Technische Universität Berlin, Berlin, 09.10.2017

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

Wiederholungsklausur: Berechenbarkeit und Komplexität

(Niedermeier/Froese/Molter, Sommersemester 2017)

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

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

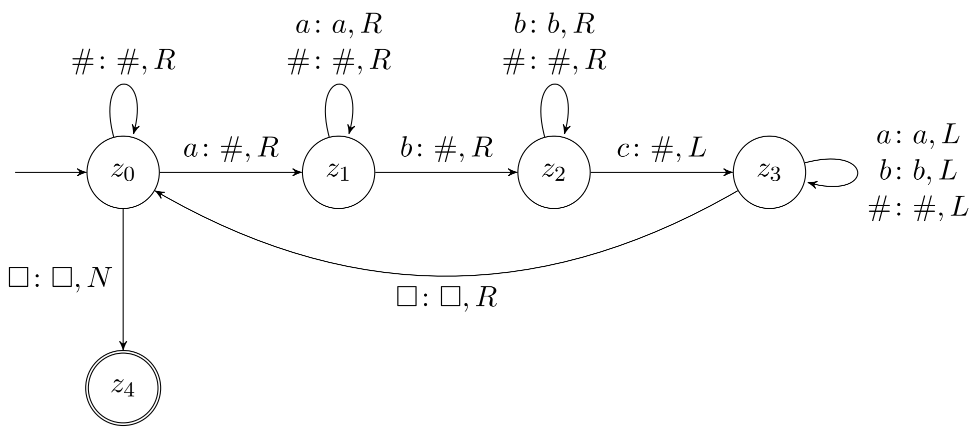
Name: .... **Matr.-Nr.: ....**

Aufgabe 1: **Turing-Maschinen** (5 + 5 Punkte)

(a) Vervollständigen Sie durch Eintragen in die vorgegebenen leeren Kästchen die Übergangsfunktion δ der folgenden deterministischen Turing-Maschine

$$M_1 = (\{z_0, z_1, z_2, z_3, z_4\}, \{a, b, c\}, \{a, b, c, \#, \Box\}, \delta, z_0, \Box, \{z_4\}),$$

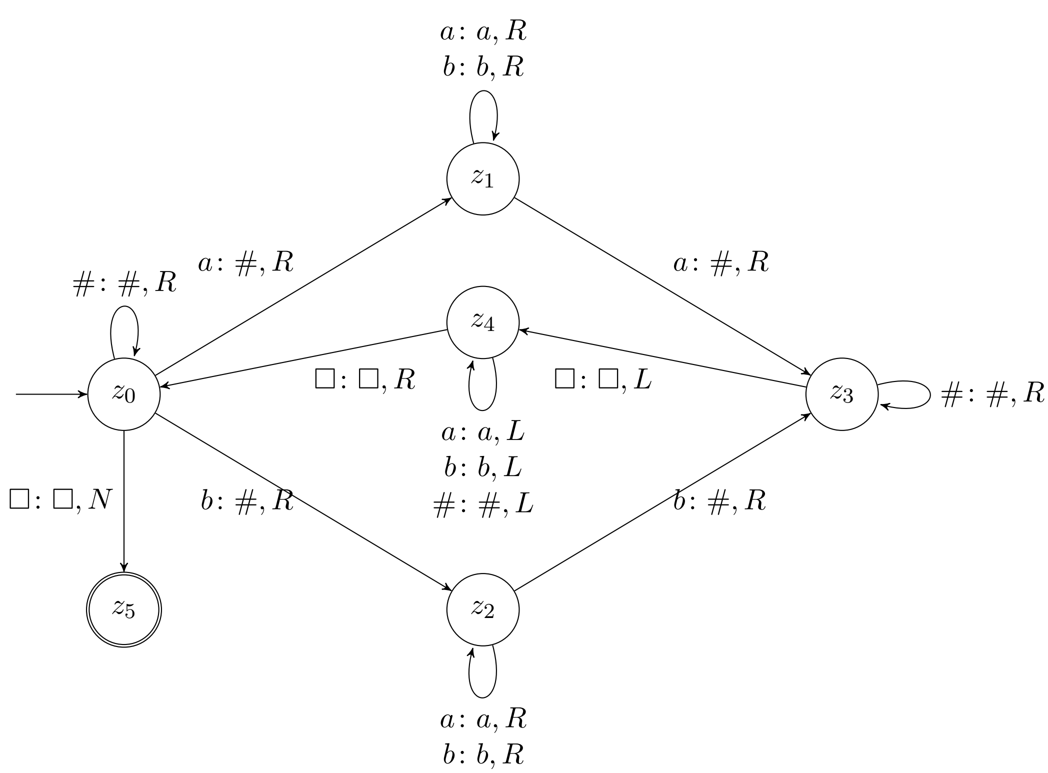
sodass $T(M_1) = \{a^n b^n c^n \mid n \geq 0\}$. Die Übergangsfunktion δ hat folgende graphische Darstellung:



(b) Vervollständigen Sie durch Eintragen in die vorgegebenen leeren Kästchen die Übergangsrelation δ der folgenden *nichtdeterministischen* Turing-Maschine

$$M_2 = (\{z_0, z_1, z_2, z_3, z_4, z_5\}, \{a, b\}, \{a, b, \#, \Box\}, \delta, z_0, \Box, \{z_5\}),$$

sodass $T(M_2) = \{ww^R \mid w \in \{a, b\}^*\}$, wobei das Wort w^R die Rückwärtsschreibweise von Wort w ist (also z.B. für $w = abb$ ist $w^R = bba$). Die Übergangsrelation δ hat folgende graphische Darstellung:



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

Aufgabe 2: **Die Komplexitätsklassen P und NP** (4 + 4 Punkte)

Im Folgenden sei Σ ein endliches Alphabet.

Begründen oder widerlegen Sie jeweils die beiden folgenden Aussagen.

(a) Seien A und B mit $B \subseteq A \subseteq \Sigma^*$ zwei Sprachen. Falls A in P liegt, so liegt auch B in P.

——————Lösung——————

Die Aussage ist falsch. Gegenbeispiel: Sei $A = \Sigma^* \in$ P und sei $B = H \subseteq A$ das Halteproblem. Dann ist B nicht in P enthalten, da das Halteproblem nicht entscheidbar ist.

————————————————————

(b) Für jede Sprache $A \subseteq \Sigma^*$ aus NP und für jede Sprache $B \subseteq \Sigma^*$ aus P gilt, falls $A \cap B$ und $A \cap (\Sigma^* \setminus B)$ in P liegen, so liegt A in P.

——————Lösung——————

Die Aussage ist korrekt. Beweis: Sei M_{AB} eine DTM, die $A \cap B$ in Polynomzeit $p(|x|)$ entscheidet und sei $M_{A\bar{B}}$ eine DTM, die $A \cap \bar{B}$ in Polynomzeit $q(|x|)$ entscheidet (beide TM'n existieren nach Voraussetzung).

Eine DTM M_A, die A in Polynomzeit entscheidet, arbeitet wie folgt. Zunächst simuliert M_A bei Eingabe x die TM M_{AB}. Falls M_{AB} die Eingabe x akzeptiert, so akzeptiert auch M_A, denn x liegt somit in A. Andernfalls wird $M_{A\bar{B}}$ auf x simuliert. Falls $M_{A\bar{B}}$ akzeptiert, so akzeptiert auch M_A wieder. Sonst wird x abgelehnt, da x in diesem Fall nicht in A liegt. Die Laufzeit beträgt $O(p(|x|) + q(|x|))$ und ist somit polynomiell.

————————————————————

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

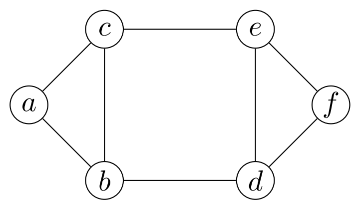
Aufgabe 3: **Graphbibliotheken** (2 + 6 Punkte)

Betrachten Sie die beiden folgenden Graphprobleme.

Dominating Set
Eingabe: Ein ungerichteter Graph G und eine Zahl $k > 0$.
Frage: Gibt es k Knoten in G, sodass jeder andere Knoten mindestens einen dieser k Knoten als Nachbarn hat?

