Technische Universität Berlin, Berlin, 28.07.2016

Name: .... **Matr.-Nr.: ....**

Klausur: Berechenbarkeit und Komplexität

(Niedermeier/Chen/Froese/Sorge, Sommersemester 2016)

Einlesezeit:	15 Minuten
Bearbeitungszeit:	60 Minuten
Max. Punktezahl:	50 Punkte

1	**2**	**3**	**4**	**5**	**Σ**
(12)	(10)	(8)	(12)	(8)	(50)

Allgemeine Hinweise:

- Es sind keinerlei Hilfsmittel erlaubt.
- Benutzen Sie einen dokumentenechten Stift in der Farbe schwarz oder blau. Insbesondere also keinen Bleistift, sondern einen Kugelschreiber oder einen nicht löschbaren Füller.
- Beschriften Sie jedes Blatt mit ihren Vor- und Nachnamen und ihrer Matrikelnummer.
- **Falls es in der Aufgabenstellung nicht explizit ausgeschlossen wird, so sind alle Antworten zu begründen! Antworten ohne Begründung erhalten 0 Punkte.**

Viel Erfolg!

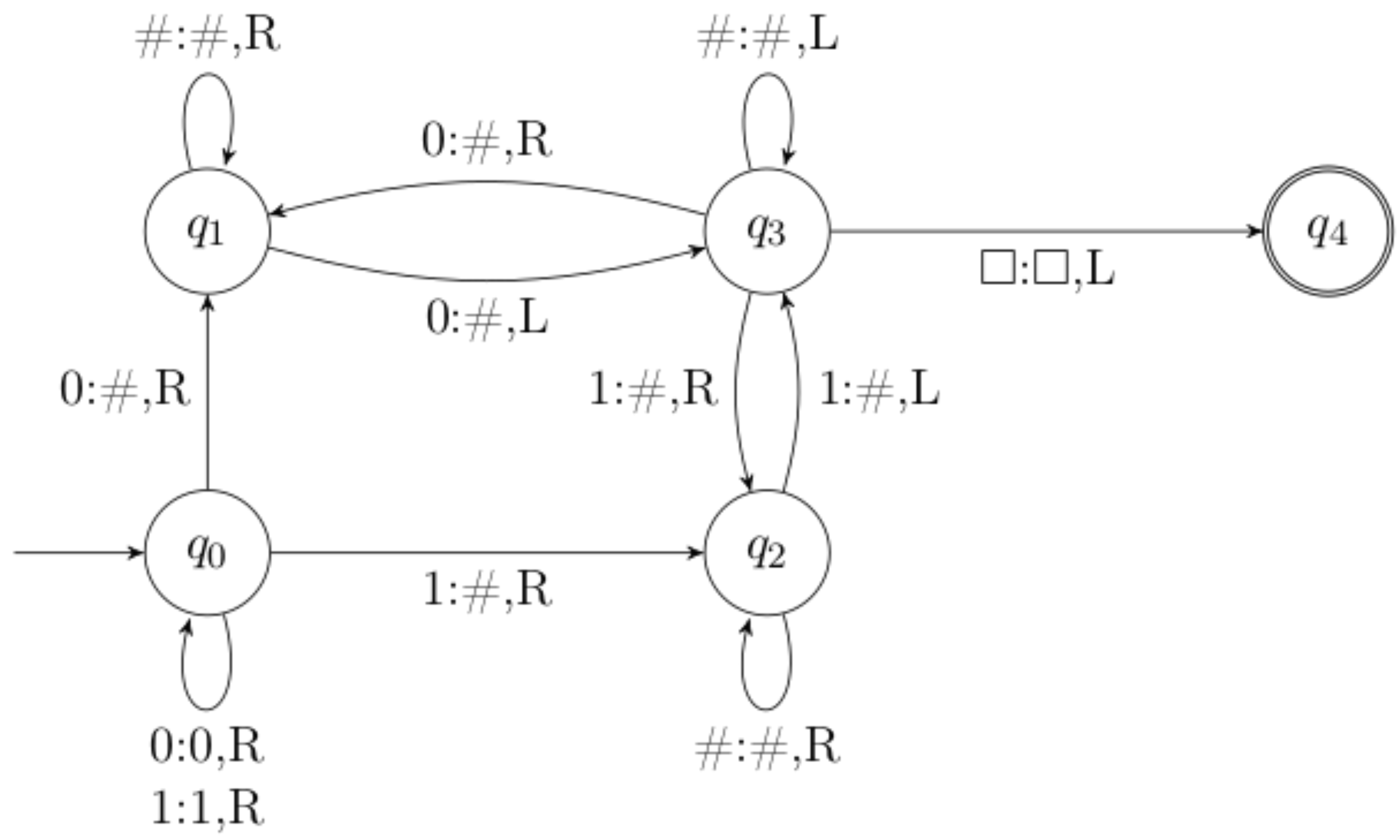
Name: .... **Matr.-Nr.: ....**

Aufgabe 1: **Turing-Maschinen** (6 + 6 Punkte)

