1. Klausur zur Vorlesung
Theoretische Grundlagen der Informatik
Wintersemester 2024/2025

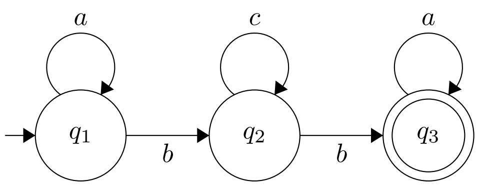
Hier Aufkleber mit Matrikelnummer anbringen

- Bringen Sie den Aufkleber mit Ihrer Matrikelnummer auf diesem Deckblatt an und beschriften Sie jedes Aufgabenblatt mit Ihrer Matrikelnummer.
- Schreiben Sie die Lösungen auf die Aufgabenblätter und Rückseiten. Am Ende der Klausur sind zusätzliche Leerseiten. Fordern Sie zusätzliches Papier bitte nur an, wenn nötig.
- Die Tackernadel darf nicht gelöst werden, Sie dürfen allerdings die NP-vollständigen Probleme im Anhang abtrennen.
- Begründen/Beweisen Sie Ihre Antworten ausreichend, wenn "Zeigen/Beweisen Sie, dass" gefordert wird.
- Als Hilfsmittel ist ein handbeschriebenes A4-Papier erlaubt.
- Einlesezeit: 15 min
  Bearbeitungszeit: 2 h

	Mögliche Punkte						Erreichte Punkte					
	a	b	c	d	e	Σ	a	b	c	d	e	Σ
Aufg. 1	1	2	2	4	–	9					–	
Aufg. 2	3	2	3	2	–	10					–	
Aufg. 3	1	4	2	–	–	7				–	–	
Aufg. 4	2	3	3	4	–	12					–	
Aufg. 5	2	2	7	–	–	11				–	–	
Aufg. 6	1	5	5	–	–	11				–	–	
Σ						60						

Problem 1: Warmup $1 + 2 + 2 + 4 = 9$ Punkte

Betrachten Sie den folgenden Automaten $\mathcal{A}$ mit implizitem Müllzustand über dem Alphabet $\{a, b, c\}$ und mit Startzustand q_1.



(a) Geben Sie für die Wörter $abccbab, abcccba$ jeweils an, ob sie zur Sprache $\mathcal{L}(\mathcal{A})$ gehören.

(b) Geben Sie einen DEA an, der $\mathcal{L}(\mathcal{A})^c$ akzeptiert.

(c) Geben Sie einen regulären Ausdruck α an, sodass $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\alpha)$.

(d) Wir definieren die Quotientensprache aus der Vorlesung wie gewohnt als

$$L_1/L_2 := \{w \in \Sigma^* \mid \exists z \in L_2 \text{ mit } w \cdot z \in L_1\}.$$

Geben Sie einen regulären Ausdruck β für $\mathcal{L}(\mathcal{A})/\{ba\}$ an. Beweisen Sie, dass $\mathcal{L}(\beta) = \mathcal{L}(\mathcal{A})/\{ba\}$.

Problem 2: Kontextfreie Sprachen $3 + 2 + 3 + 2 = 10$ Punkte

Gegeben sei die Sprache $L := \{a^n b^m c^n b^\ell \mid n, m, \ell \geqslant 0\}$.

(a) Zeigen Sie, dass die Sprache L nicht regulär ist.

(b) Geben Sie eine kontextfreie Grammatik G an, die die Sprache L erzeugt.

(c) Geben Sie einen nichtdeterministischen Kellerautomaten $\mathcal{A}$ an, sodass $L = \mathcal{L}(\mathcal{A})$.

(d) Zeigen Sie, dass es eine Sprache $L' \subseteq L$ gibt, sodass L' nicht kontextfrei ist.

Hinweis: Benutzen Sie eine bereits bekannte nicht-kontextfreie Sprache aus der Vorlesung/Übung.

Problem 3: Approximationsalgorithmen $1 + 4 + 2 = 7$ Punkte

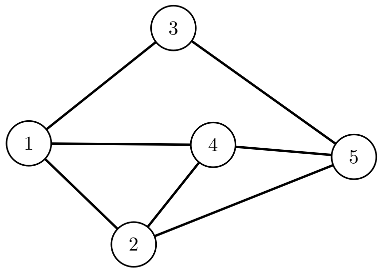
In dieser Aufgabe betrachten wir das Problem VERTEX COVER: Gegeben einen Graphen, ist das Ziel, ein kleinstmögliches *Vertex Cover* zu finden, also eine Menge von Knoten, sodass jede Kante des Graphen mindestens einen der beiden Endpunkte in der Menge hat.

> VERTEX COVER
> **Gegeben:** ein ungerichteter Graph $G = (V, E)$.
> **Aufgabe:** Berechne die Größe eines minimalen Vertex Covers $C \subseteq V$, sodass für jede Kante $\{u, v\} \in E$ gilt: $u \in C$ oder $v \in C$.