Vertex Cover
Eingabe: Ein ungerichteter Graph G und eine Zahl $k > 0$.
Frage: Gibt es k Knoten in G, sodass jede Kante mindestens einen dieser Knoten als Endpunkt hat?

(a) Geben Sie ein kleinstmögliches Dominating Set und ein kleinstmögliches Vertex Cover für den abgebildeten Graphen an (ohne Begründung).



——————Lösung——————

Dominating Set: $\{c, d\}$
Vertex Cover: $\{b, c, d, e\}$

(b) Nehmen Sie an, dass ein Polynomzeitalgorithmus $\mathcal{A}$ existiert, der Dominating Set entscheidet. Geben Sie einen Algorithmus an, der Vertex Cover in polynomieller Zeit unter Benutzung von $\mathcal{A}$ entscheidet und begründen Sie dessen Korrektheit.

——————Lösung——————

Gegeben sei die VC-Instanz $(G = (V, E), k)$ (wir nehmen hierbei an, dass G zusammenhängend ist).

Konstruiere einen neuen Graphen $G' = (V', E')$ aus G, indem für jede Kante $\{u, v\} \in E$ in G ein neuer Knoten hinzugefügt und mit beiden Endpunkten u und v verbunden wird (in $O(|E|)$ Zeit). Nun wird Algorithmus $\mathcal{A}$ auf der Instanz (G', k) laufen gelassen und die resultierende Antwort zurückgegeben. Die Laufzeit ist somit insgesamt polynomiell.

Zur Korrektheit:

Falls G ein Vertex Cover $C \subseteq V$ der Größe höchstens k besitzt, dann ist C auch ein Dominating Set für G'. Dies folgt daraus, dass G zusammenhängend ist (und somit jeder Knoten an mindestens einer Kante anliegt) und dass jede Kante aus E einen Endpunkt in C hat. Also hat jeder Knoten in $V \setminus C$ einen Nachbarn in C. Zudem hat auch jeder neu eingefügte Knoten in V' per Konstruktion einen Nachbarn in C.

Sei $D \subseteq V'$ ein Dominating Set der Größe höchstens k für G'. Wir können annehmen, dass $D \subseteq V$ gilt, da statt einem neu eingefügten Knoten auch ein beliebiger Nachbar dieses Knotens gewählt werden kann. Dann ist D auch ein Vertex Cover für G, denn für jede Kante aus E ist mindestens ein Endpunkt in D enthalten, da sonst der neu eingefügte Knoten für diese Kante nicht dominiert wäre.

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

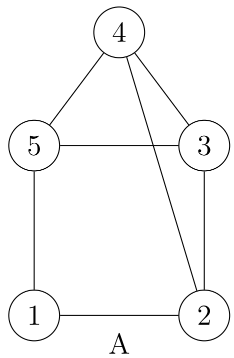
Aufgabe 4: **Polynomzeitreduktion** (3 + 2 + 1 + 3 + 3 Punkte)

Betrachten Sie die folgenden Probleme.

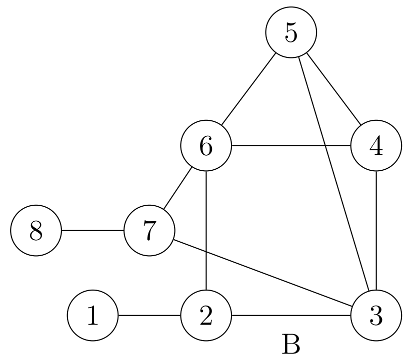
Hamiltonkreis
Eingabe: Ein ungerichteter Graph $G = (V, E)$.
Frage: Gibt es einen Kreis in G, der jeden Knoten aus V genau einmal enthält?

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

(a) Kennzeichnen Sie durch Nummerierung der Knoten in Graph A einen *Hamiltonkreis* und in Graph B einen *Hamiltonpfad*.



A



B

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

i. drei Knoten zum Eingabegraph hinzufügen, von denen zwei Knotengrad eins haben, und diese geeignet mit den restlichen Knoten verbinden,
ii. begründen, dass f in Polynomzeit berechnet werden kann,
iii. zeigen, dass für alle Graphen G gilt: $G \in$ Hamiltonkreis $\Rightarrow f(G) \in$ Hamiltonpfad und
iv. zeigen, dass für alle Graphen G gilt: $f(G) \in$ Hamiltonpfad $\Rightarrow G \in$ Hamiltonkreis.

