Technische Universität Berlin, Berlin, 04.08.2017

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

Klausur: Berechenbarkeit und Komplexität (A)

(Niedermeier/Froese/Molter, Sommersemester 2017)

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

1	**2**	**3**	**4**	**5**	**Σ**
(9)	(10)	(8)	(12)	(11)	(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.
- Beschriften Sie jedes Blatt mit ihrem 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!

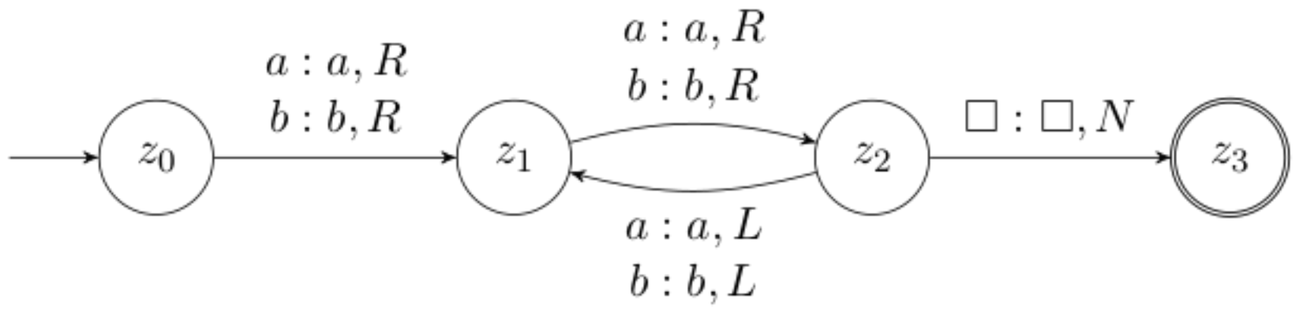
Name: ..... **Matr.-Nr.: .....**

Aufgabe 1: **Turing-Maschinen** (3 + 3 + 3 Punkte)

Betrachten Sie die deterministische Turing-Maschine

$$M = (\{z_0, z_1, z_2, z_3\}, \{a, b\}, \{a, b, \square\}, \delta, z_0, \square, \{z_3\}),$$

wobei δ wie folgt definiert ist:



(a) Hält M auf dem Eingabewort aba?

(b) Auf wievielen verschiedenen Bandzellen befindet sich der Leseschreibkopf von M maximal bei einer beliebigen Eingabe $x \in \{a, b\}^*$?

(c) Ist die von M akzeptierte Sprache $T(M)$ entscheidbar?

——————Lösung——————

(a) Die Konfigurationsfolge von M auf der Eingabe aba lautet

$$z_0aba \vdash_M^1 az_1ba \vdash_M^1 abz_2a \vdash_M^1 az_1ba \vdash_M^1 \ldots$$

Da die Konfiguration az_1ba zweimal auftaucht, hält M auf aba also nicht.

(b) Der Leseschreibkopf befindet sich bei jeder Eingabe auf maximal 3 Zellen. Wenn M hält, dann wurde der Kopf höchstens zweimal nach rechts bewegt. Wenn M nicht hält, dann bewegt sich der Kopf immer zwischen der zweiten und dritten besuchten Zelle hin und her.

(c) Ja. Die TM M akzeptiert nur Wörter der Länge 2. Also ist $T(M) = \{aa, ab, ba, bb\}$ endlich und damit entscheidbar (sogar regulär).

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

Aufgabe 2: **Die Komplexitätsklasse P** (5 + 5 Punkte)

Im Folgenden sei Σ ein endliches Alphabet und $A, B \subseteq \Sigma^*$ seien zwei Sprachen in P.

Begründen Sie für die beiden folgenden Sprachen, dass diese auch in P liegen (eine informelle algorithmische Beschreibung ist hierbei ausreichend).

(a) $A \cup B$

(b) $(\Sigma^* \setminus A) \cap (\Sigma^* \setminus B)$

————————Lösung————————

Im Folgenden sei M_A eine DTM, die A in polynomieller Zeit p_A entscheidet, und sei M_B eine DTM, die B in polynomieller Zeit p_B entscheidet (p_A, p_B seien zwei Polynome). Diese existieren nach Voraussetzung, da A und B in P liegen.

(a) Eine DTM M, die $A \cup B$ entscheidet, arbeitet wie folgt: Bei Eingabe $w \in \Sigma^*$, simuliert M zunächst M_A auf w. Falls M_A akzeptiert, so akzeptiert M die Eingabe w, denn dann gilt $w \in A$, also $w \in A \cup B$.

Falls M_A das Wort w ablehnt, dann wird M_B auf w simuliert. Falls nun M_B akzeptiert, so akzeptiert auch M, da $w \in B$, also $w \in A \cup B$. Ansonsten lehnt M das Wort w ab, da $w \notin A$ und $w \notin B$, also $w \notin A \cup B$.

Für jede Eingabe w gilt $\text{time}_M(w) \in O(p_A(|w|) + p_B(|w|))$. Somit hat M polynomielle Laufzeit.

