1. Klausur zur Vorlesung
Theoretische Grundlagen der Informatik
Wintersemester 2023/2024

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, falls Sie den gesamten Platz aufgebraucht haben.
- Die Tackernadel darf nicht gelöst werden.
- Als Hilfsmittel ist ein beschriebenes 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	2	1	3	2	–	8					–	
Aufg. 2	2	2	3	1	3	11						
Aufg. 3	2	4	1	2	–	9					–	
Aufg. 4	3	8	–	–	–	11			–	–	–	
Aufg. 5	3	3	6	–	–	12				–	–	
Aufg. 6	1	1	3	4	–	9					–	
Σ						60						

Problem 1: Automaten 2+1+3+2=8 Punkte

Sei $L \subseteq \Sigma^*$ eine Sprache. Wir haben in der Vorlesung für ein Wort $w \in \Sigma^*$ die Menge der gültigen Suffixe $S_w(L)$ bezüglich L definiert:

$$S_w(L) \coloneqq \{x \in \Sigma^* \mid wx \in L\}.$$

(a) Für ein Wort $w \in \{0,1\}^*$ und $k \in \mathbb{N}^+$ bezeichnen wir mit w_k das k-te Zeichen von w. Sei

$$L_3 \coloneqq \left\{ w \in \{0,1\}^* \mid w_{3i} = 1 \text{ für alle } i \in \mathbb{N}^+ \text{ mit } 1 \leqslant i \leqslant \frac{|w|}{3} \right\}$$

die Sprache in $\{0,1\}^*$, für die jedes dritte Zeichen eine 1 ist. Beispielsweise ist $01\mathbf{1}0 \in L_3$, aber $01\mathbf{0}0 \notin L_3$.

Geben Sie einen DEA an, der für das Wort 01 die Menge der gültigen Suffixe $S_{01}(L_3)$ erkennt.

(b) Welche Wörter enthält $S_\varepsilon(L)$ für eine beliebige Sprache L?

(c) Beweisen Sie die folgende Aussage. Eine Sprache L ist genau dann regulär, wenn $S_w(L)$ für alle $w \in \Sigma^*$ regulär ist.

(d) Beweisen Sie, dass es eine nichtreguläre Sprache L gibt, bei der $S_w(L)$ für unendlich viele $w \in \Sigma^*$ regulär ist. Sie müssen nicht zeigen, dass L nicht regulär ist.

Problem 2: 1-Zustands DPDA $2 + 2 + 3 + 1 + 3 = 11$ Punkte

Ein 1-Zustands-DPDA ist ein deterministischer Kellerautomat, der nur einen Zustand hat. In dieser Aufgabe betrachten wir nur Kellerautomaten, die mit leerem Stack akzeptieren.

Hinweis: Beim Anwenden der Übergangsfunktion eines DPDA wird zuerst das oberste Stacksymbol eingelesen. Wenn der Stack leer ist und noch ein Zeichen eingelesen werden soll, kann der DPDA somit nicht weiterarbeiten und nicht mehr akzeptieren.

(a) Betrachten Sie den 1-Zustands-DPDA A mit einzigem Zustand q, Alphabet $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$, Stackalphabet $\Gamma = \{Z_0, Z_1, Z_2, X\}$, Stackinitialisierung Z_0 und folgender Übergangsfunktion:

$$\delta(q, \mathsf{a}, Z_0) = (q, Z_1)$$
$$\delta(q, \mathsf{b}, Z_1) = (q, Z_0)$$
$$\delta(q, \mathsf{c}, Z_0) = (q, \varepsilon).$$

In allen bisher nicht definierten Fällen wird das gelesene Stacksymbol durch ein X überschrieben.

Geben Sie einen DEA A' mit Zustandsmenge $Q = \{Z_0, Z_1, Z_2, X\}$ an, der die gleiche Sprache wie A erkennt. Hierbei ist X der explizite Müllzustand von A'.

(b) Zeigen Sie, dass jede Sprache, die von einem 1-Zustands-DPDA mit leerem Stack erkannt wird, präfixfrei ist.

Hinweis: Wir nennen eine Sprache L über einem Alphabet Σ präfixfrei, wenn es für kein $w \in L$ ein $z \in \Sigma^+$ gibt, sodass $wz \in L$.

(c) Geben Sie eine nichtreguläre Sprache L an, die von einem 1-Zustands-DPDA mit leerem Stack erkannt werden kann. Beschreiben Sie einen solchen 1-Zustands-DPDA. Sie müssen nicht zeigen, dass L nicht regulär ist.

