	Abschlussklausur	
	Theoretische Informatik 2	
Prof. Dr. Roland Meyer	**28. Mai 2020**	TU Braunschweig
Sebastian Muskalla		Wintersemester 2019/20

1. **Bitte am Anfang ausfüllen:**

   Vorname:

   Nachname:

   Matrikelnummer:

   Nummer des Sitzplatzes:

   Unterschrift:

2. Die Nummer Ihrer Klausur ist `altePO # 1`. Bitte merken Sie sich die Nummer. Wir werden Ihr Klausurergebnis anonymisiert unter Verwendung dieser Nummer bekanntgeben.

3. Achten Sie darauf, dass Ihre Klausur **vollständig ist** und **getackert bleibt** (20 Blätter).

4. Benutzen Sie **nur das an dieses Blatt angeheftete Papier**. Bei Bedarf können wir weitere Leerblätter austeilen. Wenn der Platz auf der Vorderseite des jeweiligen Aufgabenblatts nicht ausreicht, **machen Sie kenntlich**, wo Sie die Bearbeitung der Aufgabe fortsetzen.

5. Als Hilfsmittel sind **ausschließlich** Sprachwörterbücher sowie ein beidseitig **handschriftlich beschriebenes DIN A4-Blatt** erlaubt. Elektronische Geräte müssen während der Klausur ausgeschaltet bleiben. Täuschungsversuche werden als nicht bestanden gewertet und dem Prüfungsamt gemeldet.

6. Schreiben Sie leserlich und bearbeiten Sie Ihre Klausur mit einem **dokumentenechten Stift** (nicht mit Bleistift, kein Tipp-Ex, kein Tintenkiller) und **nicht in roter oder grüner Farbe**.

7. Wir werden die **Klausurergebnisse** auf unserer Website bekanntgeben:
   `tcs.cs.tu-bs.de/teaching/TheoInf2Klausur_WS_20192020`.

8. Die **Bearbeitungszeit** beträgt **180 Minuten** (+ ggf. Zeit zum Lüften).

9. Mit **40 Punkten** ist die Klausur **sicher bestanden.**

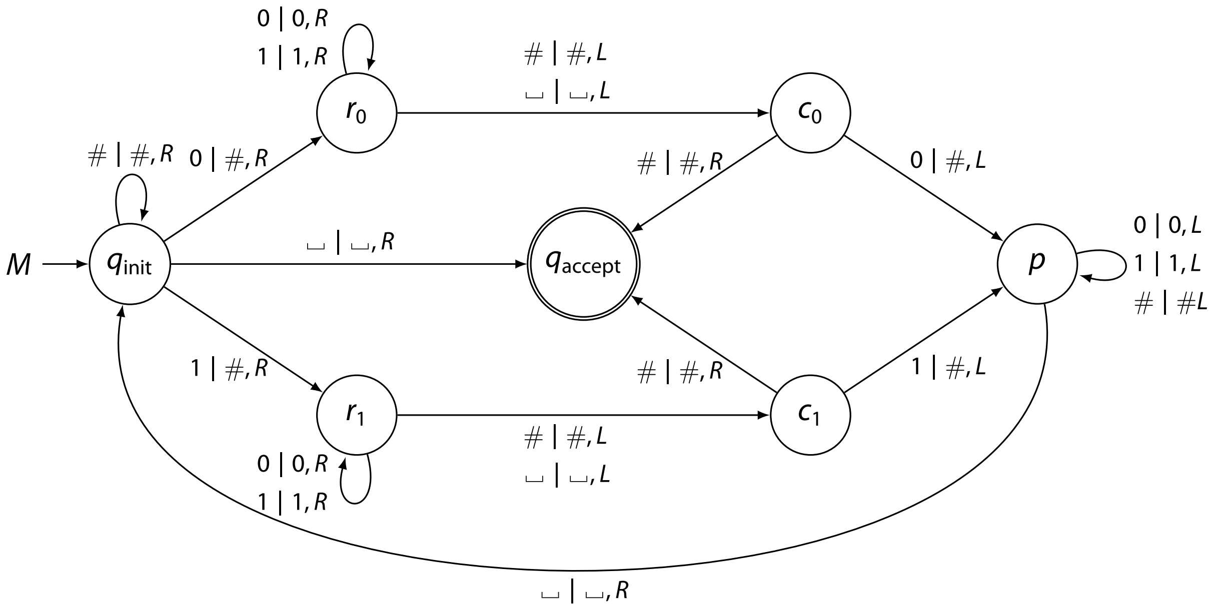
Bepunktung: (wird von den Korrektoren ausgefüllt)

Aufgabe	1	2	3	4	5	6	7	8	9	10	Σ
Max.	10	10	10	10	10	10	10	10	10	10	**100**
Punkte											

1. TM-Analyse

$\boxed{3 + 2 + 3 + 2 = 10 \text{ Punkte}}$

Betrachten Sie die 1-Band DTM M mit Eingabealphabet $\{0, 1\}$, Bandalphabet $\{0, 1, \#, \sqcup\}$ und Zustandsmenge $\{q_{\text{init}}, q_{\text{accept}}, q_{\text{reject}}, r_0, r_1, c_0, c_1, p\}$. Hierbei ist q_{init} Startzustand, q_{accept} der akzeptierende und q_{reject} der abweisende Zustand. Die Transitionsfunktion ist durch den folgenden Graphen gegeben.