(b) Eine DTM M, die $(\Sigma^* \setminus A) \cap (\Sigma^* \setminus B)$ entscheidet, arbeitet wie folgt: Bei Eingabe $w \in \Sigma^*$, simuliert M zunächst M_A auf w. Falls M_A akzeptiert, so lehnt M die Eingabe w ab, denn dann gilt $w \in A$, also $w \notin \Sigma^* \setminus A$ und somit $w \notin (\Sigma^* \setminus A) \cap (\Sigma^* \setminus B)$.

Falls M_A das Wort w ablehnt, dann wird M_B auf w simuliert. Falls nun M_B akzeptiert, so lehnt M ab, da $w \in B$, also $w \notin \Sigma^* \setminus B$ und somit $w \notin (\Sigma^* \setminus A) \cap (\Sigma^* \setminus B)$. Ansonsten akzeptiert M das Wort w, da $w \notin A$ und $w \notin B$, also $w \in (\Sigma^* \setminus A) \cap (\Sigma^* \setminus B)$.

Für jede Eingabe w gilt $\text{time}_M(w) \in O(p_A(|w|) + p_B(|w|))$. Somit hat M polynomielle Laufzeit.

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

Aufgabe 3: **Transitivität von Polynomzeitreduktionen** (4+4 Punkte)

Im Folgenden seien Σ und Π zwei endliche Alphabete. Betrachten Sie die folgenden beiden Reduktionstypen.

Definition 1. Eine Sprache $A \subseteq \Sigma^*$ heißt **linearzeit-reduzierbar** bzw. **quadratzeit-reduzierbar** auf eine Sprache $B \subseteq \Pi^*$ (in Zeichen $A \leq_m^\ell B$ bzw. $A \leq_m^q B$) genau dann, wenn es eine totale, in *linearer* Zeit ($O(|x|)$ für jedes $x \in \Sigma^*$) bzw. *quadratischer* Zeit ($O(|x|^2)$ für jedes $x \in \Sigma^*$) berechenbare Funktion $f : \Sigma^* \to \Pi^*$ gibt, sodass gilt:

$$\forall x \in \Sigma^* : x \in A \Leftrightarrow f(x) \in B.$$

(a) Begründen Sie die Transitivität für einen der beiden Reduktionstypen.

(b) Argumentieren Sie kurz (in 2-3 Sätzen), warum Transitivität im Kontext des Vollständigkeitskonzepts eine sinnvolle Eigenschaft für Reduktionen ist.

————————Lösung————————

(a) Wir zeigen, dass $\leq_m^\ell$ transitiv ist.
Seien dazu $A \subseteq \Sigma_A^*$, $B \subseteq \Sigma_B^*$ und $C \subseteq \Sigma_C^*$ Sprachen, sodass $A \leq_m^\ell B$ und $B \leq_m^\ell C$. Sei $f : \Sigma_A^* \to \Sigma_B^*$ die Reduktionsfunktion für $A \leq_m^\ell B$ und sei $g : \Sigma_B^* \to \Sigma_C^*$ die Reduktionsfunktion für $B \leq_m^\ell C$. Wir zeigen $A \leq_m^\ell C$. Als Reduktionsfunktion wählen wir $g \circ f : \Sigma_A^* \to \Sigma_C^*$. Dann gilt nach Voraussetzung für alle $x \in \Sigma_A^*$, dass

$$x \in A \Leftrightarrow f(x) \in B \Leftrightarrow g(f(x)) \in C.$$

Außerdem ist $g \circ f$ total (da f und g total sind) und $g(f(x))$ kann in $O(|x|+|f(x)|) = O(|x|)$ Zeit berechnet werden, da $|f(x)| \in O(|x|)$.

(b) Um die Vollständigkeit einer Sprache A für eine Klasse zu zeigen, reicht es dank Transitivität aus, von einer vollständigen Sprache B auf A zu reduzieren. Da sich alle Sprachen der Klasse auf B reduzieren lassen, kann man dann auch jede Sprache auf A reduzieren. Man muss somit nicht jede Sprache der Klasse einzeln auf A reduzieren.

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

Aufgabe 4: **Polynomzeitreduktion** (4+2+3+3 Punkte)

Betrachten Sie die folgenden Probleme.

Hamiltonpfad
Eingabe: Ein ungerichteter Graph $G = (V, E)$.
Frage: Gibt es einen *Pfad* in G, der jeden Knoten aus V genau einmal enthält?

Hamiltonkreis
Eingabe: Ein ungerichteter Graph $G = (V, E)$.
Frage: Gibt es einen *Kreis* in G, der jeden Knoten aus V genau einmal enthält?

