$
\begin{aligned}
S &\to \epsilon & (1)\\
S &\to T & (2)\\
T &\to TT & (3)\\
T &\to ABC & (4)\\
AB &\to BA & (5)\\
AC &\to CA & (6)\\
BA &\to AB & (7)\\
BC &\to CB & (8)\\
CA &\to AC & (9)\\
CB &\to BC & (10)\\
A &\to a & (11)\\
B &\to b & (12)\\
C &\to c & (13)
\end{aligned}
$$

Hier werden mit den Regeln (3) und (4) eine beliebige Anzahl von ABC-Gruppen erzeugt. Mit Hilfe der Regeln (5)-(10) können diese dann frei vertauscht werden. Mit den Regeln (11)-(13) werden die Nonterminale dann in die entsprechenden Terminale übersetzt.

Beispielableitung für $caabcb$ (abgeleitete Teile sind jeweils unterstrichen):

$$
\begin{aligned}
& \underline{S}\\
\Rightarrow^{(2)} & \underline{T}\\
\Rightarrow^{(3)} & \underline{T}T\\
\Rightarrow^{(4)} & ABC\underline{T}\\
\Rightarrow^{(4)} & A\underline{BC}ABC\\
\Rightarrow^{(8)} & \underline{AC}BABC\\
\Rightarrow^{(6)} & CA\underline{BA}BC\\
\Rightarrow^{(7)} & CAAB\underline{BC}\\
\Rightarrow^{(8)} & \underline{CAABCB}\\
\Rightarrow^{*} & \ caabcb
\end{aligned}
$$

b)

$$\begin{aligned} S &\to \epsilon & (1)\\ S &\to T & (2)\\ T &\to abTHK & (3)\\ T &\to abHK & (4)\\ KH &\to HK & (5)\\ abH &\to abcb & (6)\\ cbH &\to cbcb & (7)\\ cbK &\to cbac & (8)\\ acK &\to acac & (9) \end{aligned}$$

Die Grammatik ist bis auf die Terminalsymbole identisch zur Grammatik für $L_1 = \{a^n b^n c^n \mid n \in \mathbb{N}_0\}$. Beispielableitung für $ababcbcbacac$ (abgeleitete Teile sind jeweils unterstrichen):

$$\begin{aligned} &\underline{S}\\ \Rightarrow^{(2)}&\underline{T}\\ \Rightarrow^{(3)}&ab\underline{T}HK\\ \Rightarrow^{(4)}&abab H\underline{KH}K\\ \Rightarrow^{(5)}&ab\underline{abH}HKK\\ \Rightarrow^{(6)}&abab\underline{cbH}KK\\ \Rightarrow^{(7)}&ababcb\underline{cbK}K\\ \Rightarrow^{(8)}&ababcbcb\underline{acK}\\ \Rightarrow^{(9)}&ababcbcbacac \end{aligned}$$

c)

$$\begin{aligned} S &\to aTbX & (1)\\ S &\to abX & (2)\\ T &\to aTbC & (3)\\ T &\to TbC & (4)\\ T &\to TC & (5)\\ T &\to bC & (6)\\ T &\to C & (7)\\ Cb &\to bC & (8)\\ CX &\to Xc & (9)\\ X &\to c & (10) \end{aligned}$$

Hier werden a- und b-Symbole erneut beim initialen Aufbau mit Hilfe der Produktionen für S und G generiert. Mit X wird die Stelle im Wort markiert, in der im Folgenden die c-Symbole erzeugt werden müssen, deren Zahl von der Anzahl der erzeugten C-Nonterminale abhängt. Diese werden mitten im Wort erzeugt und können dann mit Regel 8 nach rechts bewegt werden.