——————Lösung——————

i. Sei $G = (V, E)$ ein ungerichteter Graph mit $V = \{v_1, \ldots, v_n\}$. Wir konstruieren den Graphen $f(G)$ durch Hinzufügen der Knoten x, y und z, wobei x mit allen Nachbarn von v_1 verbunden wird, y mit x verbunden wird, und z mit v_1 verbunden wird.
ii. Insgesamt werden also höchstens 3 Knoten und $n+2$ Kanten hinzugefügt, was in $O(n)$ Zeit möglich ist.
iii. Wir nehmen an, dass die Knoten $v_1, \ldots, v_n$ in dieser Reihenfolge einen Hamiltonkreis bilden, d.h. v_1 hat die Nachbarn v_2 und v_n. Dann existiert in $f(G)$ per Konstruktion der folgende Hamiltonpfad: $z, v_1, v_2, \ldots, v_n, x, y$.
iv. Falls in $f(G)$ ein Hamiltonpfad existiert, dann müssen z und y die Endpunkte sein, da sie vom Grad eins sind. Wir nehmen also an, dass $z, v_1, v_{i_2} \ldots, v_{i_n}, x, y$ ein Hamiltonpfad ist (wobei $i_2, \ldots, i_n$ eine Permutation der Zahlen in $\{2, \ldots, n\}$ sei). Da v_{i_n} mit x verbunden ist, ist v_{i_n} also ein Nachbar von v_1. Somit bilden die Knoten $v_1, v_{i_2} \ldots, v_{i_n}$ einen Hamiltonkreis in G.

Aufgabe 5: **Vermischtes zu Komplexitätsklassen** (4+4+4 Punkte)

(a) Sei A eine NP-vollständige Sprache. Zeigen Sie, dass $\overline{A}$ coNP-vollständig ist.

——————Lösung——————

Da $A \in \text{NP}$, gilt $\overline{A} \in \text{coNP}$ (nach Definition von coNP).

Sei $L \in \text{coNP}$. Somit gilt $\overline{L} \in \text{NP}$ und $\overline{L} \leq_m^p A$. Es existiert also eine totale, polynomiell berechenbare Funktion f, sodass

$$x \in \overline{L} \Leftrightarrow f(x) \in A.$$

Somit gilt auch

$$x \notin \overline{L} \Leftrightarrow f(x) \notin A,$$

was äquivalent ist zu

$$x \in L \Leftrightarrow f(x) \in \overline{A}.$$

Es gilt also $L \leq_m^p \overline{A}$.

(b) Sei A eine NP-vollständige Sprache. Zeigen Sie, dass „$A \in \text{P} \Rightarrow \text{coNP} = \text{P}$" gilt.

——————Lösung——————

Da A NP-vollständig ist, gilt:

$$A \in \text{P} \Rightarrow \text{P} = \text{NP} \Rightarrow \text{coP} = \text{coNP}.$$

Da $\text{P} = \text{coP}$, gilt also $\text{coNP} = \text{P}$.

(c) Begründen oder widerlegen Sie kurz die folgende Behauptung:

Unter der Annahme $\text{P} \neq \text{NP}$ gilt, dass Clique auf Graphen mit maximalem Knotengrad 17 NP-vollständig ist.

Clique
Eingabe: Ein ungerichteter Graph G und eine Zahl $k > 0$.
Frage: Hat G einen vollständigen Teilgraphen mit mindestens k Knoten?

——————Lösung——————

Die Behauptung ist falsch, da Clique auf Graphen mit Maximalgrad 17 polynomzeitlösbar ist. Somit kann es nicht NP-vollständig sein, da sonst $\text{P} = \text{NP}$ folgen würde.

Bei maximalem Knotengrad 17 kann eine Clique höchstens 18 Knoten beinhalten. Ein Algorithmus kann also einfach ablehnen, falls $k > 18$. Für $k \leq 18$ iteriere über alle Knotenteilmengen der Größe k und teste, ob diese eine Clique bilden. Die Laufzeit hierfür liegt in $O(n^{18})$.
