

**1. Klausur zur Vorlesung
Theoretische Grundlagen der Informatik
Wintersemester 2014/2015**

Hier Aufkleber mit Name und Matrikelnummer anbringen	
Vorname:	_____
Nachname:	_____
Matrikelnummer:	_____

Beachten Sie:

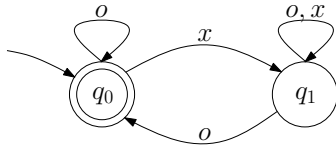
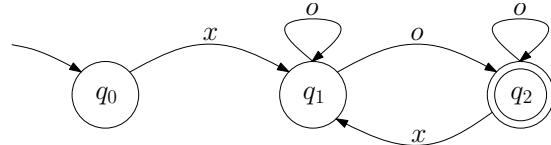
- Bringen Sie den Aufkleber mit Ihrem Namen und Matrikelnummer auf diesem Deckblatt an und beschriften Sie jedes Aufgabenblatt mit Ihrem Namen und Matrikelnummer.
- Schreiben Sie die Lösungen auf die Aufgabenblätter und Rückseiten. Zusätzliches Papier erhalten Sie bei Bedarf von der Aufsicht.
- Zum Bestehen der Klausur sind **20** der möglichen **60** Punkte hinreichend.
- Es sind keine Hilfsmittel zugelassen.

Aufgabe	Mögliche Punkte						Erreichte Punkte					
	a	b	c	d	e	Σ	a	b	c	d	e	Σ
1	2	2	3	-	-	7				-	-	
2	2	2	2	-	-	6				-	-	
3	2	4	2	-	-	8				-	-	
4	1	3	1	-	-	5				-	-	
5	1	1	1	3	2	8						
6	1	3	3	1	-	8					-	
7	5	3	-	-	-	8			-	-	-	
8	4	-	-	-	-	4		-	-	-	-	
9	6 × 1					6						
Σ						60						

Problem 1: Endliche Automaten

2+2+3=7 Punkte

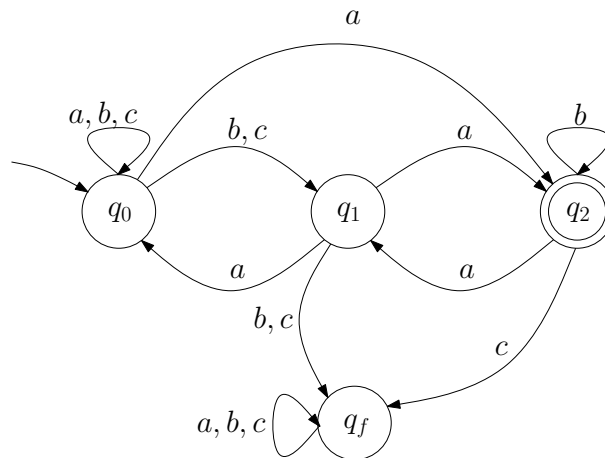
Gegeben seien die endlichen Automaten $\mathcal{A}_1 = (Q = \{q_0, q_1\}, \Sigma = \{o, x\}, \delta_1, q_0, \{q_0\})$ und $\mathcal{A}_2 = (Q = \{q_0, q_1, q_2\}, \Sigma = \{o, x\}, \delta_2, q_0, \{q_2\})$, wobei δ_1 und δ_2 durch folgende Zustandsdiagramme gegeben sind. Der Fehlerzustand ist implizit gegeben.

Endlicher Automat \mathcal{A}_1 mit δ_1 Endlicher Automat \mathcal{A}_2 mit δ_2

- (a) Geben Sie jeweils einen regulären Ausdruck für die Sprache an, die von \mathcal{A}_1 bzw. \mathcal{A}_2 akzeptiert wird.
Hinweis: Hier ist nicht verlangt, dass Sie das Verfahren aus der Vorlesung verwenden.

- (b) Konstruieren Sie einen endlichen Automaten mit maximal 5 Zuständen, der die Sprache $L = L(\mathcal{A}_1) \cup L(\mathcal{A}_2)$ akzeptiert.

