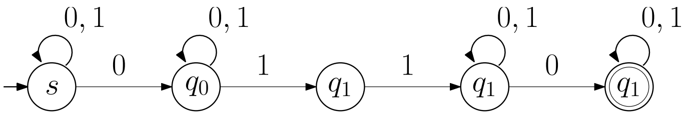
Musterlösung Informatik-III-Nachklausur

Aufgabe 1

(2+2+4+4 Punkte)

(a) $L = (0 \cup 1)^*0(0 \cup 1)^*11(0 \cup 1)^*0(0 \cup 1)^*$

(b) Der Automat ist durch folgendes Übergangsdiagramm gegeben:



(c) Man erhält die Äquivalenzklassen

$$\begin{aligned}
[\varepsilon] &= \{w \in \Sigma^* \mid |w|_0 = 0\} \\
[0] &= \{w \in \Sigma^* \mid |w|_0 = 1\} \\
[00] &= \{w \in \Sigma^* \mid |w|_0 = 2\} \\
[000] &= \{w \in \Sigma^* \mid |w|_0 > 2\}
\end{aligned}$$

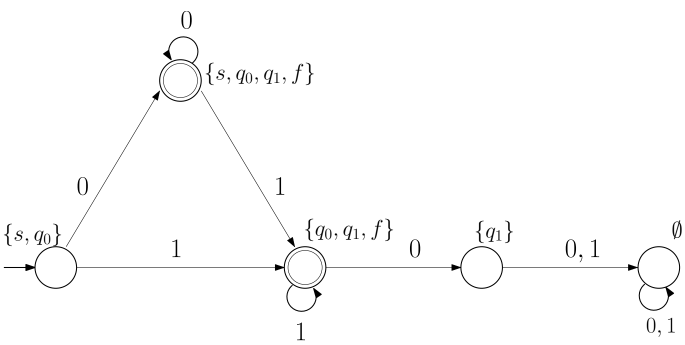
dabei bezeichnet $|w|_0$ die Anzahl der Nullen in w.

(d) Sei $\mathcal{A} := (Q, \Sigma, \delta, s, F)$ der gegebene NEA. Potenzmengenkonstruktion des zugehörigen DEA $\mathcal{A}' := (Q', \Sigma, \delta', s', F')$ ergibt:

- $Q' = \{\{s, q_0\}, \{s, q_0, q_1, f\}, \{q_0, q_1, f\}, \{q_1\}, \emptyset\}$ bzw. $Q' = 2^Q$
- $s' = E(s) = \{s, q_0\}$
- δ' ergibt sich wie folgt

	0	1
$\{s, q_0\}$	$\{s, q_0, q_1, f\}$	$\{q_0, q_1, f\}$
$\{s, q_0, q_1, f\}$	$\{s, q_0, q_1, f\}$	$\{q_0, q_1, f\}$
$\{q_0, q_1, f\}$	$\{q_1\}$	$\{q_0, q_1, f\}$
$\{q_1\}$	$\emptyset$	$\emptyset$
$\emptyset$	$\emptyset$	$\emptyset$

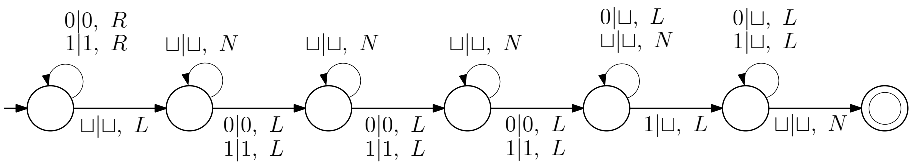
- $F' = \{q' \in Q' \mid q' \cap F \neq \emptyset\} = \{\{s, q_0, q_1, f\}, \{q_0, q_1, f\}\}$



Aufgabe 2 (2+4+6 Punkte)

(a) L_1 ist genau das Komplement von L_2. Aus Vorlesung (und Übung) ist bekannt, dass eine Sprache L bereits entscheidbar ist, wenn sowohl L als auch das Komplement von L semi-entscheidbar sind. Aus der Semi-Entscheidbarkeit von L_1 würde also unmittelbar die Entscheidbarkeit von L_2 folgen, was im Widerspruch zur Voraussetzung steht.