Außerdem geben wir die folgenden Definition eines *maximalen Matching*:

> **Definition**: Matching
> Sei $G = (V, E)$ ein ungerichteter Graph. Ein *Matching* M ist eine Menge von Kanten, sodass keine zwei Kanten aus M einen gemeinsamen Endpunkt haben. Das heißt für alle e und e' aus M gilt: $e \cap e' = \emptyset$.
> Wir nennen ein Matching M *maximal* (nicht erweiterbar), wenn wir keine weitere Kante hinzufügen können: Es gilt für alle $e \in E - M$: $M \cup \{e\}$ ist kein Matching mehr.

(a) Geben Sie für den folgenden Graphen ein minimales Vertex Cover sowie ein maximales Matching an.



Gegeben sei der folgende Approximationsalgorithmus für VERTEX COVER.

Algorithmus $\mathcal{A}$: GREEDY VERTEX COVER

Input: $G = (V, E)$
Berechne ein maximales Matching M in G
for $\{u, v\} \in M$ **do**
  Füge u und v zu C hinzu
return $|C|$

(b) Zeigen Sie, dass der Algorithmus Greedy Vertex Cover eine Approximation mit relativer Gütegarantie 2 erreicht.

(c) Sei $n \in \mathbb{N}$ beliebig. Zeigen Sie, dass es einen Graphen mit n Knoten gibt, sodass ein minimales Vertex Cover Größe OPT hat, aber Greedy Vertex Cover als Ergebnis $2 \cdot OPT$ zurückliefert.

Problem 4: Zeige oder Widerlege $2 + 3 + 3 + 4 = 12$ Punkte

Zeigen oder widerlegen Sie folgende Aussagen:

(a) Im Folgenden bezeichnen wir mit T_w die Turingmaschine mit Gödelnummer w.

Aussage: Die Sprache $\{w \in \{0,1\}^* : |w| < 2025 \text{ und } T_w \text{ akzeptiert } w \text{ nicht}\}$ ist entscheidbar.

Kreuzen Sie an: ☐ Ich zeige die Aussage. ☐ Ich widerlege die Aussage.

(b) **Aussage:** Jeder DEA, der den regulären Ausdruck $a^* \cup b^*$ erkennt, hat mindestens 3 Zustände.

Kreuzen Sie an: ☐ Ich zeige die Aussage. ☐ Ich widerlege die Aussage.

(c) Sei $\Sigma = \{a, b\}$. Für $k \in \mathbb{N}$ bezeichnen wir mit DEA^k die Klasse aller Sprachen $L \subseteq \Sigma^*$, für die es einen DEA ohne impliziten Müllzustand[1] gibt, der L akzeptiert und höchstens k Zustände hat.

Aussage: Für alle $k \in \mathbb{N}$ gilt: Die Klasse DEA^k ist unter Vereinigung abgeschlossen.

Tipp: Nutzen Sie Aussage (b).

Kreuzen Sie an: ☐ Ich zeige die Aussage. ☐ Ich widerlege die Aussage.

(d) **Aussage:** Die Sprache $\{(w_1, w_2) : L(T_{w_1}) \subseteq L(T_{w_2})\}$ ist nicht semi-entscheidbar.

Kreuzen Sie an: ☐ Ich zeige die Aussage. ☐ Ich widerlege die Aussage.

Möglicher Lösungsansatz: Benutzen Sie, dass die Sprache $\{(w_1, w_2) : L(T_{w_1}) = L(T_{w_2})\}$ *aus der Übung nicht semi-entscheidbar ist.*

[1]Die Übergangsfunktion $\delta : Q \times \Sigma \to Q$ muss total sein, d.h. für jedes Paar $(q, a) \in Q \times \Sigma$ ist $\delta(q, a)$ definiert.

Problem 5: NP-Vollständigkeit $2 + 2 + 7 = 11$ Punkte

Gegeben sei das folgende Problem:

KLEINER MENGENSCHNITT
Gegeben: Eine Grundmenge M, Teilmengen $L_1, \ldots, L_m \subseteq M$, sowie Zahlen $k, \ell \leqslant m$.
Problem: Gibt es eine Menge an Indizes $I \subseteq \{1, \ldots, m\}$ mit $|I| = k$, sodass

$$\left| \bigcap_{i \in I} L_i \right| \leqslant \ell?$$

(a) Gegeben sei folgende Instanz von KLEINER MENGENSCHNITT:

Grundmenge $M = \{1, 2, 3, 4, 5\}$ und Teilmengen

$$L_1 = \{1, 2, 3, 4\}, \quad L_2 = \{1, 2, 3\}, \quad L_3 = \{2, 4\}, \quad L_4 = \{2, 5\}, \quad L_5 = \{2, 3, 5\}$$

sowie $k = 3$ und $\ell = 1$. Zeigen Sie, dass es sich um eine Ja-Instanz handelt. Geben Sie, wenn möglich, einen Wert für ℓ an, sodass es sich um eine Nein-Instanz handeln würde.

(b) Zeigen Sie, dass KLEINER MENGENSCHNITT $\in \mathsf{NP}$.