Hierbei steht wie üblich eine mit $a \mid b, d$ beschriftete Kante von q nach q' für eine Transition $\delta(q, a) = (q', b, d)$ der Maschine. Kanten mit mehreren Beschriftungen stehen für mehrere Transitionen. Alle fehlenden Transitionen führen in den abweisenden Zustand q_{reject}.

Wir gehen davon aus, dass die Maschine bei Eingabe w in der Konfiguration $\ldots \sqcup q_{\text{init}}\, w \sqcup \ldots$ startet, d.h. links und rechts von der Eingabe befinden sich $\sqcup$-Symbole und der Lese-/Schreibkopf zeigt auf den ersten Buchstaben der Eingabe.

a) Beschreiben Sie, wie sich Maschine M verhält, wenn Sie auf einer Eingabe $w \in \{0, 1\}^*$ mit **gerader Länge** gestartet wird.

b) In wie fern verhält sich die Maschine anders, wenn Sie auf einer Eingabe $w \in \{0, 1\}^*$ mit **ungerade Länge** gestartet wird?

c) Geben Sie die Sprache $\mathcal{L}(M) \subseteq \{0, 1\}^*$ von M an und begründen Sie Ihre Antwort kurz.

d) Wie viele Schritte benötigt M, wenn sie auf einer Eingabe der Länge n gestartet wird? Geben Sie eine grobe Abschätzung und eine kurze Begründung an.

Zu Aufgabe 1:

2. NP-Vollständigkeit

$\boxed{4 + 6 = 10 \text{ Punkte}}$

Betrachten Sie das folgende Problem.

Pfad-mit-allen-außer-2 (PAB2)	
Gegeben:	Ein gerichteter Graph $G = (V, E)$.
Entscheide:	Gibt es zwei unterschiedliche Knoten $v_1 \neq v_2$ und einen Pfad im Graphen, der v_1, v_2 **nicht** und alle anderen Knoten **genau einmal** besucht?

Zeigen Sie, dass PAB2 NP-vollständig (bzgl. Polynomialzeit-Reduktionen) ist:

a) „Membership“: PAB2 $\in$ NP.

b) „Hardness“: PAB2 ist NP-schwer (bzgl. Polynomialzeit-Reduktionen).

3. Entscheidbarkeit

$4 + 6 = 10$ Punkte

Betrachten Sie das folgende Problem.

<u>5-ACCEPT</u>

Gegeben: Kodierung w einer DTM mit Eingabealphabet $\{0, 1\}$.

Entscheide: Gibt es ein Wort $x \in \mathcal{L}(M_w)$ mit Länge $|x| = 5$?

a) Zeigen Sie, dass 5-ACCEPT semi-entscheidbar ist.

b) Zeigen Sie, dass 5-ACCEPT unentscheidbar ist. Verwenden Sie **nicht** den Satz von Rice.

4. NL-Vollständigkeit

$\boxed{4 + 6 = 10 \text{ Punkte}}$

Betrachten Sie das folgende Problem.

Keine Tour durch 3 Knoten (NO3TOUR)
Gegeben: Ein gerichteter Graph $G = (V, E)$ und Knoten s, t, r.
Entscheide: Gibt es in G **keinen** Kreis der Form $s \to^* t \to^* r \to^* s$?

Zeigen Sie, dass NO3TOUR NL-vollständig (bzgl. logspace-many-one-Reduktionen) ist:

a) „Membership": NO3TOUR $\in$ NL.

b) „Hardness": NO3TOUR ist NL-schwer (bzgl. logspace-many-one-Reduktionen).

5. Berechenbarkeit

10 Punkte

Es sei $\Sigma = \{0, 1\}$. Betrachten Sie die Funktion *shortest* $: \Sigma^* \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$. Die Funktion nimmt als Argumente $w \in \Sigma^*$, die Kodierung einer deterministischen Turingmaschine, und Zahlen n, m. Der Funktionswert $\mathit{shortest}(w, n, m)$ ist k, wenn k die kleinste Zahl ist, so dass es ein Wort x mit Länge $|x| = k \leqslant n$ gibt, auf dem die durch w kodierte Maschine innerhalb von höchstens m Schritten hält ($\mathrm{Time}_{M_w}(x) \leqslant m$). Wenn es kein solches Wort gibt, dann ist $\mathit{shortest}(w, n, m) = n + 1$.

