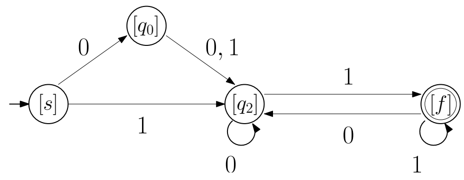
Musterlösung Informatik-III-Klausur

Aufgabe 1 (1+4+3+4 Punkte)

(a) 01010 wird nicht akzeptiert: $s \to q_0 \to q_1 \to q_2 \to f \to q_2$

10101 wird akzeptiert: $s \to q_2 \to q_2 \to f \to q_2 \to f$

(b) ε: $\{s, q_0, q_1, q_2\}, \{f\}$
0: $\{s, q_0, q_1, q_2\}, \{f\}$
1: $\{s, q_0\}, \{q_1, q_2\}, \{f\}$
00: $\{s, q_0\}, \{q_1, q_2\}, \{f\}$
01: $\{s\}, \{q_0\}, \{q_1, q_2\}, \{f\}$
10: $\{s\}, \{q_0\}, \{q_1, q_2\}, \{f\}$
11: $\{s\}, \{q_0\}, \{q_1, q_2\}, \{f\}$
Wörter der Länge 3 trennen keine Zustände mehr.



(c)

$$
\begin{aligned}
L_1/L_2 &= \{\varepsilon, b, a\} \\
L_1 \backslash L_2 &= \{aba, bab\} \\
(L_2)^2 &= \{bbbb, bbab, bbba, abbb, abab, abba, babb, baab, baba\}
\end{aligned}
$$

(d) Wir nehmen an, L sei regulär. Für $n \in \mathbb{N}$ sei $z_n := a^n b^n c^{2n} \in L$ mit $i = j = k = n$. Sei $z_n = uvw$ eine Zerlegung mit $|v| \geq 1$ und $|uv| \leq n$. Dann gilt $v = a^i$ mit $1 \leq i \leq n$ und somit $uv^2w = a^{n+i}b^n c^{2n}$. Wegen $n + i > n = k$ ist also $uv^2w \notin L$. Dies ist ein Widerspruch zum Pumping-Lemma. Also ist L nicht regulär.

Aufgabe 2 (1+3+5+3 Punkte)

(a) Die Menge der (Turing-)berechenbaren Funktionen ist genau die Menge der im intuitiven Sinne überhaupt berechenbaren Funktionen.

(b) Aus der Vorlesung ist bekannt, dass entscheidbare Sprachen bzgl. Vereinigung, Konkatenation und Komplementbildung abgeschlossen sind. Es gilt

$$L' = L \cup L_1 \cup 0L \cup 1L^c \ .$$

- L ist entscheibar nach Voraussetzung.
- L_1 ist kontextfrei und somit enscheidbar (vgl. Vorlesung). Eine kontextfreie Grammatik ist etwa durch die folgende Regelmenge gegeben:

$$R := \{ \ S \quad \rightarrow \quad 0S1 \mid \varepsilon \ \} \ .$$

- $0L$ ist als Konkatenation zweier entscheidbarer Sprachen wieder entscheidbar. Die Sprache $\{0\}$ ist regulär und somit entscheidbar.
- $1L^c$ ist wie oben als Konkatenation zweier entscheidbarer Sprachen wieder entscheidbar. Die Sprache $\{1\}$ ist wieder regulär und somit entscheidbar und die Sprache L^c ist als Komplement einer entscheidbaren Sprache entscheidbar.

Da L' die Vereinigung von 4 entscheidbaren Sprachen ist, ist L' auch entscheidbar.

(c)

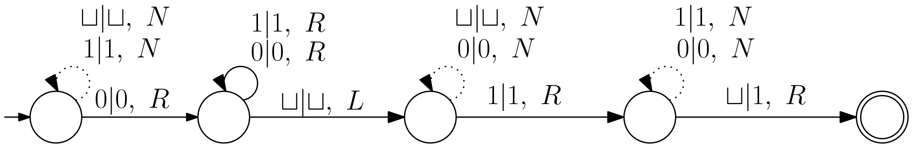
$$\begin{array}{rcl}
\sqcup & (s) & 01 \\
\sqcup & (s) & \sqcup 01 \\
X & (q_0) & 01 \\
X0 & (q_0) & 1 \\
X01 & (q_0) & \sqcup \\
X0 & (q_1) & 1 \\
X & (z_1) & 0 \\
\sqcup & (z_1) & X0 \\
\sqcup & (z_1) & \sqcup X0 \\
\sqcup & (q_0) & 1X0
\end{array}$$

Zu Beginn der Berechnung steht die TM am Wortanfang. Zunächst geht sie nach links und schreibt ein X vor die Eingabe.

Die Arbeitsweise der TM ist dann wie folgt: Die TM liest das Zeichen der Eingabe, welches am weitesten rechts auf dem Band steht. Falls dieses Zeichen ein X ist, so überschreibt die TM dieses mit $\sqcup$ und hält im (akzeptierendem) Zustand f (und zwar nur dann). Falls es sich bei dem gelesenen Zeichen um eine Null oder eine Eins handelt, wird diese Information im Zustand z_0 bzw. z_1 gespeichert und mit $\sqcup$ überschrieben. Anschließend geht die TM zum ersten $\sqcup$-Symbol links von der Eingabe und überschreibt dies mit der gespeicherten Zahl.