- (c) Gegeben sei der endliche Automat $\mathcal{A}_3 = (Q = \{q_0, q_1, q_2, q_f\}, \Sigma = \{a, b, c\}, \delta_3, q_0, \{q_2\})$, wobei δ_3 durch folgendes Zustandsdiagramm gegeben ist.



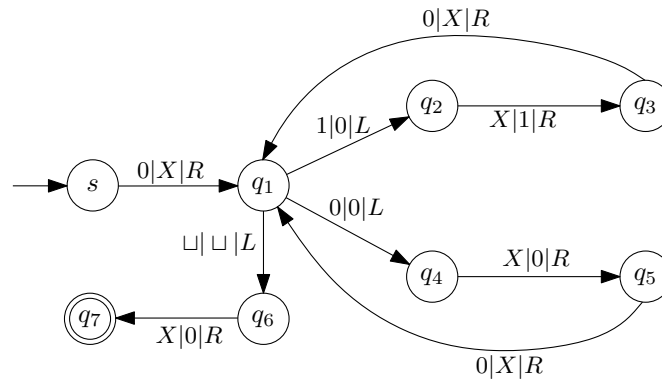
Geben Sie zu \mathcal{A}_3 einen äquivalenten vollständigen deterministischen endlichen Automaten tabellarisch an und verwenden Sie hierfür die Potenzmengenkonstruktion.

Problem 2: Turingmaschine

2+2+2=6 Punkte

Gegeben sei die folgende Turingmaschine

$$\mathcal{M} = (Q = \{s, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}, \Sigma = \{0, 1\}, \Gamma = \Sigma \cup \{\sqcup, X\}, \delta, s, F = \{q_7\}).$$

Die Übergangsfunktion δ von \mathcal{M} ist durch das untenstehende Zustandsübergangsdiagramm dargestellt.

Für \mathcal{M} gelten folgende **Konventionen**: \mathcal{M} hält und akzeptiert sobald ein Endzustand erreicht wird. Alle Übergänge, die aus Endzuständen herausführen, werden demnach nicht verwendet. \mathcal{M} hält und **akzeptiert nicht** für nicht dargestellte Zustandsübergänge.

- (a) Dokumentieren Sie die Berechnung des Wortes 01101. Geben Sie dazu für **jeden** Schritt die aktuelle Konfiguration an.

- (b) Geben Sie die Sprache $L(\mathcal{M}) \subseteq \Sigma^*$ an, die \mathcal{M} akzeptiert. Geben Sie außerdem die Funktion $f_{\mathcal{M}}: \Sigma^* \rightarrow \Gamma^*$ an, die \mathcal{M} realisiert. Begründen Sie Ihre Antwort, indem Sie das Verhalten der Turingmaschine auf einer Eingabe informell beschreiben.

- (c) Geben Sie eine Turingmaschine \mathcal{M}' an, die folgende Eigenschaften besitzt.

- \mathcal{M}' akzeptiert genau die Sprache $L(\mathcal{M})$.
- \mathcal{M}' realisiert dieselbe Funktion wie \mathcal{M} .
- \mathcal{M}' hat maximal drei Zustände.

Hinweis: Sie dürfen dieselben Konventionen verwenden, wie sie für \mathcal{M} angegeben sind.

Problem 3: Reguläre Sprachen

2+4+2=8 Punkte

- (a) Zeigen oder widerlegen Sie, dass die Sprache $L_1 = \{a^n b^k a^n \in \{a, b\}^* \mid k \in \mathbb{N}_0, n \in \mathbb{N} \setminus \{0\}, n \leq 3\}$ regulär ist.

- (b) Zeigen oder widerlegen Sie, dass die Sprache $L_2 = \{a^{i!} \in \{a\}^* \mid i \in \mathbb{N}\}$ regulär ist.

- (c) Zeigen Sie mithilfe eines Gegenbeispiels, dass die Umkehrung des Pumping-Lemmas nicht gilt. Begründen Sie kurz Ihre Antwort.

Problem 4: Berechenbarkeit – Halteproblem

1+3+1=5 Punkte

- (a) Sei $L_1 = \{\langle \mathcal{M} \rangle \mid \text{Die Turingmaschine } \mathcal{M} \text{ berechnet das Spiegelwort der Eingabe}\}$. Zeigen oder widerlegen Sie, dass L_1 entscheidbar ist.