(b) Die TM ist gegeben durch



(c) Die ersten 9 Schritte der Verarbeitung des Wortes 0100 sind gegeben durch die folgenden Konfigurationen:

	(s)	0100	(1)
$\sqcup$	(q_0)	100	(2)
1	(q_0)	00	(3)
10	(q_0)	0	(4)
100	(q_0)	$\sqcup$	(5)
10	(q_2)	0	(6)
1	(q_4)	$0\,\sqcup$	(7)
$\sqcup$	(q_4)	$10\,\sqcup$	(8)
$\sqcup$	(q_4)	$\sqcup 10\,\sqcup$	(9)
$\sqcup$	(s)	$10\sqcup$	(10)

Die Turingmaschine $\mathcal{M}$ akzeptiert die Sprache der Palindrome über dem Alphabet $\Sigma = \{0, 1\}$, also

$$L = \{w \in \{0,1\}^* \;\mid\; w = w^R\}\ .$$

Im Zustand s liest $\mathcal{M}$ das am weitesten links stehende Zeichen ein und merkt sich dieses über die Zustände q_0/q_1. Außerdem wird das Zeichen gelöscht. Dann geht die Maschine zu dem am weitesten rechts stehenden Zeichen und vergleicht dieses im Zustand q_2/q_3 mit dem gemerkten Zeichen. Falls die Zeichen unterschiedlich sind bricht $\mathcal{M}$ nichtakzeptierend ab. Ansonsten löscht $\mathcal{M}$ das Zeichen, geht zum am weitesten links stehenden Zeichen und beginnt von vorne.

$\mathcal{M}$ stoppt akzeptierend, falls im Zustand s nur noch 0 oder 1 Zeichen auf dem Band stehen.

Aufgabe 3

(1+1+5+5 Punkte)

(a) $NPI = NP \setminus (P \cup NPC)$

(b) $S = \{\{1,3,5\},\{2\},\{4,6\}\}$

(c) (IS) liegt in NP, da sich in Polynomialzeit für eine geratene Knotenmenge M für jede Kante überprüfen lässt, an wievielen Knoten aus M sie anliegt.

Um zu Zeigen, dass sich jedes Problem aus NP polynomiell auf (IS) reduzieren lässt, reduzieren wir das NP-vollständige Problem (VCD) auf (IS).

Zu einer Instanz $I = (G = (V,E),k)$ von (VCD) konstruieren wir die Instanz $I' = (G = (V,E), n-k)$ von IS. Mit der Beobachtung gilt, dass I genau dann lösbar ist, wenn I' lösbar ist. Offensichtlich ist die Transformation polynomial.

Die Beobachtung gilt wegen folgender Argumentation: Sei S ein IS. Dann gilt für alle $(u,v) \in E : u \notin S \vee v \notin S$. Damit gilt für alle $(u,v) \in E : u \in V \setminus S \vee v \in V \setminus S$ und damit $V \setminus S$ ist ein VC.

(d) Angenommen es gibt einen Algorithmus A, der für jede Instanz I von VCO in polynomieller Zeit eine Lösung $apx(I)$ liefert mit $|apx(I) - opt(I)| < \epsilon$ für ein festes ϵ (oBdA $\epsilon \in \mathbb{N}$).

Zu einer Instanz $I = (G = (V,E),k)$ von VCD konstruieren wir eine Instanz G' von VCO indem wir den Graphen G $2\epsilon + 1$ mal kopieren. Dann gilt:

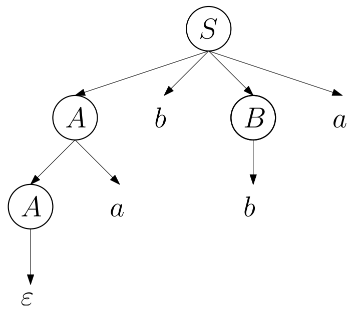
$$\begin{aligned}
|apx(G') - opt(G')| &< \epsilon \\
|apx(G') - (2\epsilon+1)opt(G)| &< \epsilon \\
\left|\frac{apx(G')}{(2\epsilon+1)} - opt(G)\right| &< \frac{\epsilon}{(2\epsilon+1)} < \frac{1}{2}
\end{aligned}$$

wenn man also $apx(G')/(2\epsilon+1)$ rundet hat man in Polynomialzeit die optimale Lösung von VCO angewandt auf G gefunden. Man muss nur noch überprüfen ob $opt(G)$ kleiner als k ist und hat damit in Polynomialzeit VC gelöst im Widerspruch zur Annahme $P \neq NP$.