Betrachten Sie die nichtdeterministische Turing-Maschine

$$M = (\{q_0, q_1, q_2, q_3, q_4\}, \{0, 1\}, \{0, 1, \#, \Box\}, \delta, q_0, \Box, \{q_4\}),$$

wobei δ wie folgt definiert ist:



Hinweis: Beispielsweise bedeutet der Pfeil mit Beschriftung „0:#,L" von q_1 zu q_3, dass M im Zustand q_1 beim Lesen von 0 in den Zustand q_3 übergeht, die 0 durch # ersetzt, und ihren Leseschreibkopf nach links bewegt.

a) Finden Sie ein Wort der Länge drei, das nicht von der Turing-Maschine M akzeptiert wird. Beweisen Sie, dass die Turing-Maschine M ihr Wort nicht akzeptiert.

b) Vervollständigen Sie die Konfigurationsfolge

$$q_0 0110 \vdash 0 q_0 110 \vdash \dots$$

um zu zeigen, dass die Turing-Maschine M das Wort 0110 akzeptiert (ohne weitere Begründungen).

Name: .... **Matr.-Nr.: ....**

Aufgabe 2: **Reduzierbarkeit und (Un-)Entscheidbarkeit** (4 + 6 Punkte)

Gegeben seien zwei endliche Alphabete Σ und Γ. Wir definieren, dass eine Sprache $L_1 \subseteq \Sigma^*$ **doppel-reduzierbar** auf eine andere Sprache $L_2 \subseteq \Gamma^*$ ist, in Zeichen $L_1 \leq_d L_2$, falls zwei totale, berechenbare Funktionen f_1 und $f_2 \colon \Sigma^* \to \Gamma^*$ existieren, so dass für alle Wörter $x \in \Sigma^*$ gilt:

$$\big((x \in L_1) \Leftrightarrow (f_1(x) \in L_2)\big) \textbf{ und } \big((x \in L_1) \Leftrightarrow (f_2(x) \notin L_2)\big).$$

Begründen oder widerlegen Sie die Korrektheit folgender Aussagen.

a) Falls L_1 doppel-reduzierbar auf L_2 ist und zusätzlich L_2 entscheidbar ist, dann ist L_1 entscheidbar.

b) Sei $H \subseteq \Sigma^*$ das allgemeine Halteproblem und sei $L \subseteq \Gamma^*$ eine beliebige Sprache. Falls H reduzierbar auf L ist, dann ist H auch doppel-reduzierbar auf L. In Zeichen:

$$(H \leq L) \Rightarrow (H \leq_d L).$$

Hinweis: Das allgemeine Halteproblem H ist semi-entscheidbar und $\Sigma^* \setminus H$ ist unentscheidbar.

Name: .... **Matr.-Nr.:** ....

Aufgabe 3: **Satz von Rice** (4 + 4 Punkte)

Sei Σ ein endliches Alphabet und sei M_w die von $w \in \Sigma^*$ kodierte Turing-Maschine. Der aus der Vorlesung bekannte Satz von Rice lautet wie folgt:

> Sei $\mathcal{R}$ die Menge aller Turing-berechenbaren Funktionen. Sei außerdem $\mathcal{S} \subseteq \mathcal{R}$ eine nicht-triviale Teilmenge von $\mathcal{R}$, das heißt, $\mathcal{S} \neq \emptyset$ und $\mathcal{S} \neq \mathcal{R}$.
> Dann ist die Sprache $\mathcal{C}(\mathcal{S}) := \{w \in \Sigma^* \mid \text{die von } M_w \text{ berechnete Funktion liegt in } \mathcal{S}\}$ unentscheidbar.

Zeigen Sie für die folgenden beiden Sprachen entweder deren Unentscheidbarkeit mit Hilfe des Satzes von Rice oder beschreiben Sie die Vorgehensweise einer Turing-Maschine, die die Sprache entscheidet.

a) $L_1 \coloneqq \{w \in \Sigma^* \mid \text{die von } M_w \text{ berechnete Funktion ist } \textit{keine} \text{ konstante Funktion}\}$

b) $L_2 \coloneqq \{w \in \Sigma^* \mid M_w \text{ akzeptiert } w \text{ in höchstens } |w|^2 \text{ Schritten}\}$

Name: .... **Matr.-Nr.:** ....