- (b) Gegeben sei die folgende Sprache über dem Alphabet $\Sigma = \{0, 1\}$

$L_2 = \{(v, w) \in \Sigma^* \times \Sigma^* \mid \text{Die Turingmaschinen } T_v \text{ und } T_w \text{ halten auf exakt denselben Eingaben}\}$.

Dabei bezeichnet T_w die Turingmaschine, die durch die Gödelnummer w codiert ist. Zeigen Sie, dass L_2 nicht entscheidbar ist.

Hinweis: Sie dürfen verwenden, dass die Sprache $L_3 = \{w \mid T_w \text{ hält auf keiner Eingabe}\}$ nicht entscheidbar ist.

- (c) Sei L_4 eine unentscheidbare Sprache. Zeigen oder widerlegen Sie folgende Aussage: Jede Teilmenge von L_4 ist ebenfalls unentscheidbar.

Problem 5: CLIQUE – NP-Vollständigkeit

1+1+1+3+2=8 Punkte

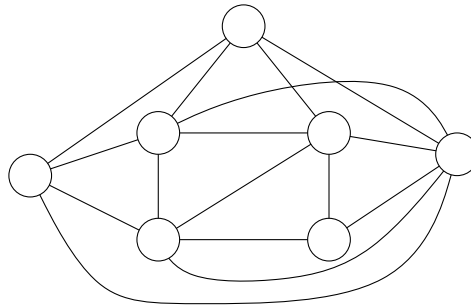
Sei $G = (V, E)$ ein ungerichteter Graph. Die Menge $C \subseteq V$ heißt *Clique* von G , falls für alle Knoten $u, v \in C$ mit $u \neq v$ gilt $\{u, v\} \in E$. Das NP-vollständige Problem CLIQUE ist dann wie folgt definiert:

Problem CLIQUE

Gegeben: Ungerichteter Graph $G = (V, E)$ und Parameter k .

Frage: Existiert eine Clique C von G mit $|C| \geq k$?

- (a) Existiert eine Clique C mit $|C| \geq 4$ in dem unten abgebildeten Graphen? Falls ja, dann markieren Sie die Knoten einer solchen Clique. Andernfalls begründen Sie Ihre Antwort.



- (b) Geben Sie für CLIQUE das entsprechende Optimierungs- und Optimalwertproblem an.

- (c) Zeigen Sie, dass wenn CLIQUE in der Komplexitätsklasse P liegt, dann gilt $P = NP$.

(d) Das Problem GANZZAHLIGESPROGRAMM ist wie folgt definiert.

Problem GANZZAHLIGESPROGRAMM

Gegeben: $a_{ij} \in \mathbb{Z}_0, b_i, c_j \in \mathbb{Z}_0, 1 \leq i \leq m, 1 \leq j \leq n, B \in \mathbb{Z}_0$

Frage: Existieren $x_1, \dots, x_n \in \mathbb{N}_0$, sodass

$$\sum_{j=1}^n c_j \cdot x_j = B \quad \text{und} \quad \sum_{j=1}^n a_{ij} \cdot x_j \leq b_i \quad \text{für } 1 \leq i \leq m?$$

Geben Sie eine polynomielle Reduktion von dem Entscheidungsproblem CLIQUE auf GANZZAHLIGESPROGRAMM an. Geben Sie insbesondere die Interpretation der eingeführten Variablen an und begründen Sie die Korrektheit der Reduktion.

- (e) Welches in der Vorlesung vorgestellte Entscheidungsproblem modelliert das unten stehende ganzzahlige Programm für einen gegebenen Graphen $G = (V, E)$? Begründen Sie Ihre Antwort.

