

**2. Klausur zur Vorlesung
Theoretische Grundlagen der Informatik
Wintersemester 2020/2021**

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

Beachten Sie:

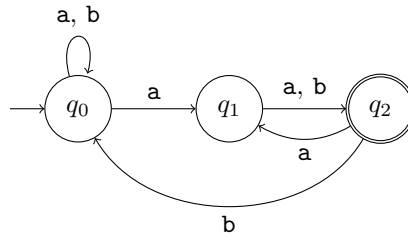
- Bringen Sie den Aufkleber mit Ihrem Namen und Ihrer Matrikelnummer auf diesem Deckblatt an und beschriften Sie jedes Aufgabenblatt mit Ihrem Namen und Matrikelnummer.
- Schreiben Sie die Lösungen auf die Aufgabenblätter und Rückseiten. Am Ende der Klausur sind zusätzliche Leerseiten. Fordern Sie zusätzliches Papier bitte nur an, falls Sie den gesamten Platz aufgebraucht haben.
- Es sind keine Hilfsmittel zugelassen.

	Mögliche Punkte						Erreichte Punkte					
	a	b	c	d	e	Σ	a	b	c	d	e	Σ
Aufg. 1	2	4	2	1	–	9					–	
Aufg. 2	7					7						
Aufg. 3	4	3	4	–	–	11				–	–	
Aufg. 4	2	7	–	–	–	9			–	–	–	
Aufg. 5	2	1	2	3	3	11						
Aufg. 6	4	4	–	–	–	8			–	–	–	
Aufg. 7	5					5						
Σ						60						

Problem 1: Endliche Automaten

2 + 4 + 2 + 1 = 9 Punkte

Gegeben ist folgender NEA:



(a) Geben Sie für die Sprache, die von diesem Automaten erkannt wird, einen regulären Ausdruck an.

(b) Geben Sie einen äquivalenten DEA an.

- (c) Sei Σ ein beliebiges Alphabet mit 2 Zeichen. Beschreiben Sie ein *deterministisches* Verfahren, das als Eingabe einen NEA \mathcal{A} mit n Zuständen und ein Wort w der Länge m bekommt und entscheidet, ob w von \mathcal{A} akzeptiert wird. Ist die Laufzeit des Verfahrens polynomiell in der Eingabegröße?

- (d) Das Wortproblem für reguläre Sprachen kann in Linearzeit gelöst werden. Warum widerspricht das nicht Ihrer Lösung für Aufgabenteil (c)?

Problem 2: Pumping-Lemma

7 Punkte

Wir betrachten die folgenden zwei Sprachen.

$$L_1 = \{w \in \{0, 1\}^* : |w|_1 - |w|_0 = 1\}$$
$$L_2 = \{w \in \{0, 1\}^* : |w|_1 - |w|_0 \equiv 1 \pmod{2}\},$$

wobei $|w|_i$ für $i = 0, 1$ die Anzahl der Vorkommen von i in w bezeichnet.

Ist L_1 bzw. L_2 regulär? Zeigen bzw. widerlegen Sie jeweils.

Problem 3: Grammatiken

4 + 3 + 4 = 11 Punkte

Wir definieren die Sprachen L über dem Alphabet $\Sigma = \{0, 1\}$ und L' über dem Alphabet $\Sigma' = \{a, b\}$ durch

$$L = \{(01)^n(10)^n \mid n > 0\} \text{ und} \\ L' = \{a^n b \mid n > 0\} \cup \{b a^n \mid n > 0\}.$$

- (a) Geben Sie eine Grammatik für L und eine Grammatik für L' an.

Nun betrachten wir ein Wort $w \in L$ und ersetzen jede 0 in w durch ein beliebiges Wort aus L' .

Beispiel: $w = 01011010 \rightsquigarrow ab1ba11ab1ab, baa1ab11aaab1ba, \dots$

Die Sprache aller Wörter, die so entstehen können, bezeichnen wir mit $\alpha_w(L')$. Außerdem definieren wir

$$\alpha_L(L') = \bigcup_{w \in L} \alpha_w(L').$$

Das heißt, $\alpha_L(L')$ enthält alle Wörter, die entstehen, wenn wir mit einem beliebigen Wort aus L beginnen und jede 0 durch ein beliebiges Wort aus L' ersetzen.

- (b) Geben Sie eine Grammatik für $\alpha_L(L')$ an.

- (c) Seien nun L und L' beliebige kontextfreie Sprachen über $\Sigma = \{0, 1\}$ bzw. $\Sigma' = \{a, b\}$. Zeigen Sie, dass $\alpha_L(L')$ ebenfalls kontextfrei ist.

Problem 4: NP-Vollständigkeit

2 + 7 = 9 Punkte

In der Vorlesung wurde das NP-vollständige Entscheidungsproblem 3-SAT vorgestellt:

Gegeben: Menge U von Variablen
Menge C von Klauseln über U , wobei jede Klausel höchstens drei Literale enthält

Frage: Existiert eine Belegung der Variablen mit **wahr** und **falsch**, sodass in jeder Klausel mindestens ein Literal **wahr** ist?