Aufgabe 4

(Aufgabe 4 Punkte)

(a) Syntaxbaum für *abba*



Der Syntaxbaum ist eindeutig: An der Wurzel des Baumes muss die Regel $S \to AbBa$ angewendet werden, da es keine weitere Regel mit linker Seite S gibt. Um das Wort *abba* erzeugen zu können, muss das A zu a und B zu b abgeleitet werden. Dies ist nur möglich durch Anwendung der Regeln $A \to Aa$, gefolgt von $A \to \varepsilon$, und $B \to b$.

(b) Ursrüngliche Grammatik

$$\begin{array}{rcl} R := \{\ S & \to & AbBa, \\ A & \to & Aa \mid \varepsilon, \\ B & \to & S \mid b\ \} \end{array}$$

1. Schritt: Alle rechten Seiten haben die Form α mit $\alpha \in \Sigma$ oder $\alpha \in V^*$

$$\begin{array}{rcl} R := \{\ S & \to & AY_bBY_a, \\ A & \to & AY_a \mid \varepsilon, \\ B & \to & S \mid b, \\ Y_a & \to & a, \\ Y_b & \to & b\ \} \end{array}$$

2. Schritt: Alle rechten Seiten haben Länge ≤ 2

$$\begin{array}{rcl} R := \{\ S & \to & AC_1, \\ A & \to & AY_a \mid \varepsilon, \\ B & \to & S \mid b, \\ Y_a & \to & a, \\ Y_b & \to & b, \\ C_1 & \to & Y_bC_2, \\ C_2 & \to & BY_a\ \} \end{array}$$

3. Schritt: Elimination von ε-Produktionen

$$\begin{array}{rcl} R := \{\ S & \rightarrow & AC_1 \mid C_1, \\ A & \rightarrow & AY_a \mid Y_a, \\ B & \rightarrow & S \mid b, \\ Y_a & \rightarrow & a, \\ Y_b & \rightarrow & b, \\ C_1 & \rightarrow & Y_bC_2, \\ C_2 & \rightarrow & BY_a\ \} \end{array}$$

4. Schritt: Elimination von Kettenregeln

$$\begin{array}{rcl} R := \{\ S & \rightarrow & AC_1 \mid Y_bC_2, \\ A & \rightarrow & AY_a \mid a, \\ B & \rightarrow & AC_1 \mid Y_bC_2 \mid b, \\ Y_a & \rightarrow & a, \\ Y_b & \rightarrow & b, \\ C_1 & \rightarrow & Y_bC_2, \\ C_2 & \rightarrow & BY_a\ \} \end{array}$$

(c)

$$\begin{array}{cccc} \{S\} & & & \\ \{A,B\} & \{A\} & & \\ \{S\} & \{S\} & \emptyset & \\ \{B\} & \{A\} & \{B\} & \{B\} \\ \hline\hline b & a & b & b \end{array}$$

Das Wort *babb* ist in $L(G)$ enthalten, weil S in der Menge enthalten ist, die vom Algorithmus im letzten Schritt berechnet wird.

(d) Für $n \in \mathbb{N}$ sei $w_n := a^n b^n \# a^n$. Sei $w_n = uvwxy$ mit $|vx| \geq 1$ und $|vwx| \leq n$. Annahme: $uv^iwx^iy \in L$ für alle $i \in \mathbb{N}$.

Falls vx das Zeichen $\#$ enthält, gilt $uv^0wx^0y \notin L$, da jedes Wort aus L dieses Zeichen enthalten muss. Wegen $|vwx| \leq n$ gilt zusätzlich $vx = a^ib^j$ oder $vx = b^ia^j$ mit $i + j \geq 1$.