(d) Zeigen Sie, dass es reguläre Sprachen gibt, die nicht von einem 1-Zustands-DPDA mit leerem Stack erkannt werden können.

Wir betrachten nun 1-Zustands-NPDAs. Ein 1-Zustands-NPDA ist ein nichtdeterministischer Kellerautomat mit nur einem Zustand.

(e) Zeigen Sie, dass es für jeden gegebenen NEA $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ einen 1-Zustands-NPDA gibt, der die gleiche Sprache wie A erkennt.

Problem 3: Grammatiken 2+4+1+2 = 9 Punkte

Wir betrachten das Alphabet $\Sigma \coloneqq \{\mathsf{a}, \mathsf{b}\}$. Für ein Wort $w \in \Sigma^*$ und ein Zeichen $x \in \Sigma$ bezeichnen wir mit $|w|_x$ die Anzahl an Vorkommen des Zeichens x im Wort w. Sei

$$L \coloneqq \{w \in \Sigma^* \mid |w|_{\mathsf{a}} = |w|_{\mathsf{b}}\},$$

die Sprache, die alle Wörter enthält, die genauso häufig das Zeichen b wie das Zeichen a enthalten.

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

Betrachten Sie die kontextfreie Grammatik $G = (\Sigma, \{S, A, B\}, S, R)$ mit Ableitungsregeln

$$\begin{aligned} S &\to \varepsilon \mid \mathsf{a}B \mid \mathsf{b}A \\ A &\to S\mathsf{a}S \\ B &\to S\mathsf{b}S. \end{aligned}$$

Wir ordnen Wörtern Gewichte zu. Dazu definieren wir für $w \in \Sigma^*$ das Gewicht $c(w)$ als

$$c(w) = |w|_{\mathsf{a}} - |w|_{\mathsf{b}}.$$

Ferner definieren wir für ein Nichtterminal X der Grammatik G die Grammatik $G_X = (\Sigma, \{S, A, B\}, X, R)$. Die Sprache, die von G_X erzeugt wird, besteht also aus allen Wörtern, die von X abgeleitet werden können.

(b) Sie dürfen ohne Beweis annehmen, dass die folgenden beiden Gleichungen gelten.

$$L(G_A) = \{w \in \Sigma^* \mid c(w) = 1\} \tag{1}$$
$$L(G_B) = \{w \in \Sigma^* \mid c(w) = -1\} \tag{2}$$

Zeigen Sie die folgende Gleichung.

$$L(G_S) = \{w \in \Sigma^* \mid c(w) = 0\}.$$

Hinweis: Achten Sie darauf, beide Inklusionen zu zeigen.

(c) Ist L kontextfrei? Begründen Sie.

Wir nennen eine Turingmaschine *in-place*, wenn sie bei der Abarbeitung nur Zellen beschreibt, in denen zu Beginn die Eingabe stand.

(d) Beschreiben Sie eine nicht-deterministische in-place Turingmaschine, die die Sprache L entscheidet.

Problem 4: NP-Vollständigkeit 3 + 8 = 11 Punkte

GREEN-WASHING
Gegeben:

- Eine Menge U von Elementen
- eine Familie G von Teilmengen $G_1, \ldots, G_\ell \subseteq U$, welche wir als *grüne Mengen* bezeichnen
- eine Familie R von Teilmengen $R_1, \ldots, R_m \subseteq U$, welche wir als *Regenbogen-Mengen* bezeichnen
- und ein Parameter $k \in \mathbb{N}^+$, der die Anzahl der verfügbaren Farben angibt.

Problem: Können die Elemente von U mit k Farben so gefärbt werden, dass jede grüne Menge mindestens ein grünes Element enthält und jede Regenbogen-Menge keine zwei Elemente der gleichen Farbe enthält?

3SAT
Gegeben: Menge U an aussagenlogischen Variablen und eine Menge C an Klauseln mit je maximal drei Literalen.
Problem: Gibt es eine Belegung der Variablen in U, sodass in jeder Klausel mindestens ein Literal wahr ist?

(a) Geben Sie für GREEN-WASHING und 3SAT an, woraus eine Instanz bzw. Lösung des Problems im Allgemeinen besteht und geben sie jeweils ein Beispiel ohne leere Mengen an. Das Beispiel einer Lösung sollte die Lösung der zugehörigen Beispiel-Instanz sein, wählen Sie also eine Ja-Instanz. Füllen Sie dafür die Tabelle unten aus.

	3SAT	GREEN-WASHING
Instanz (allgemein)		
Instanz (Beispiel)		
Lösung (allgemein)		
Lösung (Beispiel)		

(b) Zeigen Sie, dass GREEN-WASHING NP-vollständig ist. Sie dürfen annehmen, dass 3SAT NP-schwer ist.