Beispielableitung für $aabbccc$ (abgeleitete Teile sind jeweils unterstrichen):

$$\begin{aligned}
&\underline{S}\\
\Rightarrow^{(1)}\ & a\underline{T}bX\\
\Rightarrow^{(3)}\ & aa\underline{T}bCbX\\
\Rightarrow^{(4)}\ & aaCb\underline{Cb}X\\
\Rightarrow^{(8)}\ & aa\underline{Cb}bCX\\
\Rightarrow^{(8)}\ & aab\underline{Cb}CX\\
\Rightarrow^{(8)}\ & aabbC\underline{CX}\\
\Rightarrow^{(9)}\ & aabb\underline{CX}c\\
\Rightarrow^{(9)}\ & aabb\underline{X}cc\\
\Rightarrow^{(10)}\ & aabbccc
\end{aligned}$$

Tutoraufgabe 4 (Synchronisiertes Produkt):

```
while (true) {
    x := 1;
    /* wait for x != 1 */
    while (x = 1);
    print ('a');
}
```

Programm P_1

```
x := 0;
/* wait for x != 0 */
while (x = 0);
print ('b');
```

Programm P_2

a) Modellieren Sie den Kontrollfluss von P_1 durch einen NFA M_1 mit 3 Zuständen, sowie den den Kontrollfluss von P_2 durch einen NFA M_2 mit 4 Zuständen. Verwenden Sie hierbei für beide Automaten die Signatur $\Sigma = \{\texttt{x = 0?}, \texttt{x = 1?}, \texttt{x := 0}, \texttt{x := 1}, \texttt{print('a')}, \texttt{print('b')}\}$.

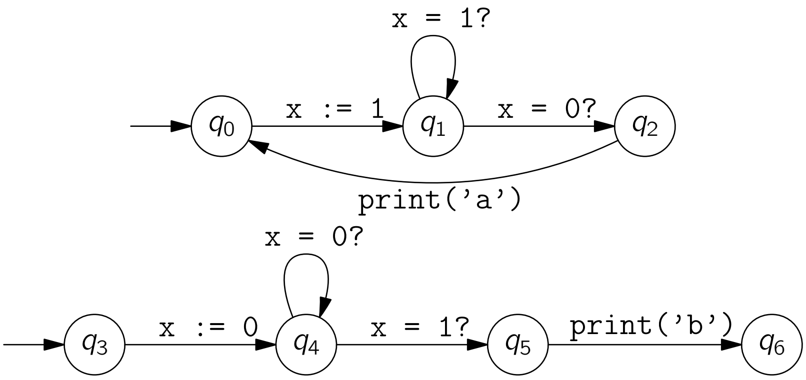
b) Berechnen Sie das unsynchronisierte Produkt $M_1 \sqcup M_2$. Geben Sie den resultierenden NFA an.

c) Modellieren Sie die boolesche Variable `x` des Programms durch den NFA B_x. Berechnen Sie den NFA $M := (M_1 \sqcup M_2) \circ B_x$ und geben Sie die Automaten B_x und M an.

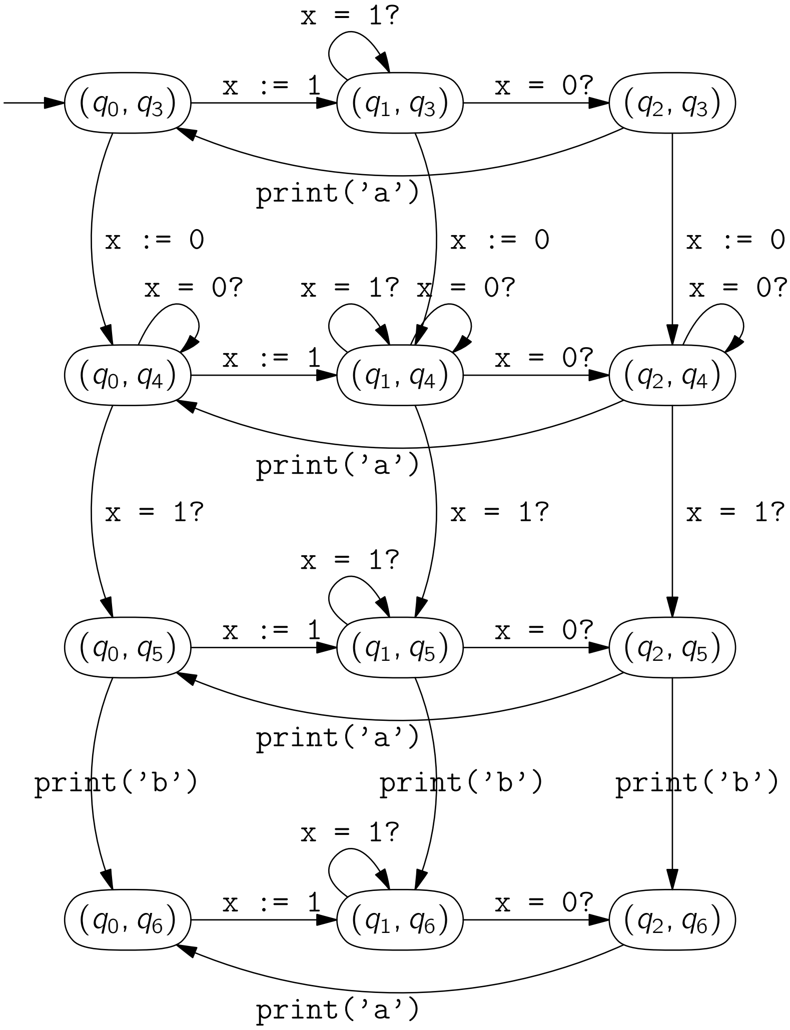
d) Ist es möglich, dass Programm P_1 und P_2 (wenn sie gleichzeitig ausgeführt werden) die Zeichenfolge `aba` ausgeben? Begründen Sie ihre Antwort mit dem NFA M.