Falls $vx = a^ib^j$ mit $i + j \geq 1 \Rightarrow uv^0wx^0y = a^{n-i}b^{n-j}\#a^n$ mit $n - i + n - j < 2n$. Damit ist die Länge des hinteren Teilwortes länger als die Hälfte des vorderen Teilwortes und es gilt $uv^0wx^0y \notin L$ im Widerspruch zur Annahme.

Falls $vx = b^ia^j$ mit $i + j \geq 1 \Rightarrow uv^2wx^2y = a^nb^{n+i}\#a^{n+j}$. Falls $j > 0$ ist, ist $uv^2wx^2y \notin L$, da das vordere Teilwort nur n as enthält und das hintere mindestens $n + 1$. Also muss $j = 0$ sein und damit $i \geq 1$. In diesem Fall müßte das hintere Teilwort aber mindestens eine Länge von $n+1$ haben. Damit führt auch dieser Fall in einen Widerspruch und somit $uv^2wx^2y \notin L$ im Widerspruch zur Annahme.

Somit ist eine Zerlegung nach dem Pumping-Lemma nicht möglich, die Sprache kann also nicht kontextfrei sein.

Aufgabe 5 (12 Punkte)

Kreuzen Sie für folgende Aussagen an, ob diese wahr oder falsch sind.
Hinweis: Für jede richtige Antwort gibt es einen Punkt, für jede falsche Antwort wird ein Punkt abgezogen. Es wird keine negative Gesamtpunktzahl für diese Aufgabe geben.

	Wahr	Falsch
Sei L eine Sprache, die nicht regulär ist. Dann existiert zu jedem $n \in \mathbb{N}$ ein Wort $w \in L$ mit $\|w\| \geq n$, so dass für jede Zerlegung $w = uvx$ mit $\|uv\| \leq n$ und $\|v\| > 0$ ein $i \in \mathbb{N}_0$ existiert, so dass $uv^ix \notin L$.		X
Für jede Sprache vom Typ Chomsky 3, die nicht das leere Wort erzeugt, gibt es eine Grammatik in Greibach Normalform.	X	
Das Komplement einer entscheidbaren Sprache ist semi-entscheidbar.	X	
Falls $\mathcal{P} \neq \mathcal{NP}$, gilt $\mathcal{P} \neq co - \mathcal{P}$.		X
Sei L eine Sprache mit Nerode-Index $k < \infty$. Dann gibt es einen DEA mit k Zuständen der genau L akzeptiert.	X	
Es gibt einen deterministischen endlichen Automaten $\mathcal{A}$ mit der Eigenschaft, dass das Komplement der von $\mathcal{A}$ akzeptierten Sprache semi-entscheidbar aber nicht entscheidbar ist.		X
Wenn $\mathcal{P} = \mathcal{NP}$, dann ist jedes Problem aus $\mathcal{P} \setminus \{\emptyset, \Sigma^*\}$ $\mathcal{NP}$-vollständig.	X	
Sei L $\mathcal{NP}$-vollständig. Dann gibt es keine deterministische Turingmaschine, die genau L akzeptiert.		X
Das Problem SAT ist entscheidbar.	X	
Sei L eine reguläre Sprache. Dann gibt es eine Turingmaschine $\mathcal{M}$, die genau L akzeptiert und für jede Eingabe w zu jedem Zeitpunkt der Berechnung höchstens $\|w\|$ Speicherzellen auf dem Band belegt.	X	
Seien L_1 und L_2 Sprachen für die es je einen Kellerautomaten gibt, der genau sie akzeptiert. Dann gibt es einen Kellerautomaten der $L_1 \cap L_2$ akzeptiert.		X
Es gibt eine deterministische Turingmaschine $\mathcal{M}$, die zu jedem deterministischen endlichen Automaten $\mathcal{A}$ einen zu $\mathcal{A}$ äquivalenten deterministischen endlichen Automaten $\mathcal{A}'$ mit minimaler Anzahl an Zuständen berechnet.	X	