Geben Sie eine Polynomzeitreduktion f von Hamiltonpfad auf Hamiltonkreis an, indem Sie

(a) einen Knoten zum Eingabegraph hinzufügen und diesen geeignet mit den restlichen Knoten verbinden,

(b) begründen, dass f in Polynomzeit berechnet werden kann,

(c) zeigen, dass für alle Graphen G gilt: $G \in$ Hamiltonpfad $\Rightarrow f(G) \in$ Hamiltonkreis und

(d) zeigen, dass für alle Graphen G gilt: $f(G) \in$ Hamiltonkreis $\Rightarrow G \in$ Hamiltonpfad.

Lösung

(a) Sei $G = (V, E)$ mit $|V| = n$. Wir definieren $f(G) := G'$, wobei G' durch Hinzufügen eines neuen Knotens v zu G entsteht, wobei wir v mit allen Knoten in G durch eine Kante verbinden.

(b) Um $f(G)$ zu erzeugen, muss nur der neue Knoten v mit allen n ursprünglichen Knoten in G verbunden werden. Dies lässt sich in $O(n)$ Zeit berechnen.

(c) Seien $v_1, \ldots, v_n$ die Knoten auf einem Hamiltonpfad in G. Dann existieren also die Kanten $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}$ in G und somit auch in G'. Außerdem enthält G' per Definition in (a) die Kanten $\{v_n, v\}$ und $\{v_1, v\}$. Also existiert ein Hamiltonkreis $v_1, \ldots, v_n, v$ in G'.

(d) Seien $v_1, \ldots, v_n, v$ die Knoten auf einem Hamiltonkreis in G'. Dann existieren also die Kanten $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_n, v\}$ und $\{v, v_1\}$ in G'. Also enthält G die Kanten $\{v_1, v_2\}, \ldots, \{v_{n-1}, v_n\}$ und somit einen Hamiltonpfad auf den Knoten $v_1, \ldots, v_n$.

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

Aufgabe 5: **Vermischtes zu Komplexitätsklassen** (2 + 2 + 2 + 5 Punkte)

(a) Geben Sie eine Definition der Klasse NP an (ohne Begründung).

(b) Beschreiben Sie kurz und informell einen Algorithmus, der zeigt, dass SAT $\in$ PSPACE.

SAT
Eingabe: Aussagenlogische Formel F.
Frage: Ist F erfüllbar, d.h. gibt es eine $\{0,1\}$-wertige Belegung der in F verwendeten Booleschen Variablen derart, dass F zu wahr (d.h. 1) ausgewertet wird?

(c) Beschreiben Sie kurz und informell einen Algorithmus, der zeigt, dass TAUT $\in$ PSPACE.

TAUT
Eingabe: Aussagenlogische Formel F.
Frage: Ist F eine Tautologie, d.h. wird F für alle $\{0,1\}$-wertigen Belegungen der in F verwendeten Booleschen Variablen zu wahr (d.h. 1) ausgewertet?

(d) Geben Sie ein Inklusionsdiagramm an, das die Klassen P, PSPACE, coNP, DTIME(n^2) und NP enthält, und begründen Sie die angegebenen Inklusionen.

——————Lösung——————

(a) NP := $\bigcup_{k\geq 1}$ NTIME(n^k)

(b) Sei n die Anzahl der Variablen in F. Probiere jede der 2^n möglichen Variablenbelegungen aus und setze diese in F ein. Falls F für mindestens eine Belegung erfüllt wird, so akzeptiere F. Sonst lehne ab. Hierbei benutzen wir n Bandzellen zum Speichern der aktuellen Belegung und überschreiben diese jedes mal mit der nächsten Belegung. Der Platzbedarf ist also in $O(n + |F|)$, wobei $|F|$ die Länge der Kodierung von F ist.

(c) Analog zu (b): Probiere jede der 2^n möglichen Variablenbelegungen aus und setze diese in F ein. Falls F für alle Belegungen erfüllt wird, so akzeptiere F. Sonst lehne ab. Der Platzbedarf ist $O(n + |F|)$.

(d) Es gilt DTIME(n^2) $\subseteq$ P $= \bigcup_{k\geq 1}$ DTIME(n^k).

Es gilt P $\subseteq$ NP, da jede DTM auch eine NTM ist.

Es gilt P $\subseteq$ coNP, da coP = P. Sei $L \in$ P, dann gilt auch $\bar{L} \in$ P, also $\bar{L} \in$ NP und somit $L \in$ coNP.

Es gilt NP $\subseteq$ PSPACE. Eine DTM kann alle möglichen Zertifikate polynomieller Länge ausprobieren und jeweils in polynomieller Zeit überprüfen. Falls ein verifiziertes Zertifikat existiert, wird die Eingabe akzeptiert. Dafür benötigt sie auf dem Band nur polynomiell viele Zellen für das aktuelle Zertifikat, sowie höchstens polynomiell viele Zellen zum Verifizieren.

Es gilt coNP $\subseteq$ PSPACE (analog zu NP $\subseteq$ PSPACE). Falls alle Zertifikate verifiziert wurden, so wird die Eingabe akzeptiert. Dafür benötigt sie auf dem Band nur polynomiell viele Zellen für das aktuelle Zertifikat, sowie höchstens polynomiell viele Zellen zum Verifizieren.