Lösung:

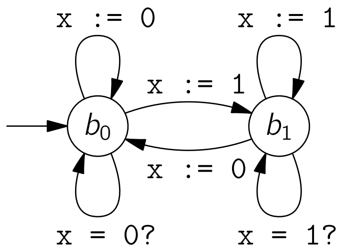
a)

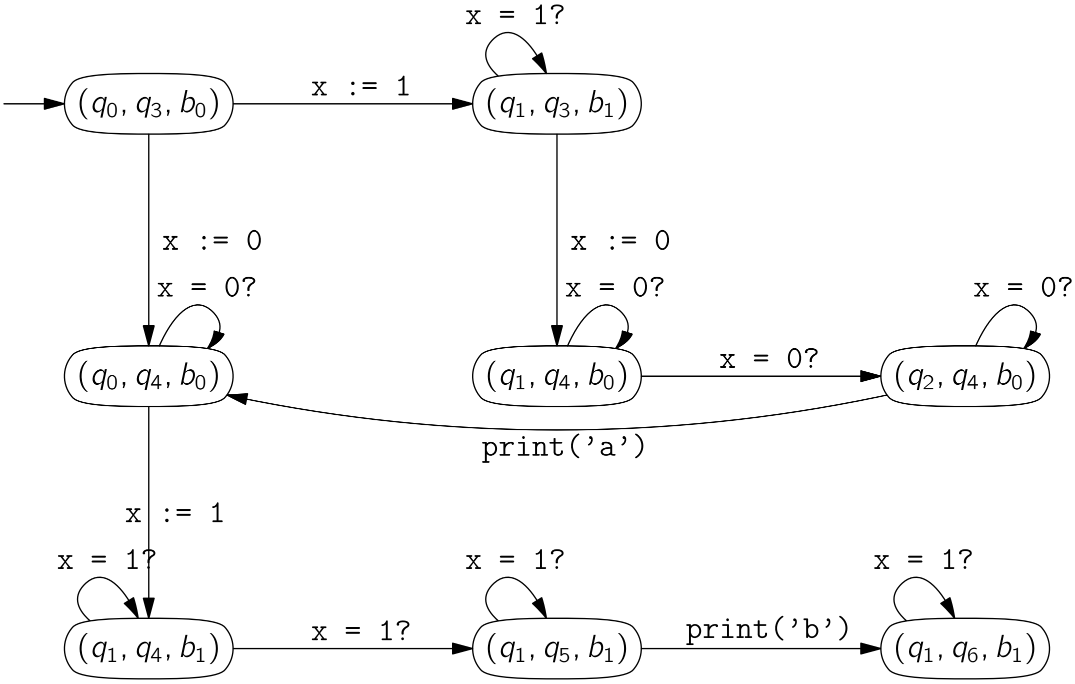

x = 1?
q_0
x := 1
q_1
x = 0?
q_2
print('a')
x = 0?
q_3
x := 0
q_4
x = 1?
q_5
print('b')
q_6

b)


x = 1?
(q_0, q_3)
x := 1
(q_1, q_3)
x = 0?
(q_2, q_3)
print('a')
x := 0
x := 0
x := 0
x = 0?
x = 1?
x = 0?
x = 0?
(q_0, q_4)
x := 1
(q_1, q_4)
x = 0?
(q_2, q_4)
print('a')
x = 1?
x = 1?
x = 1?
x = 1?
(q_0, q_5)
x := 1
(q_1, q_5)
x = 0?
(q_2, q_5)
print('a')
print('b')
print('b')
print('b')
x = 1?
(q_0, q_6)
x := 1
(q_1, q_6)
x = 0?
(q_2, q_6)
print('a')