(c) Zeigen Sie, dass KLEINER MENGENSCHNITT NP-schwer ist. Reduzieren Sie hierzu von einem geeigneten Problem im Anhang der Klausur auf KLEINER MENGENSCHNITT.

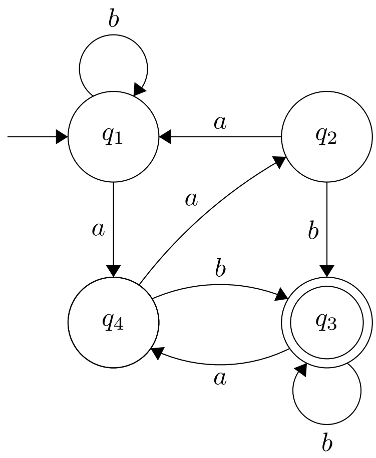
Hinweis: Es gilt $\left(\bigcup_{i\in I} L_i\right) = M - \bigcap_{i\in I} L_i^c$, *wobei* $-$ *die Mengendifferenz ist.*

Problem 6: Synchronisierende DEAs 1 + 5 + 5 = 11 Punkte

Sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DEA. Wir nennen ein Wort $w \in \Sigma^*$ *synchronisierend* , falls man in jedem Zustand beim Einlesen des Wortes w in demselben Zustand q landet, d.h. es existiert ein Zustand $q \in Q$, sodass für alle Zustände $p \in Q$ gilt, dass $\delta^*(p, w) = q$.

Wir nennen einen DEA synchronisierend, falls für diesen DEA ein synchronisierendes Wort existiert. [2]

(a) Zeigen Sie, dass der folgende DEA synchronisierend ist, indem Sie ein synchronisierendes Wort w angeben.



(b) Beweisen Sie folgende Äquivalenz: Ein DEA A ist synchronisierend genau dann, wenn für jedes Zustandspaar $p_1, p_2 \in Q$ ein Wort $w \in \Sigma^*$ existiert, sodass

$$\delta^*(p_1, w) = \delta^*(p_2, w).$$

[2] Folgendes Problem ist ein offenes Problem in der Automatentheorie: Hat jeder synchronisierender DEA mit n Zuständen ein synchronisierendes Wort der Länge höchstens $(n-1)^2$?

Achtung: Achten Sie auf die Quantorenreihenfolge!

(c) Geben Sie einen Polynomialzeitalgorithmus an, der für einen gegebenen DEA A entscheidet, ob dieser synchronisierend ist.

Hinweis: Sie dürfen das Resultat aus der Teilaufgabe b) benutzen. Es kann hilfreich sein, einen Produktautomaten zu konstruieren.

Die folgenden Probleme können Sie als NP-vollständig annehmen und für Reduktionen verwenden.

Subset Sum
Gegeben: Zahlen $a_1 \dots a_n \in \mathbb{N}$, und Wert $t \in \mathbb{N}$.
Problem: Gibt es $I \subseteq \{1, \dots, n\}$, sodass $\sum_{i \in I} a_i = t$?

Clique
Gegeben: Ein ungerichteter Graph $G = (V, E)$ und eine Zahl $k < |V|$.
Problem: Gibt es eine Clique der Größe mindestens k?

Hamilton-Kreis
Gegeben: Ein ungerichteter Graph $G = (V, E)$.
Problem: Gibt es einen Kreis in G, der jeden Knoten genau einmal besucht?

3-COLOR
Gegeben: Ein ungerichteter Graph $G = (V, E)$.
Problem: Gibt es eine Funktion $c : V \to \{1, 2, 3\}$, sodass für jede Kante $\{u, v\}$ gilt $c(u) \neq c(v)$?

Partition
Gegeben: Zahlen $a_1 \dots a_n \in \mathbb{N}$.
Problem: Gibt es $I \subseteq \{1, \dots, n\}$, sodass $\sum_{i \in I} a_i = \sum_{j \notin I} a_j$?

3-Sat
Gegeben: Menge U an aussagenlogischen Variablen und eine Menge C von Klauseln mit genau drei Literalen.
Problem: Gibt es eine Wahrheitsbelegung der Variablen aus U, sodass alle Klauseln aus C erfüllt sind?

Sat
Gegeben: Menge U an aussagenlogischen Variablen und eine Menge C von Klauseln.
Problem: Gibt es eine Wahrheitsbelegung der Variablen aus U, sodass alle Klauseln aus C erfüllt sind?

Set Cover
Gegeben: Eine Grundmenge M, Teilmengen $U_1, \dots, U_m \subseteq M$ mit $\bigcup_{i=1}^{m} U_i = M$, sowie eine Zahl c.
Problem: Gibt es eine Indexmenge $I \subseteq \{1, \dots, m\}$ mit $|I| \leqslant c$ und $\bigcup_{i \in I} U_i = M$?

Dominating Set
Gegeben: Ein Graph $G = (V, E)$ und eine Zahl k.
Gesucht: Eine Teilmenge $D \subseteq V$, sodass $|D| \leqslant k$ und für alle Knoten $v \in V$ gilt $v \in D$ oder v hat einen Nachbarn $u \in D$, also $\{u, v\} \in E$.