Anmerkung: In der Vorlesung wurde gefordert, dass jede Klausel genau drei Literale enthält. Hier wird nur gefordert, dass jede Klausel höchstens drei Literale enthält. Auch in dieser Variante ist 3-SAT NP-vollständig.

Wir betrachten nun das leicht abgewandelte Entscheidungsproblem MONOTONE NOT ALLEQUAL (MNAE):

Gegeben: Menge U von Variablen
Menge C von Klauseln über U , wobei jede Klausel höchstens drei Literale enthält und kein Literal negiert ist

Frage: Existiert eine Belegung der Variablen mit **wahr** und **falsch**, sodass in jeder Klausel mindestens ein Literal **wahr** ist und mindestens ein Literal **falsch** ist?

Sie dürfen voraussetzen, dass MONOTONE NOT ALLEQUAL NP-vollständig ist.

Außerdem betrachten wir das folgende MENGENFÄRBUNGSPROBLEM (MFP):

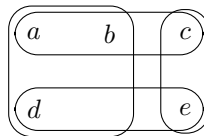
Gegeben: Eine Menge X
Eine Menge \mathcal{F} von höchstens dreielementigen Teilmengen von X

Frage: Existiert eine Färbung der Elemente aus X in rot und blau, sodass keine Menge aus \mathcal{F} einfarbig ist? (Eine Färbung weist jedem Element aus X genau eine der Farben rot oder blau zu.)

(a) Zeigen Sie für die folgenden beiden Instanzen, dass es sich um Ja-Instanzen handelt. Tragen Sie hierzu die Farben bzw. Wahrheitswerte in die Tabelle unten ein.

- MENGENFÄRBUNGSPROBLEM

$$X = \{a, b, c, d, e\}, \mathcal{F} = \{\{a, b, c\}, \{a, b, d\}, \{c, e\}, \{d, e\}\}$$



- MONOTONE NOT ALLEQUAL

$$U = \{a, b, c, d, e\}, C = \{(a, b, c), (a, b, d), (c, e), (d, e)\}$$

	a	b	c	d	e
MENGENFÄRBUNGSPROBLEM					
MONOTONE NOT ALLEQUAL					

(b) Zeigen Sie, dass das MENGENFÄRBUNGSPROBLEM NP-vollständig ist.

Problem 5: Pizza

2 + 1 + 2 + 3 + 3 = 11 Punkte

Der ebenso gewissenhafte wie geschäftstüchtige Doktor Meta hat beschlossen, dem Verbrechen den Rücken zu kehren. Stattdessen hat er einen Pizzaliefersdienst gegründet. Er hat sich eine unendlich große Armee von Robotern gebaut, die beliebig viele Pizzen gleichzeitig ausliefern können. Leider hat er aber vergessen, Köche einzustellen. Also muss er alle Pizzen selber zubereiten.

Sei S die Menge der Bestellungen, die Doktor Meta innerhalb eines Tages bekommt. Jede Bestellung j trifft zu einem Bestellzeitpunkt B_j ein und hat eine Zubereitungszeit Z_j sowie eine Lieferzeit L_j . Sobald er eine Bestellung zu einem Zeitpunkt $F_j \geq B_j + Z_j$ fertig zubereitet hat, kann sie dank der Roboterarmee direkt ausgeliefert werden und kommt zum Zeitpunkt $A_j = F_j + L_j$ beim Kunden an. Weil Doktor Meta möglichst früh Feierabend machen möchte, sucht er eine Zubereitungsreihenfolge, die $A_{\max} = \max_{j \in S} A_j$ minimiert, also die Ankunftszeit der letzten Bestellung.

Da Doktor Meta kein gelernter Koch ist, kann er immer nur eine Bestellung gleichzeitig zubereiten. Außerdem ist er nicht in der Lage, die Zubereitung einer Bestellung zu unterbrechen. Wenn er eine Bestellung fertig zubereitet hat, wählt er eine beliebige neue Bestellung aus, die schon eingetroffen ist, aber noch nicht abgearbeitet wurde, und bereitet sie zu. Sollte es gerade keine offenen Bestellungen geben, ruht er sich aus und schmiedet Weltherrschaftspläne, bis die nächste Bestellung eintrifft.

Hier noch einmal eine Übersicht der verwendeten Variablen:

Gegeben: Menge S von Bestellungen. Für jede Bestellung $j \in S$:

- Bestellzeitpunkt B_j
- Zubereitungszeit Z_j
- Lieferzeit L_j

Ergebnis: Für jede Bestellung $j \in S$:

- Fertigstellungszeitpunkt $F_j \geq B_j + Z_j$
- Auslieferungszeitpunkt $A_j = F_j + L_j$

Ziel: Minimiere $A_{\max} = \max_{j \in S} A_j$

Doktor Meta ist mit der Situation unzufrieden. Er vermutet, dass er sein Vorgehen verbessern und früher Feierabend machen könnte, wenn er die Zubereitung einer Bestellung unterbrechen könnte.