Aufgabe 4: **NP-Vollständigkeit und Polynomzeitreduktion** (2 + 5 + 5 Punkte)

Betrachten Sie die aus der Vorlesung bekannten NP-vollständigen Probleme Dominating Set und Set Cover:

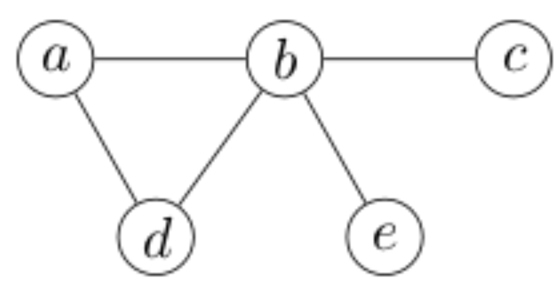
Dominating Set
Eingabe: Ein ungerichteter Graph $G = (V, E)$ und eine natürliche Zahl $k \in \mathbb{N}$.
Frage: Existiert eine Teilmenge $W \subseteq V$, sodass $|W| \leq k$ und jeder Knoten in $V \setminus W$ mindestens einen Nachbarn in W hat?

Set Cover
Eingabe: Eine Grundmenge X, eine Familie $\mathcal{F} = \{S_1, \ldots, S_m\}$, wobei jedes S_i eine Teilmenge von X ist, und eine natürliche Zahl $k \in \mathbb{N}$.
Frage: Existiert eine Teilfamilie $\mathcal{F}' \subseteq \mathcal{F}$, sodass $|\mathcal{F}'| \leq k$ und die Vereinigung der enthaltenen Mengen die Grundmenge ergibt, also $\bigcup_{S \in \mathcal{F}'} S = X$?

Für einen ungerichteten Graphen $G = (V, E)$ bezeichnen wir mit $N_G(v) \subseteq V$ die Menge der Nachbarn eines Knotens $v \in V$. Beispielsweise sind $N_G(a) = \{b, d\}$ und $N_G(b) = \{a, c, d, e\}$ im unten abgebildeten Graphen G. Betrachten Sie die polynomzeitberechenbare Funktion f, die für jeden Graphen G und jede natürliche Zahl $k \in \mathbb{N}$ wie folgt definiert ist:

$$f(\langle G = (V, E), k \rangle) = \langle \{N_G(v) \mid v \in V\}, k \rangle.$$

a) Sei $\langle G, 1 \rangle$ eine Eingabeinstanz für Dominating Set, wobei G wie folgt definiert ist:



Geben Sie $f(\langle G, 1 \rangle)$ an (ohne Begründung).

b) Begründen Sie anhand des Beispiels aus Teilaufgabe a), warum die Funktion f *keine* Reduktionsfunktion von Dominating Set auf Set Cover ist.

c) Korrigieren Sie die Definition von f, sodass sich eine polynomzeitberechenbare Reduktionsfunktion f' von Dominating Set auf Set Cover ergibt. Beweisen Sie anschließend die Korrektheit der Reduktion vermöge f'.

Name: .... **Matr.-Nr.:** ....

Aufgabe 5: **Vermischtes zu P, NP, und anderen Komplexitätsklassen** $(2+2+4$ Punkte$)$

Im Folgenden sei Σ ein endliches Alphabet.

a) Beweisen oder widerlegen Sie:

Wenn $\mathrm{P} = \mathrm{NP}$ gilt, dann liegt jede NP-schwere Sprache $A \subseteq \Sigma^*$ in P.

b) Für eine monoton wachsende Funktion $f\colon \mathbb{N} \to \mathbb{N}$ sei $\mathrm{DSPACE}(f)$ die Klasse aller Sprachen $L \subseteq \Sigma^*$, für die eine deterministische Turing-Maschine M mit $T(M) = L$ existiert, sodass M für jede Eingabe $x \in \Sigma^*$ hält und höchstens $f(|x|)$ viele Zellen auf dem Band modifiziert.

Beweisen oder widerlegen Sie:

Für jede monoton wachsende Funktion $f\colon \mathbb{N} \to \mathbb{N}$ gilt $\mathrm{DTIME}(f) \subseteq \mathrm{DSPACE}(f)$.

c) *Zur Erinnerung*: Das Komplement $\overline{L} \subseteq \Sigma^*$ einer Sprache $L \subseteq \Sigma^*$ ist definiert als $\overline{L} := \Sigma^* \setminus L$. Die Komplexitätsklasse coNP ist definiert als $\mathrm{coNP} := \{L \subseteq \Sigma^* \mid \overline{L} \in \mathrm{NP}\}$. Eine Sprache $A \subseteq \Sigma^*$ heißt **coNP-vollständig**, falls $A \in \mathrm{coNP}$ und für alle $B \in \mathrm{coNP}$ gilt, dass $B \leq_m^p A$.

Beweisen oder widerlegen Sie:

Für jede NP-vollständige Sprache $A \subseteq \Sigma^*$ gilt, dass $\overline{A}$ coNP-vollständig ist.