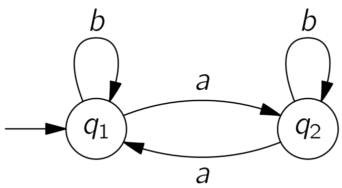
c)

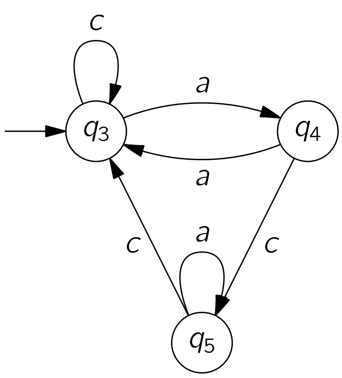




d) Nein, der NFA M beschreibt keinen Pfad, auf dem nach einem `print('b')` noch ein `print('a')` folgen kann.

Hausaufgabe 5 (Synchronisiertes Produkt): (4 Punkte)





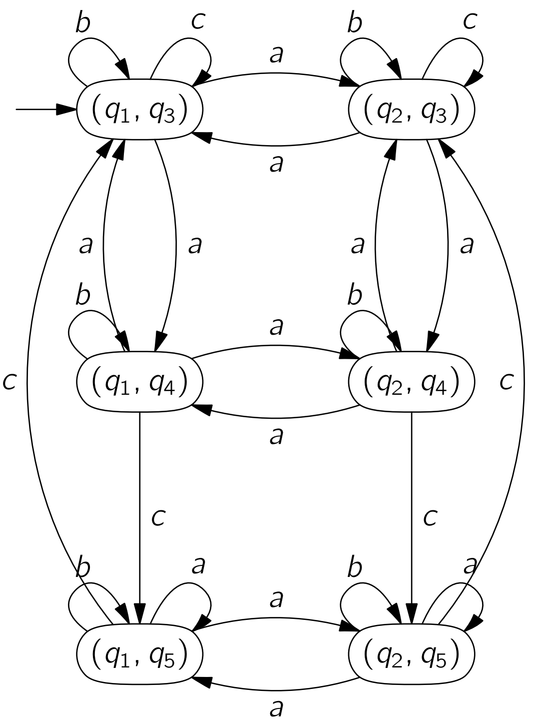
Gegeben seien die DFAs M_1 und M_2. Berechnen Sie:

a) $M_1 \sqcup M_2$

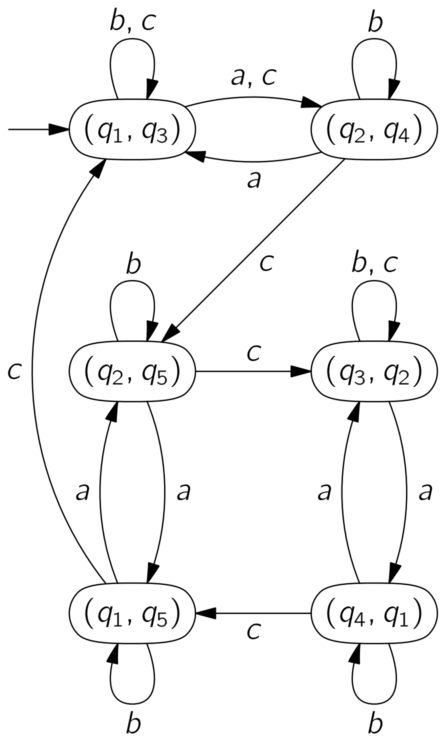
b) $M_1 \circ M_2$

Lösung:

a)



b)



Tutoraufgabe 6 (Petrinetze):