Die Turingmaschine berechnet die Identität, da die Zeichen von rechts nach links gelesen und anschließend wieder geschrieben werden.

(d) Eine DTM, die das gewünschte leistet, ist gegeben durch:



Aufgabe 3 (2+2+2+6 Punkte)

(a) $\mathcal{P} \subseteq co-\mathcal{P}$: Sei L in $\mathcal{P}$. Dann gibt es eine deterministische Turingmaschine $\mathcal{M}$ mit akzeptierendem Zustand q_J und ablehnendem Zustand q_N, die L in Polynomialzeit entscheidet. Konstruiert man zu $\mathcal{M}$ eine Turingmaschine $\mathcal{M}'$, die sich von $\mathcal{M}$ nur dadurch unterscheidet, dass q_J und q_N vertausch werden, so erhält man eine deterministische Turingmaschine $\mathcal{M}'$, die L^c in Polynomialzeit entscheidet. Damit gilt $L \in \mathcal{P} \Rightarrow L^c \in \mathcal{P} \Leftrightarrow L \in co-\mathcal{P}$.

$co-\mathcal{P} \subseteq \mathcal{P}$: Sei $L \in co-\mathcal{P}$. Dann ist $L^c \in \mathcal{P}$. Dann gibt es eine DTM, die L^c im Polynomialzeit entscheidet. Mit der gleichen Argumentation wie oben gibt es dann auch eine DTM, die $(L^c)^c = L$ in Polynomialzeit entscheidet. Also gilt auch $L \in \mathcal{P}$.

(b) Ein Turing Reduktion von R auf R' ist eine Orakel-Turingmaschine, deren Orakel die Relation R' realisiert und die selbst in polynomialer Zeit eine Funktion f berechnet, die R realisiert.

Da es für R bereits einen Polynomialzeitalgorithmus gibt, gibt es auch eine Turingmaschine, die R in polynomieller Zeit berechnet. Diese Turingmaschine erweitern wir um ein Orakel das R' realisiert und ignorieren das Orakel. Dadurch erhalten wir die gesuchte Orakel-Turingmaschine.

(c) Ein PAS ist eine Familie von Algorithmen $\{A_\epsilon | \epsilon > 0\}$, so dass A_ϵ ein ϵ-approximiernder Algorithmus ist. Der Algorithmus A_4 hat also relative Gütegarantie 5.

(d) ZEHNTELN liegt in NP: Um zu überprüfen, ob eine geratene Lösung $\widetilde{M}$ richtig ist, müssen wir lediglich über die Elemente in $\widetilde{M}$ bzw. $M \setminus \widetilde{M}$ summieren und überprüfen, ob die Summe der Elemente in $\widetilde{M}$ ein 9-faches der Summe der Elemente in $M \setminus \widetilde{M}$ ist. Dies ist in Polynomialzeit möglich.

Wir zeigen dass sich jedes Problem aus $\mathcal{NP}$ polynomiell auf ZEHNTELN reduzieren lässt, indem wir das NP-vollständige Problem PARTITION auf ZEHNTELN reduzieren.

Zu einer Instanz $I = (M, w)$ von PARTITION konstruieren wir eine Instanz $\tilde{I} = (M \cup \{m'\}, \widetilde{w})$ von ZEHNTELN mit

$$\widetilde{w}(m) = \begin{cases} w(m) & , m \in M \\ 4\sum_{w \in M} w(m) & , m = m' \end{cases}$$

Offensichtlich ist die Transformation polynomiell.

Es gilt: I hat genau dann eine Lösung, wenn $\tilde{I}$ eine Lösung hat:

Sei M' eine Lösung von I, dann

$$\sum_{m \in M'} w(m) = \sum_{m \in M \setminus M'} w(m) \Longrightarrow \sum_{m \in M'} \widetilde{w}(m) + \widetilde{w}(m') = 9 \sum_{m \in M \setminus M'} \widetilde{w}(m)$$

Sei $\widetilde{M}$ eine Lösung von $\tilde{I}$, dann gilt $m' \in \widetilde{M}$ und damit

$$\begin{aligned} \sum_{\substack{m \in \widetilde{M}, \\ m \neq m'}} \widetilde{w}(m) + \qquad \widetilde{w}(m') &= 9 \sum_{m \in M \setminus \widetilde{M}} \widetilde{w}(m) \\ \sum_{\substack{m \in \widetilde{M}, \\ m \neq m'}} w(m) + 4 \sum_{m \in M} w(m) &= 9 \sum_{m \in M \setminus \widetilde{M}} w(m) \\ \sum_{\substack{m \in \widetilde{M}, \\ m \neq m'}} w(m) &= \sum_{m \in M \setminus \widetilde{M}} w(m) \end{aligned}$$

Aufgabe 4 (1+4+3+4 Punkte)

(a) Die Grammatik ist in Greibach-Normalform, da alle Regeln von der Form $A \to a\alpha$ mit $A \in V$, $a \in \Sigma$, $\alpha \in V^*$ sind.