(a) Zeigen Sie, dass Doktor Metas Vermutung stimmt. Gegeben seien zwei Bestellungen:

- Bestellung 1: Bestellzeitpunkt $B_0 = 0$, Zubereitungszeit $Z_0 = 5$, Lieferzeit $L_0 = 1$
- Bestellung 2: Bestellzeitpunkt $B_1 = 1$, Zubereitungszeit $Z_1 = 1$, Lieferzeit $L_1 = 5$

Markieren Sie in den unten stehenden Tabellen die Zeitpunkte, während denen die Bestellungen zubereitet bzw. geliefert werden.

Doktor Metas Lösung (ohne Unterbrechen):

Zubereitung 1:														
Lieferung 1:														
Zubereitung 2:														
Lieferung 2:														
	0	1	2	3	4	5	6	7	8	9	10	11	12	

Optimale Lösung (mit Unterbrechen):

Zubereitung 1:														
Lieferung 1:														
Zubereitung 2:														
Lieferung 2:														
	0	1	2	3	4	5	6	7	8	9	10	11	12	

Geben Sie für beide Lösungen den Wert von A_{\max} an.

Gegeben: Menge S von Bestellungen. Für jede Bestellung $j \in S$:

- Bestellzeitpunkt B_j
- Zubereitungszeit Z_j
- Lieferzeit L_j

Ergebnis: Für jede Bestellung $j \in S$:

- Fertigstellungszeitpunkt $F_j \geq B_j + Z_j$
- Auslieferungszeitpunkt $A_j = F_j + L_j$

Ziel: Minimiere $A_{\max} = \max_{j \in S} A_j$

Wir bezeichnen im Folgenden mit A_{\max}^* den spätesten Ankunftszeitpunkt in einer optimalen Lösung mit Unterbrechen und mit A_{\max} den spätesten Ankunftszeitpunkt in Doktor Metas Lösung ohne Unterbrechen.

(b) Zeigen Sie: $A_{\max}^* \geq \sum_{j \in S} Z_j$.

(c) Zeigen Sie, dass für jede Bestellung j gilt: $A_{\max}^* \geq B_j + Z_j + L_j$.

(d) Sei x die letzte Bestellung, die in Doktor Metas Lösung beim Kunden ankommt. Zeigen Sie:
 $F_x \leq B_x + \sum_{j \in S} Z_j$.

- (e) Zeigen Sie, dass das Vorgehen von Doktor Meta ein Approximationsalgorithmus mit relativer Gütegarantie 2 ist.

Problem 6: 2-Kopf-Turingmaschinen

4 + 4 = 8 Punkte

Wir erweitern das Modell einer Turingmaschine, indem wir einen weiteren Kopf hinzufügen. Die Köpfe haben eine gemeinsame Zustandsmenge Q und gemeinsame Eingabe- und Bandalphabete Σ und Γ . Beide Köpfe arbeiten auf dem selben Band, wobei abwechselnd je ein Schritt der Übergangsfunktionen für Kopf 1 und Kopf 2 ausgeführt wird. Auch wenn Kopf 2 an der Reihe ist, ist bekannt, was Kopf 1 gerade liest, und umgekehrt. Die Übergangsfunktion hat also die Form

$$\delta: Q \times \Gamma \times \Gamma \times \{1, 2\} \rightarrow Q \times \Gamma \times \{L, N, R\},$$

wobei der vierte Parameter angibt, welcher Kopf an der Reihe ist. Die Übergangsfunktion wird also abwechselnd mit 1 bzw. 2 als viertes Argument aufgerufen. Wir nennen eine solche Turingmaschine eine *2-Kopf-Turingmaschine*.

- (a) Sei $\Sigma = \{a, b, \#\}$. Beschreiben Sie eine 2-Kopf-Turingmaschine, die die Sprache $\{w\#w\# \mid w \in \{a, b\}^*\}$ in Linearzeit erkennt. Begründen Sie kurz die Laufzeit.

- (b) Zeigen Sie, dass 2-Kopf-Turingmaschinen und klassische Turingmaschinen (d.h. 1 Kopf, 1 Band, wie in Vorlesung definiert) gleich mächtig sind. Simulieren Sie hierzu eine klassische Turingmaschine mit einer 2-Kopf-Turingmaschine und umgekehrt.

Problem 7: Semi-Entscheidbarkeit

5 Punkte

Zeigen Sie, dass die Menge der semi-entscheidbaren Sprachen unter Konkatenation abgeschlossen ist. D.h. zeigen Sie für zwei semi-entscheidbare Sprachen L und L' , dass die Sprache $L \cdot L' = \{ww' \mid w \in L, w' \in L'\}$ ebenfalls semi-entscheidbar ist.