In dieser Aufgabe sollen Sie Petrinetze nutzen, um Prozesse aus dem Alltag zu modellieren. Verwenden Sie die unterstrichenen Begriffe als Stellen und Transitionen und wählen Sie geeignete Kanten zwischen diesen, um die im Aufgabentext dokumentierten Zusammenhänge darzustellen.

a) Wir betrachten ein Callcenter. Dort gibt es <u>Agenten</u>, die <u>Anrufe entgegennehmen</u> und den berichteten <u>Problemfall</u> im internen Ticketsystem registrieren. Im Folgenden versuchen die Agenten nun, das Problem <u>direkt zu lösen</u>, z.B. mit Hilfe eines Fragenkataloges und Standardantworten. Gelingt dies, ist der Problemfall abgeschlossen und der Agent steht für einen weiteren Anruf zur Verfügung. In allen anderen Fällen wird der Problemfall <u>eskaliert</u> und im Ticketsystem als <u>ungelöstes Problem</u> markiert, damit ein Techniker den Fall analysieren kann.

Erstellen Sie nun ein Petrinetz, das diesen Prozess dokumentiert.

b) Wir betrachten nun die Technik-Abteilung eines Software-Unternehmens. In einem internen System werden berichtete <u>Probleme</u> verwaltet. Hat ein <u>Techniker</u> gerade keine andere Aufgabe, kann eines dieser Probleme <u>analysiert</u> werden. Dabei wird ein minimaler <u>Testfall</u> erstellt, der das Problem reproduziert. Danach kann sich der Techniker wieder anderen Aufgaben zuwenden.

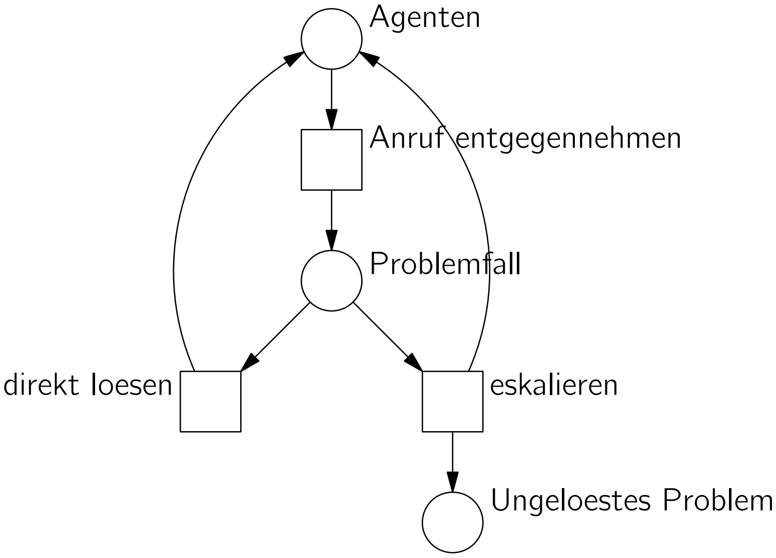
Mit Hilfe des Testfalls kann ein Techniker diesen <u>lösen</u>. Um zu verhindern, dass das Problem wieder eingeführt wird, wird der Testfall als <u>Regressionstest</u> gespeichert.

Erstellen Sie nun ein Petrinetz, das diesen Prozess dokumentiert.

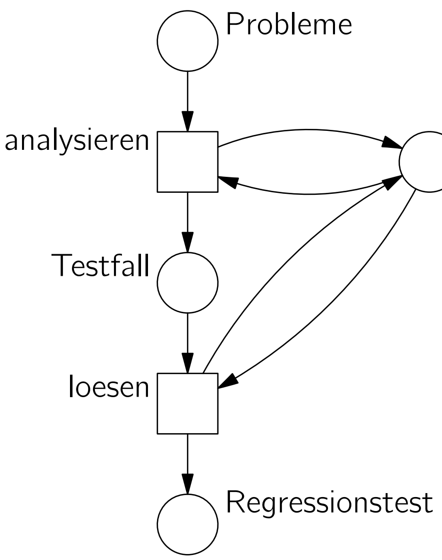
c) Erklären Sie, wie Sie die beiden Petrinetze sinnvoll miteinander verbinden können. Beschreiben Sie, welche Komponenten sich in den beiden Netzen entsprechen und welchen Prozess das Ergebnis modelliert.