Ganzzahliges Programm WASMAGDASWOHLSEIN:

Für jeden Knoten $u \in V$ gibt es die Variablen x_a^u , x_b^u und x_c^u . Außerdem sollen folgende Ungleichungen gelten.

$$\begin{array}{llll} \text{Für alle } u \in V: & x_a^u + x_b^u + x_c^u \leq 1 & -x_a^u - x_b^u - x_c^u \leq -1 & (1) \\ \text{Für alle } \{u, v\} \in E: & x_a^u + x_a^v \leq 1 & x_b^u + x_b^v \leq 1 & x_c^u + x_c^v \leq 1 & (2) \\ \text{Für alle } u \in V: & x_a^u \leq 1 & x_b^u \leq 1 & x_c^u \leq 1 & (3) \\ \text{Für alle } u \in V: & -x_a^u \leq 0 & -x_b^u \leq 0 & -x_c^u \leq 0 & (4) \end{array}$$

Problem 6: HITTINGSET – Approximation

1+3+3+1=8 Punkte

Das Problem 4-HITTINGSET ist wie folgt definiert:

Problem 4-HITTINGSET

Gegeben: Universum $\mathcal{U} = \{1, \dots, n\}$ und Mengensystem $\mathcal{C} \subseteq 2^{\mathcal{U}}$, wobei $2^{\mathcal{U}}$ die Potenzmenge von \mathcal{U} bezeichnet. Jede Menge $A \in \mathcal{C}$ enthalte maximal 4 Elemente, d.h. $|A| \leq 4$.

Gesucht: Hitting Set X für \mathcal{C} mit minimaler Kardinalität.

Hinweis: Eine Menge $X \subseteq \mathcal{U}$ heißt *Hitting Set* für \mathcal{C} , falls für jedes $A \in \mathcal{C}$ gilt $A \cap X \neq \emptyset$.

Betrachten Sie folgenden Algorithmus:

Algorithmus 1 : HittingSetApproximation

Eingabe : $\mathcal{U} = \{1, \dots, n\}$, $\mathcal{C} \subseteq 2^{\mathcal{U}}$ mit $|A| \leq 4$ für alle $A \in \mathcal{C}$.

Ausgabe : Hitting Set X für \mathcal{C} .

```

1  $X \leftarrow \emptyset$ 
2 solange  $\mathcal{C} \neq \emptyset$  tue
3    $A \leftarrow$  wähle eine beliebige Menge  $A \in \mathcal{C}$ .
4    $X \leftarrow X \cup A$ 
5    $\mathcal{C} \leftarrow \mathcal{C} \setminus \{A \in \mathcal{C} \mid A \cap X \neq \emptyset\}$ 

```

Für eine beliebige Instanz I von 4-HITTINGSET, bezeichne $\mathcal{A}(I)$ die Kardinalität des von Algorithmus 1 zurückgegebenen Hitting Sets, sowie $\text{OPT}(I)$ die Kardinalität einer optimalen Lösung für I .

(a) Geben Sie die Menge X an, die Algorithmus 1 bei Eingabe der folgenden Instanz berechnet:

$$\mathcal{U} = \{1, 2, 3, 4, 5, 6, 7\}, \quad \mathcal{C} = \{c_1 = \{2, 3, 6\}, c_2 = \{1, 3, 4, 7\}, c_3 = \{4, 6, 7\}, c_4 = \{1, 5\}\}.$$

In Zeile 3 werde jeweils die Menge c_i mit dem kleinsten Index i unter allen Mengen in \mathcal{C} gewählt.

$X =$ _____

(b) Geben Sie eine Familie $I_1 = (\mathcal{U}_1, \mathcal{C}_1), I_2 = (\mathcal{U}_2, \mathcal{C}_2), I_3 = (\mathcal{U}_3, \mathcal{C}_3), \dots$ von Instanzen für 4-HITTINGSET an, sodass gilt $|\mathcal{U}_k| \in \Omega(k)$, $|\mathcal{C}_k| \in \Omega(k)$ und $\mathcal{A}(I_k) = 4 \text{OPT}(I_k)$ für alle $k \in \mathbb{N}$. Begründen Sie Ihre Antwort.

- (c) Betrachten Sie eine (feste) optimale Lösung X^* . Zeigen Sie, dass jede in Zeile 3 ausgewählte Menge A mindestens ein Element $z \in X^* \setminus X$ enthält.

- (d) Zeigen Sie, dass für jede beliebige Instanz I von 4-HITTINGSET gilt $\mathcal{A}(I) \leq 4 \text{OPT}(I)$.