Beweisen Sie, dass die Funktion *shortest* berechenbar ist.
Geben Sie hierzu einen Algorithmus (als Pseudo-Code) an.

Hinweis: Die Kodierung der Zahlen in der Ein- und Ausgabe der Funktion (z.B. unär oder binär) ist für die Bearbeitung der Aufgabe unerheblich.

6. Konstruktion einer DTM

10 Punkte

Es sei $\Sigma = \{a, b\}$ und $w \in \Sigma^*$ ein Wort. Wir bezeichnen mit $|w|_a$ und $|w|_b$ die Anzahl der *a*s bzw. *b*s in w.

Konstruieren Sie eine **deterministische** Turingmaschine M, welche die Sprache

$$\mathcal{L} = \left\{ w \in \Sigma^* \;\middle|\; (|w|_a)^2 = |w|_b \right\}$$

entscheidet. Beispielsweise soll M die Wörter ε, ab und $aabbbb$ akzeptieren.

- Erklären Sie die Arbeitsweise der Maschine ausführlich. Geben Sie insbesondere die Aufgabe jedes Kontrollzustands der Maschine an.
- Geben Sie die Transitionen der Maschine explizit an, z.B. in Form einer Tabelle oder als Zustandsgraph.
- Sie können wahlweise annehmen, dass das Band auf beiden Seiten der Eingabe mit $\sqcup$-Symbolen gefüllt ist, oder dass das Band auf der linken Seite durch ein \$-Symbol beschränkt ist. Geben Sie an, wofür Sie sich entschieden haben und geben Sie an, auf welches Symbol der Lese-/Schreibkopf initial zeigt.

Hinweis: Die Turingmaschine darf mehrere Bänder verwenden.

Zu Aufgabe 6:

7. Orakelmaschinen

10 Punkte

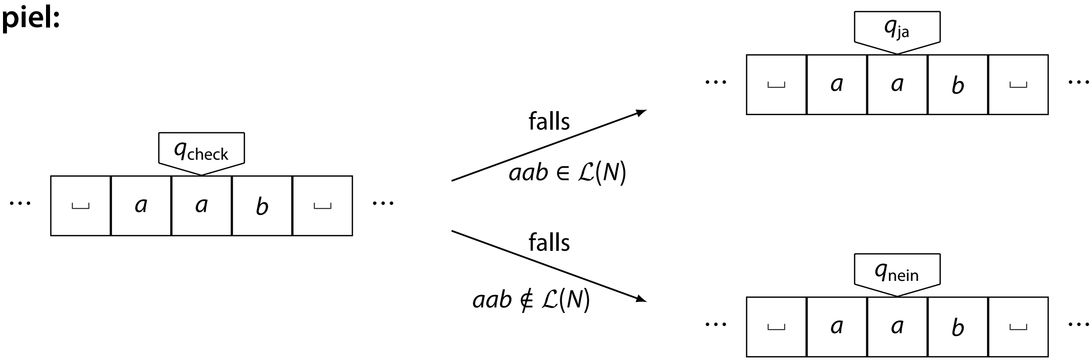
Es sei N eine 1-Band DTM mit Eingabealphabet Γ, die ein Entscheider ist.

Eine Maschine M mit N-Orakel ist eine 1-Band DTM mit Bandalphabet Γ, die über drei spezielle Zustände q_{check}, q_{ja} und q_{nein} verfügt. Mit Hilfe dieser Zustände kann die Maschine in einem einzigen Schritt bestimmen, ob der aktuelle Bandinhalt Element von $\mathcal{L}(N)$ ist.

- In allen Zuständen außer q_{check} verhält sich die Maschine wie eine gewöhnliche DTM; ihre Transitionen sind durch die Transitionsfunktion spezifiziert.
- Wenn die Maschine im Laufe einer Berechnung den Zustand q_{check} betritt, z.B. in Konfiguration $w\ q_{\text{check}}\ v$, so geht die Maschine im nächsten Schritt entweder in Zustand q_{ja}, falls Bandinhalt $w.v \in \mathcal{L}(N)$, oder in q_{nein}, falls $w.v \notin \mathcal{L}(N)$. Bandinhalt und Kopfposition bleiben unverändert.

Die Sprache $\mathcal{L}(M)$ von M ist wie üblich als die Menge der Eingaben definiert, die von der Maschine nach endlich vielen Schritten akzeptiert werden.