Lösung:

a)



b)



c) Die Stellen „Ungeloest“ im Callcenter-Petrinetz und „Probleme“ im Petrinetz für die Technik-Abteilung entsprechen sich. Die Komposition der beiden Netze an dieser Stelle entspricht dann der Übergabe eines Problems an einen Techniker.

Hausaufgabe 7 (Petrinetze): (3 + 2 = 5 Punkte)

In dieser Aufgabe sollen Sie Petrinetze nutzen, um Prozesse aus dem Alltag zu modellieren. Verwenden Sie die unterstrichenen Begriffe als Stellen und Transitionen und wählen Sie geeignete Kanten zwischen diesen, um die im Aufgabentext dokumentierten Zusammenhänge darzustellen.

a) Wir betrachten den Übungsbetrieb einer Vorlesung "Formale Sprachen, Automaten und Petrinetze". Hier werden <u>Übungsblätter</u> bereitgestellt, die dann von <u>Studierenden</u> <u>bearbeitet</u> werden. Die so entstehenden <u>Übungsabgaben</u> können dann entweder in einem <u>Tutorium abgegeben</u> oder in den <u>Abgabekasten</u> <u>geworfen</u> werden. Abgaben im Tutorium landen direkt auf dem <u>Korrekturstapel</u> eines Tutors. Abgaben im Übungskasten werden von einem <u>Assistenten</u> <u>sortiert</u> (wenn er dafür Zeit hat) und kommen dann auf den Korrekturstapel eines Tutors.

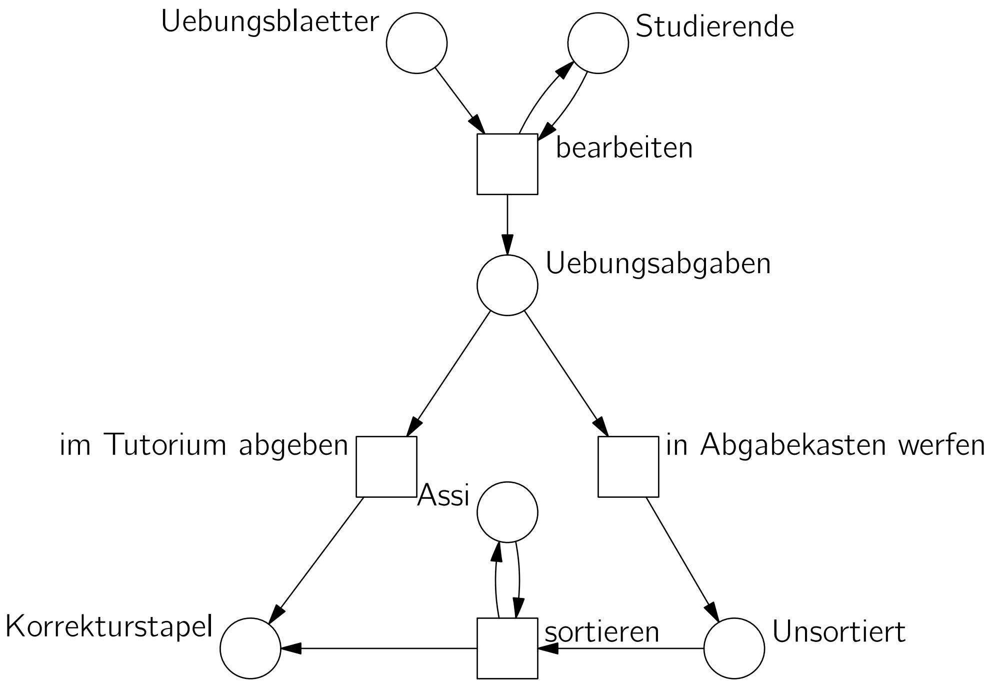
Erstellen Sie nun ein Petrinetz, das diesen Prozess dokumentiert.

b) Wir betrachten nun die Arbeit von <u>Tutoren</u>. Übungsabgaben vom <u>Korrekturstapel</u> werden <u>korrigiert</u>, wenn der Tutor gerade keine anderen Verpflichtungen hat. Bei der Korrektur wird ein <u>Rückgabestapel</u> aufgebaut. Gleichzeitig entsteht eine <u>Punkteliste</u>. Die korrigierten Übungsabgaben vom Rückgabestapel werden <u>zurückgeben</u>, wenn ein Tutor gerade nicht anderweitig beschäftigt ist, meist im Tutorium. Außerdem werden die Punkte von der Punkteliste <u>eingetragen</u>, wenn der Tutor Zeit hat.