Problem 7: Grammatiken und Chomsky-Hierarchien

5+3=8 Punkte

Gegeben sei ein Konzept einer imperativen Programmiersprache namens PABA. Aus dieser Programmiersprache ist folgender Auszug gegeben.

```
1 i = 30.  
2  
3 DO 3 TIMES.  
4     i = i + 1.  
5 ENDDO.
```

Ein PABA-Programm ist eine Folge von *Anweisungen*. Jede Anweisung endet mit einem '.'. PABA unterscheidet verschiedene Anweisungen.

Eine *Zuweisung* ist eine Anweisung, siehe Zeile 1 und Zeile 4. Sie besteht aus einer linken und einer rechten Seite, die durch ein '=' getrennt sind. Die linke Seite einer Zuweisung ist eine *Variable*. Die rechte Seite einer Zuweisung ist ein *Term*. Ein Term ist eine Variable, *Zahl* oder eine *Addition* zweier Terme, siehe Zeile 4. Eine Zahl ist nicht negativ. Eine Variable beginnt immer mit einem *Buchstaben* gefolgt von beliebig vielen Buchstaben und Zahlen. Eine Addition besteht aus zwei Termen getrennt durch das Symbol '+'.
.

Eine *do-Anweisung* beginnt mit dem Schlüsselwort DO gefolgt von einer *Bedingung*, siehe Zeile 3. Eine *Bedingung* besteht aus einer Variablen oder Zahl gefolgt vom Schlüsselwort TIMES.
.

Der *do-Block* ist eine Anweisung, die aus einer Folge von Anweisungen besteht, siehe Zeile 3–5: Zuerst kommt eine do-Anweisung gefolgt von beliebig vielen Anweisungen. Ein do-Block endet immer mit dem Schlüsselwort ENDDO.
.

Hinweis: Für die bessere Lesbarkeit enthält das oben angegebene Beispielprogramm Leerzeichen und Zeilenumbrüche. Diese sind nicht Teil der Grammatik.

- (a) Entwerfen Sie eine Grammatik, welche die Sprache aller möglichen PABA-Programme erzeugt. Vervollständigen Sie hierzu die folgende Grammatik $G = (\Sigma, V, S, R)$.

Die Produktionen von G sind bereits teilweise unten gegeben. Die Nichtterminale aus V haben die Form $\langle Name \rangle$. Alle anderen Symbole in der folgenden Regelmenge sind Terminale, wobei ε das leere Wort bezeichnet. Wenn nötig, führen Sie weitere Terminale ein.

$S \rightarrow$

$\langle Anweisung \rangle \rightarrow$

$\langle do - Block \rangle \rightarrow$

$\langle do - Anweisung \rangle \rightarrow$

$\langle Bedingung \rangle \rightarrow$

$\langle Zuweisung \rangle \rightarrow$

$\langle Term \rangle \rightarrow$

$\langle Variable \rangle \rightarrow$

$\langle HilfsVariable \rangle \rightarrow$

$\langle Zahl \rangle \rightarrow \langle Ziffer \rangle \langle Zahl \rangle \mid \langle Ziffer \rangle$

$\langle Ziffer \rangle \rightarrow 0 \mid 1 \mid 2 \mid \dots \mid 9$

$\langle Buchstabe \rangle \rightarrow a \mid \dots \mid z \mid A \mid \dots \mid Z$

- (b) Vervollständigen Sie die unten angegebene Tabelle, indem Sie für jeden Chomsky-Typ k die Sprachklasse angeben, die genau die Sprachen enthält, die von Chomsky- k -Grammatiken erzeugt werden. Geben Sie zudem für jede Sprachklasse ein Rechnermodell an, das die von Chomsky- k -Grammatiken, aber nicht die von Chomsky- $(k - 1)$ -Grammatiken erzeugten Sprachen akzeptiert. Geben Sie außerdem jeweils die Entscheidbarkeit des Wortproblems an.

Chomsky-Typ	Sprachklasse	Rechnermodell	Entscheidbarkeit des Wortproblems
Typ-0			
Typ-1			
Typ-2			
Typ-3			

Problem 8: Kellerautomaten

4 Punkte

Gegeben seien die folgenden zwei Sprachen:

$$L_1 = \{a^n b^2 c^{3n} \in \{a, b, c\}^* \mid n \in \mathbb{N} \setminus \{0\}\},$$

$$L_2 = \{a^n b^{2n} c^{3n} \in \{a, b, c\}^* \mid n \in \mathbb{N}\}.$$

Geben Sie für jede der beiden Sprachen an, ob ein Kellerautomat existiert, der die Sprache erkennt. Sollte ein solcher Kellerautomat existieren, so geben Sie einen an, der maximal 4 Zustände besitzt und mit leerem Stack akzeptiert. Andernfalls geben Sie den maximalen Chomsky-Typ einer Grammatik an, welche die Sprache erzeugt.