Beispiel:



Erklären Sie, wie aus einer gegebenen Maschine M mit N-Orakel eine herkömmliche deterministische Turingmaschine M' mit $\mathcal{L}(M') = \mathcal{L}(M)$ konstruiert werden kann. Geben Sie hierzu insbesondere an, wie M' eine Transition von M simuliert, z.B. in Form von Pseudo-Code.

Hinweise: Die von Ihnen konstruierte DTM M' darf mehrere Bänder verwenden. Beachten Sie, dass N ein Entscheider ist, also auf allen Eingaben nach endlich vielen Schritten hält.

Zu Aufgabe 7:

8. Quiz

$\boxed{2 + 2 + 3 + 3 = 10 \text{ Punkte}}$

Beantworten Sie die folgenden Fragen. Begründen Sie Ihre Antwort mit einem kurzen Beweis oder einem Gegenbeispiel.

a) Es sei $\mathcal{L} \subseteq \Sigma^*$ eine semi-entscheidbare Sprache. Ist dann $\overline{\mathcal{L}} = \Sigma^* \setminus \mathcal{L}$ immer unentscheidbar?

b) Es sei $\mathcal{L} \subseteq \Sigma^*$ eine semi-entscheidbare Sprache. Ist dann $\overline{\mathcal{L}} = \Sigma^* \setminus \mathcal{L}$ immer semi-entscheidbar?

c) Betrachten Sie das folgende Problem.

Hamiltonian-Cycle-100	
Gegeben:	Eine gerichteter Graph $G = (V, E)$ mit höchstens 100 Knoten.
Entscheide:	Gibt es in G einen Hamiltonschen Kreis?

Ist Hamiltonian-Cycle-100 NP-vollständig?

d) Zu einer reellen Zahl $r \in \mathbb{R}$ definieren wir eine Funktion $f_r : \mathbb{N} \to \{0, 1, 2, \ldots, 9\}$, so dass $f_r(n)$ die n-te Nachkommastelle von r ist. Beispielsweise gilt $f_\pi(1) = 1$ und $f_\pi(2) = 4$, denn $\pi = 3,14\ldots$ Ist f_r für jede reelle Zahl r berechenbar?

9. Der Boolesche Abschluss

$\boxed{2 + 8 = 10 \text{ Punkte}}$

Zu einer Klasse $\mathcal{C}$ von Problemen definieren wir den **Booleschen Abschluss** BoolCl($\mathcal{C}$) von $\mathcal{C}$ als die kleinste Klasse von Problemen, die folgendes erfüllt:

- $\mathcal{C} \subseteq$ BoolCl($\mathcal{C}$), also wenn $\mathcal{L} \in \mathcal{C}$, dann auch $\mathcal{L} \in$ BoolCl($\mathcal{C}$),
- wenn $\mathcal{L} \in$ BoolCl($\mathcal{C}$), dann gilt auch für das Komplement $\overline{\mathcal{L}} \in$ BoolCl($\mathcal{C}$),
- wenn $\mathcal{L}, \mathcal{L}' \in$ BoolCl($\mathcal{C}$) Probleme über dem gleichen Alphabet sind, dann gilt für ihre Vereinigung $\mathcal{L} \cup \mathcal{L}' \in$ BoolCl($\mathcal{C}$),
- wenn $\mathcal{L}, \mathcal{L}' \in$ BoolCl($\mathcal{C}$) Probleme über dem gleichen Alphabet sind, dann gilt für ihren Schnitt $\mathcal{L} \cap \mathcal{L}' \in$ BoolCl($\mathcal{C}$).

Beweisen Sie:

a) Angenommen, wir hätten NP $\neq$ coNP gezeigt. Dann folgt NP $\neq$ BoolCl(NP).

b) Es gilt PSPACE = BoolCl(PSPACE).

10. Entscheidbarkeit II

10 Punkte

Wir betrachten das Problem

$$\mathcal{L} = \left\{ w_1 \# w_2 \in \{0,1\}^* \# \{0,1\}^* \;\middle|\; \mathcal{L}\left(M_{w_1}\right) \cap \overline{\mathcal{L}\left(M_{w_2}\right)} = \emptyset \right\} \subseteq \{0,1,\#\}^*.$$

Ein Paar $w_1 \# w_2$ ist also genau dann in $\mathcal{L}$, wenn die Sprache $\mathcal{L}(M_{w_1})$ der von w_1 kodierten Maschine und das Komplement $\overline{\mathcal{L}(M_{w_2})}$ der Sprache der von w_2 kodierten Maschine kein gemeinsames Element enthalten.

Beweisen Sie, dass $\mathcal{L}$ weder semi-entscheidbar, noch co-semi-entscheidbar ist.

1. Zusatzblatt

2. Zusatzblatt

3. Zusatzblatt

4. Zusatzblatt

5. Zusatzblatt

6. Zusatzblatt