(b) Der Kellerautomat ist gegeben durch

- $Q = \{q\}$
- $\Sigma' = \Sigma$
- $\Gamma = \{S, B, N, E\}$
- $q_0 = q$
- $Z_0 = S$
- $F = \emptyset$
- δ:

$$\begin{array}{rcl} \delta(q, 0, S) & = & \{(q, SE)\} \\ \delta(q, 1, S) & = & \{(q, BN)\} \\ \delta(q, 1, B) & = & \{(q, BN), (q, N)\} \\ \delta(q, 0, N) & = & \{(q, \varepsilon)\} \\ \delta(q, 1, E) & = & \{(q, \varepsilon)\} \end{array}$$

Der Kellerautomat ist nichtdeterministisch, da $|\delta(q, 1, B)| = 2$ gilt.

(c) Ursprüngliche Regelmenge:

$$\begin{array}{rcl} R := \{\ S & \to & 0SE \mid 1BN, \\ B & \to & 1BN \mid 1N, \\ E & \to & 1, \\ N & \to & 0\ \} \end{array}$$

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

$$\begin{array}{rcl} R := \{\ S & \to & NSE \mid EBN, \\ B & \to & EBN \mid EN, \\ E & \to & 1, \\ N & \to & 0\ \} \end{array}$$

2. Schritt: alle rechten Seiten haben Länge ≤ 2

$$\begin{array}{rcl} R := \{\ S & \to & NC_1 \mid EC_2, \\ B & \to & EC_2 \mid EN, \\ E & \to & 1, \\ N & \to & 0, \\ C_1 & \to & SE, \\ C_2 & \to & BN\ \} \end{array}$$

Schritte 3 und 4 entfallen. Die resultierende Grammatik ist bereits in Chomsky-Normalform.

(d) Wir nehmen an, L sei kontextfrei. Für $n \in \mathbb{N}$ sei $z_n := a^n b^{2n} c^{3n} \in L$. Sei $z_n = uvwxy$ eine Zerlegung mit $|vx| \geq 1$ und $|vwx| \leq n$.

Wegen $|vwx| \leq n$ kann vx höchstens zwei Sorten von Zeichen enthalten (nur as und bs oder nur bs und cs). Also gilt $uv^0wx^0y = a^{n-i}b^{2n-j}c^{3n-k}$, wobei entweder $i + j \geq 1$ und $k = 0$ oder $j + k \geq 1$ und $i = 0$ gelten muss (wegen $|vx| \geq 1$).

Falls $i + j \geq 1$ ist, folgt $k = 0$. Dann ist $uv^0wx^0y = a^{n-i}b^{2n-j}c^{3n} \notin L$, da $|uv^0wx^0y|_c > 3|uv^0wx^0y|_a$ (wenn vx mindestens ein a enthält) oder $|uv^0wx^0y|_b < 2|uv^0wx^0y|_a$ (wenn vx kein a und mindestens ein b enthält).

Falls $j + k \geq 1$ ist, folgt $i = 0$. Dann ist $uv^0wx^0y = a^n b^{2n-j} c^{3n-k} \notin L$, da $|uv^0wx^0y|_c < 3|uv^0wx^0y|_a$ (wenn vx mindestens ein c enthält) oder $|uv^0wx^0y|_b < 2|uv^0wx^0y|_a$ (wenn vx mindestens ein b enthält).

Beide Fälle führen in einen Widerspruch zum Pumping-Lemma. Also ist L nicht kontextfrei.

Aufgabe 5

(12 × 1 Punkte)

	Wahr	Falsch
Es gibt Probleme in NP, die in polynomieller Zeit lösbar sind.	X	
Jedes NP-vollständige Problem ist semi-entscheidbar, aber nicht entscheidbar.		X
Die Klasse der von deterministischen Kellerautomaten akzeptierten Sprachen ist gleich der Klasse der von nichtdeterministischen Kellerautomaten akzeptierten Sprachen.		X
Das Post'sche Korrespondenzproblem ist NP-vollständig.		X
Der Schnitt zweier semi-entscheidbarer Sprachen ist entscheidbar.		X
Das Problem 2-SAT ist NP-vollständig.		X (Falls $\mathcal{P} \neq \mathcal{NP}$.)
Ein Problem Π ist genau dann NP-vollständig, wenn man jedes andere Problem Π' polynomiell auf Π reduzieren kann.		X
Es gibt eine Chomsky-0 Grammatik für die universelle Sprache.	X	
Der Schnitt zweier kontextfreier Sprachen ist kontextfrei.		X
Zu jedem NEA gibt es einen DEA mit nicht mehr Zuständen, der die gleiche Sprache akzeptiert.		X
Jede endliche Teilmenge der Diagonalsprache ist regulär.	X	
Folgende Instanz des Post'schen Korrespondenzproblems besitzt keine Lösung: $\{(a,b),(b,a)\}$.	X	