Problem 9: Verschiedenes

6 Punkte

Jeder der folgenden Aussagenblöcke umfasst mehrere Einzelaussagen. Die sich unterscheidenden Aussagenbausteine sind durch Kästchen gekennzeichnet. Kreuzen Sie genau jene Bausteine an, die in einer wahren Einzelaussage enthalten sind. Jeder Aussagenblock enthält mindestens eine wahre Einzelaussage. Unvollständig oder falsch angekreuzte Aussagenblöcke werden mit null Punkten bewertet. Sie erhalten einen Punkt für jeden Aussagenblock, für den Sie genau die richtige Menge an Aussagenbausteinen angekreuzt haben.

Falls $P \neq NP$ gilt, dann

- gibt es keinen absoluten polynomiellen Approximationsalgorithmus für das Problem CLIQUE.
- gibt es keinen absoluten polynomiellen Approximationsalgorithmus für das Problem KNAPSACK.
- kann für einen Graphen $G = (V, E)$ und eine Knotenmenge $V' \subseteq V$ nicht in polynomieller Zeit überprüft werden, ob V' eine Clique in G ist.
- hält eine Turingmaschine, die eine semientscheidbare Sprache akzeptiert, nicht notwendigerweise auf jeder Eingabe.

Für jede Chomsky-2-Grammatik G in Chomsky-Normalform

- gibt es einen nicht-deterministischen endlichen Automaten, der $L(G)$ akzeptiert.
- kann der Cocke-Younger-Kasami Algorithmus für die Überprüfung, ob ein Wort in $L(G)$ liegt, verwendet werden.
- gibt es eine Turingmaschine, die $L(G)$ akzeptiert.
- enthält $L(G)$ unendlich viele Wörter.

Seien L_1 und L_2 zwei entscheidbare Sprachen über dem Alphabet Σ , dann ist auch die Sprache

- $L_1 \cup L_2$
- $L_1 \cap L_2$
- $L_1 \cdot L_2$
- $(\Sigma^* \setminus L_1) \cup L_2$

entscheidbar.

Jede kontextfreie Grammatik G , die nicht das leere Wort erzeugt,

- besitzt für jedes ableitbare Wort einen eindeutigen Syntaxbaum.
- kann in eine Grammatik G' in Chomsky-Normalform überführt werden, sodass $L(G) = L(G')$.
- kann in einen endlichen Automaten \mathcal{A} überführt werden, der die Sprache $L(G)$ akzeptiert.
- kann in eine Grammatik G' in Greibach-Normalform überführt werden, sodass $L(G) = L(G')$.

Jeder deterministische Kellerautomat $\mathcal{A} = (Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)$, der durch einen Endzustand akzeptiert, hat die Eigenschaft,

- dass für alle $q \in Q$, $a \in \Sigma$ und $Z \in \Gamma$ gilt $|\delta(q, a, Z)| + |\delta(q, \varepsilon, Z)| > 1$.
- dass für alle $q \in Q$, $a \in \Sigma$ und $Z \in \Gamma$ gilt $|\delta(q, a, Z)| \leq 2$.
- dass \mathcal{A} die Eingabe akzeptiert, wenn \mathcal{A} in einen Zustand aus F wechselt.
- dass er nicht die Sprache $L = \{ww^R \mid w \in \{0, 1\}^*\}$ akzeptiert, wobei w^R das Spiegelwort von w bezeichnet.

Für die Entropie einer diskreten, endlichen Zufallsvariable mit n Zeichen gilt:

- Sie wird maximal, wenn alle Zeichen gleichwahrscheinlich sind.
- Sie beträgt maximal $\log_2(n)$.
- Sie beträgt maximal $1/\log_2(n)$.
- Sie beträgt $n \cdot \log_2(n)$.