Hinweis: Nutzen Sie Regenbogen-Mengen der Größe 2, um zu erzwingen, dass ein Element grün und das andere nicht grün ist.

Problem 5: Approximation 3 + 3 + 6 = 12 Punkte

SQUARE-COVERING
Gegeben: Ein Quadrat Q mit Seitenlänge $\ell \in \mathbb{N}^+$, sowie eine Menge $M = \{Q_1, \ldots, Q_n\}$ weiterer Quadrate mit Seitenlängen $\ell_i \in \mathbb{N}^+$.
Frage: Wieviel Fläche von Q kann maximal durch Quadrate in M überdeckt werden? Hierbei dürfen sich die Q_i überlappen und auch aus Q herausragen.

Hierbei gelten folgende Regeln für alle SQUARE-COVERING Instanzen:

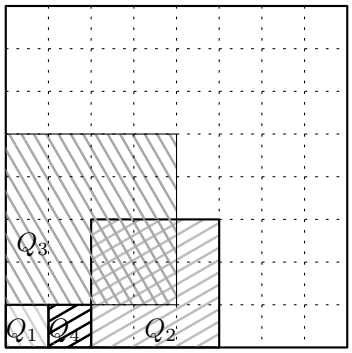
- Q sowie alle Quadrate in M sind achsenparallel.
- Q hat seine linke untere Ecke an Position $(0, 0)$.
- Die Quadrate aus M dürfen beliebig verschoben (aber nicht gedreht) werden.

Der folgende Algorithmus $\mathcal{A}$ platziert die Quadrate in M zeilenweise in aufsteigender Größe. Hierbei wird das erste Quadrat einer Zeile immer direkt oberhalb des ersten Quadrats der vorherigen Zeile platziert und alle Quadrate einer Zeile werden auf der gleichen Höhe platziert. Eine neue Zeile wird begonnen, wenn die vorherige Zeile mindestens bis zur Hälfte von Q gefüllt ist:

Algorithmus $\mathcal{A}$: SQUARE-COVERING APPROX

```
1 sortiere M aufsteigend; ab jetzt: ℓ_i ⩽ ℓ_{i+1} für alle i ∈ {1, ..., n − 1}
2 x ← ∞, y ← 0, h ← 0
3 Für i ← 1 bis n
4 |   Wenn x ⩾ ⌈ℓ/2⌉                          (⌈·⌉ bedeutet Aufrunden)
5 |   |   x ← 0, y ← y + h, h ← ℓ_i
6 |   platziere die linke untere Ecke von Q_i an Position (x, y)
7 |   x ← x + ℓ_i
8 return durch M überdeckte Fläche von Q
```

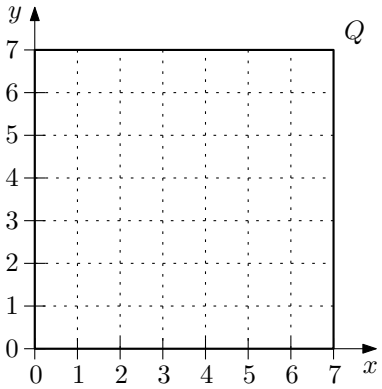
Beispiel: Auf der Eingabe $M = \{Q_1, Q_2, Q_3, Q_4\}$ mit $\ell_1 = 1$, $\ell_2 = 3$, $\ell_3 = 4$, $\ell_4 = 1$ und $\ell = 8$ platziert $\mathcal{A}$ die Quadrate aus M wie folgt und überdeckt dabei eine Fläche von 23 in Q:



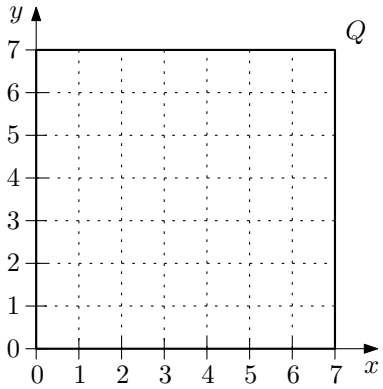
(a) Führen Sie $\mathcal{A}$ auf der Instanz I mit $M = \{Q_1, Q_2, Q_3, Q_4\}$ aus, wobei $\ell_1 = 1$, $\ell_2 = 3$, $\ell_3 = 3$, $\ell_4 = 4$ und $\ell = 7$ aus. Tragen Sie das Ergebnis in eine Kopie des auf der nächsten Seite gegebenen Quadrats Q ein.