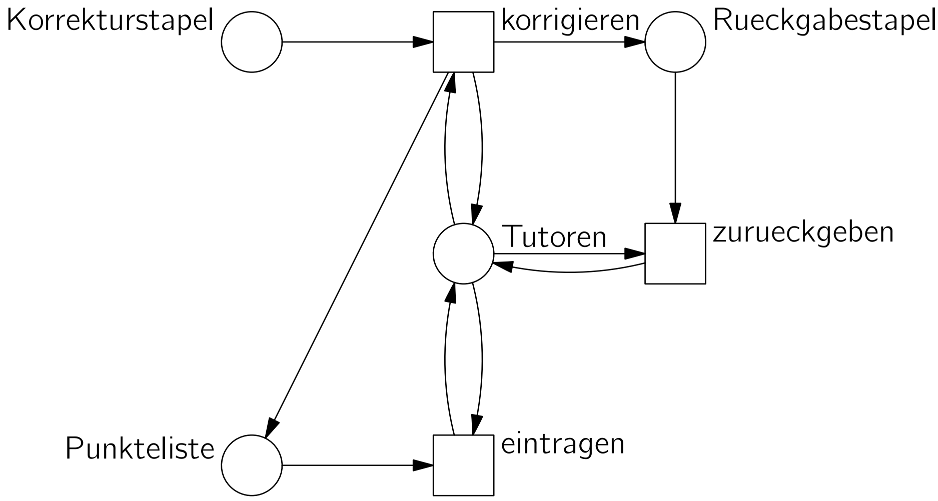
Erstellen Sie nun ein Petrinetz, das diesen Prozess dokumentiert.

Lösung:

a)

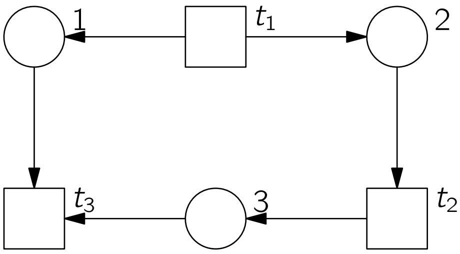


b)



Tutoraufgabe 8 (Erreichbarkeit):

Gegeben sei folgendes Petrinetz P:



a) Berechnen Sie die Inzidenzmatrix $D = D^+ + D^-$.

b) Zeigen oder widerlegen Sie, dass folgende Ableitungsequenzen existieren. Geben Sie im Falle der Existenz die Folge der ausgelösten Transitionen an und im Falle der Nichtexistenz einen Beweis mit Hilfe von Satz 5.2.5.

Hinweis: Verwenden Sie zum Konstruieren einer Folge für die Ableitung $m \longrightarrow^* m'$ eine Lösung x der Gleichung $m' = m + x \cdot D$.

- $(0, 0, 0) \longrightarrow^* (3, 0, 2)$
- $(0, 0, 0) \longrightarrow^* (3, 2, 1)$

Lösung:

a)

$$D^+ = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

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

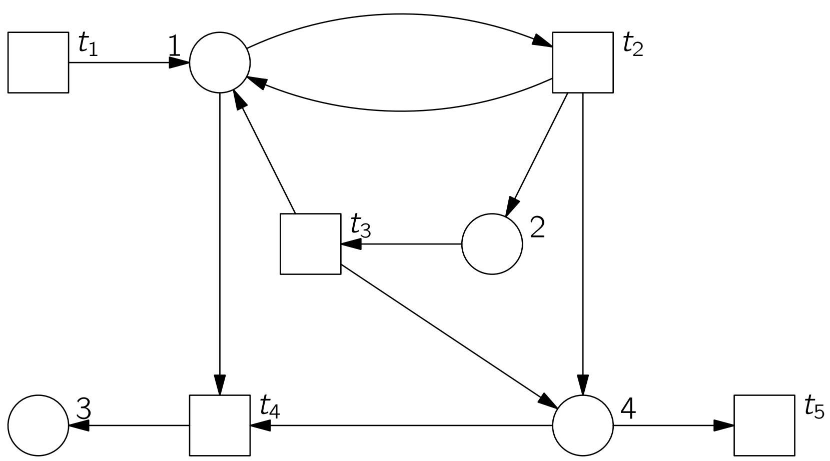
$$D = \begin{pmatrix} 1 & 1 & 0 \\ 0 & -1 & 1 \\ -1 & 0 & -1 \end{pmatrix}$$