Tragen Sie eine optimale Überdeckung von Q in eine weitere Kopie auf der nächsten Seite ein.

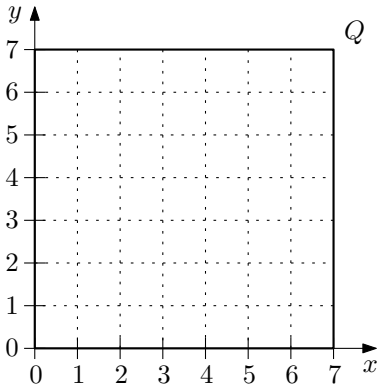
Geben Sie $\mathcal{A}(I)$ und $\mathrm{OPT}(I)$ an.



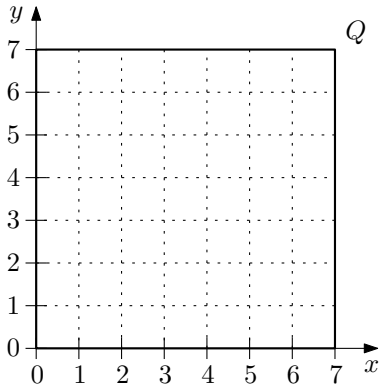
Tragen Sie hier die von $\mathcal{A}$ berechnete Lösung ein.



Kopie, falls Sie Ihre Antwort korrigieren wollen. Sollten Sie beide Kopien von Q benutzen, markieren Sie deutlich, welche bewertet werden soll!



Tragen Sie hier eine optimale Lösung ein.



Kopie, falls Sie Ihre Antwort korrigieren wollen. Sollten Sie beide Kopien von Q benutzen, markieren Sie deutlich, welche bewertet werden soll!

(b) Zeigen Sie, dass in einer von $\mathcal{A}$ berechneten Lösung jeder Punkt in Q von höchstens zwei Quadraten überdeckt wird.

(c) Zeigen Sie, dass $\mathcal{A}$ ein polynomieller Approximationsalgorithmus mit relativer Gütegarantie 4 für SQUARE-COVERING ist. (Sie dürfen ohne Beweis annehmen, dass die Return-Anweisung in Zeile 8 nur polynomielle Laufzeit benötigt.)

Hinweis: Unterscheiden Sie, ob $\mathcal{A}$ Quadrate so platziert, dass sie aus Q herausragen oder nicht und zeigen Sie für jeden Fall die Approximationsgüte.

Problem 6: Entscheidbarkeit 1+1+3+4 = 9 Punkte

In dieser Aufgabe betrachten wir nur Gödelnummern von deterministischen Turingmaschinen mit Eingabealphabet $\Sigma \coloneqq \{0, 1\}$.

Sei $L \subseteq \Sigma^*$ und $w \in \Sigma^*$ ein Wort. Wir sagen L ist *w-präfixfrei*, wenn w von keinem Wort aus L ein Präfix ist. Das heißt, hängen wir ein beliebiges $z \in \Sigma^*$ an w an, so gilt $wz \notin L$.

(a) Geben Sie eine Sprache L' an, die 0-präfixfrei ist.

Betrachten Sie die Sprache

$$S \coloneqq \{\langle M\rangle \# w \mid L(M) \text{ ist } w\text{-präfixfrei}\}.$$

(b) Sei $M_\emptyset$ die Turingmaschine, die jedes Wort ablehnt. Zeigen Sie, dass $\langle M_\emptyset\rangle \# w \in S$ für alle Wörter $w \in \Sigma^*$.

Aus der Vorlesung wissen Sie, dass die universelle Sprache

$$L_u \coloneqq \{\langle M\rangle \# w \mid w \in L(M)\}$$

unentscheidbar ist.

(c) Betrachten Sie ein Tupel $\langle M\rangle \# w$ bestehend aus der Gödelnummer einer Turingmaschine M und einem Wort $w \in \Sigma^*$. Konstruieren Sie eine deterministische Turingmaschine $T_{M,w}$, sodass

$$w \in L(M) \iff L(T_{M,w}) \text{ ist nicht } w\text{-präfixfrei.}$$

Begründen Sie, warum die von Ihnen angegebene Turingmaschine $T_{M,w}$ die obige Eigenschaft hat. Ihre Konstruktion muss aus w und $\langle M\rangle$ berechenbar sein.

(d) Zeigen Sie, dass S unentscheidbar ist, indem Sie von der universellen Sprache reduzieren. Sie dürfen in dieser Aufgabe nicht den Satz von Rice verwenden.

Hinweis: Verwenden Sie Aufgabe (c).