b)
- Nach Satz 5.2.5 der Vorlesung, existiert eine solche Transitionenfolge nicht, da $(3, 0, 2) = \vec{x} \cdot D$ keine Lösung hat.
- Wir suchen eine Lösung für x in $D^T\vec{x} = (3, 2, 1)^T$. Dies ist der Fall für $\vec{x} = (3, 1, 0)^T$ (d.h. drei mal die Transition t_1 und ein mal die Transition t_2). Wir suchen nun eine Reihenfolge dieser Transitionen, in der sie anwendbar sind:

$$\begin{aligned} (0, 0, 0) &\xrightarrow{t_1} (1, 1, 0) \\ &\xrightarrow{t_2} (1, 0, 1) \\ &\xrightarrow{t_1} (2, 1, 1) \\ &\xrightarrow{t_1} (3, 2, 1) \end{aligned}$$

Hausaufgabe 9 (Erreichbarkeit): (2 + 2 = 4 Punkte)

Gegeben sei folgendes Petrinetz P:



a) Berechnen Sie die Inzidenzmatrix $D = D^+ + D^-$.

b) Zeigen oder widerlegen Sie, dass folgende Ableitungsequenzen existieren. Geben Sie im Falle der Existenz die Folge der ausgelösten Transitionen an und im Falle der Nichtexistenz einen Beweis mit Hilfe von Satz 5.2.5.

Hinweis: Verwenden Sie zum Konstruieren einer Folge für die Ableitung $m \longrightarrow^* m'$ eine Lösung x der Gleichung $m' = m + x \cdot D$.

- $(0, 0, 0, 0) \longrightarrow^* (1, 1, 1, 1)$
- $(0, 0, 0, 0) \longrightarrow^* (1, 2, 3, 4)$

Lösung:

a)

$$D^+ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

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

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

b)

- Wir suchen eine Lösung für x in $D^T\vec{x} = (1,1,1,1)^T$. Dies ist der Fall für $\vec{x} = (1,2,1,1)^T$. Wir suchen nun eine Reihenfolge dieser Transitionen, in der sie anwendbar sind:

$$\begin{aligned} (0,0,0,0) &\xrightarrow{t_1} (1,0,0,0) \\ &\xrightarrow{t_2} (1,1,0,1) \\ &\xrightarrow{t_4} (0,1,1,0) \\ &\xrightarrow{t_3} (1,0,1,1) \\ &\xrightarrow{t_2} (1,1,1,2) \\ &\xrightarrow{t_5} (1,1,1,1) \end{aligned}$$

- Wir suchen eine Lösung für x in $D^T\vec{x} = (3,2,1)^T$. Dies ist der Fall für $\vec{x} = (1,5,3,3,1)^T$. Wir suchen nun eine Reihenfolge dieser Transitionen, in der sie anwendbar sind:

$$\begin{aligned}
(0,0,0,0) &\xrightarrow{t_1} (1,0,0,0)\\
&\xrightarrow{t_2} (1,1,0,1)\\
&\xrightarrow{t_2} (1,2,0,2)\\
&\xrightarrow{t_2} (1,3,0,3)\\
&\xrightarrow{t_2} (1,4,0,4)\\
&\xrightarrow{t_2} (1,5,0,5)\\
&\xrightarrow{t_3} (2,4,0,6)\\
&\xrightarrow{t_3} (3,3,0,7)\\
&\xrightarrow{t_3} (4,2,0,8)\\
&\xrightarrow{t_4} (3,2,1,7)\\
&\xrightarrow{t_4} (2,2,2,6)\\
&\xrightarrow{t_4} (1,2,3,5)\\
&\xrightarrow{t_5} (1,2,3,4)
\end{aligned}$$