1. Klausur zur Vorlesung
Theoretische Grundlagen der Informatik
Wintersemester 2019/2020

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.

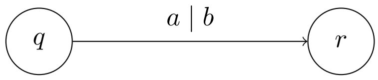
	Mögliche Punkte					Erreichte Punkte				
	a	b	c	d	Σ	a	b	c	d	Σ
Aufg. 1	2	2	3	3	10					
Aufg. 2	3	3	–	–	6			–	–	
Aufg. 3	4	2	3	–	9				–	
Aufg. 4	3	4	–	–	7			–	–	
Aufg. 5	1	2	3	2	8					
Aufg. 6	2	4	4	–	10				–	
Aufg. 7	2	3	2	3	10					
Σ					60					

Problem 1: Endliche Automaten mit Ausgabe $\quad 2 + 2 + 3 + 3 = 10$ Punkte

In dieser Aufgabe betrachten wir deterministische endliche Automaten, die bei jedem Zustandsübergang ein Zeichen ausgeben. Der Einfachheit halber gehen wir davon aus, dass Eingabe und Ausgabe dasselbe Alphabet Σ benutzen. Es gibt zwei Möglichkeiten, die Ausgabe zu definieren:

Bei *Mealy-Automaten* hat jeder Zustandsübergang seine eigene Ausgabe. Ein Mealy-Automat ist also ein 5-Tupel $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ mit Übergangsfunktion $\delta\colon Q \times \Sigma \to Q \times \Sigma$.

Beispiel:

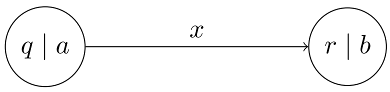


Beim Übergang von q zu r mit gelesenem Zeichen a wird das Zeichen b ausgegeben.

Bei *Moore-Automaten* hängt die Ausgabe nur vom erreichten Zustand ab. Formal ist ein Moore-Automat ein 6-Tupel $\mathcal{A} = (Q, \Sigma, \delta, \lambda, s, F)$ mit Übergangsfunktion $\delta\colon Q \times \Sigma \to Q$ wie üblich und Ausgabefunktion $\lambda\colon Q \to \Sigma$.

Achtung: Auch ein Moore-Automat gibt *nur bei einem Zustandsübergang* ein Zeichen aus. Der Startzustand s erzeugt zu Beginn der Abarbeitung noch keine Ausgabe. Wird s aber später durch einen Zustandsübergang wieder erreicht, wird $\lambda(s)$ ausgegeben.

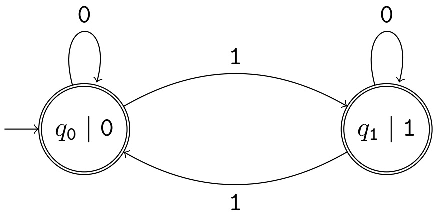
Beispiel:



Zustand q gibt beim Erreichen Zeichen a aus, während Zustand r Zeichen b ausgibt.

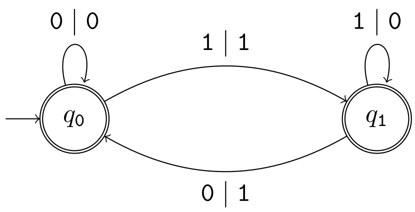
Wir nennen zwei Automaten *äquivalent*, wenn sie für jedes Eingabewort aus Σ^* dasselbe Akzeptanzverhalten haben und dieselbe Ausgabe produzieren.

(a) Geben Sie einen äquivalenten Mealy-Automat zu folgendem Moore-Automat an:



(b) Beschreiben Sie, wie für jeden Moore-Automat ein äquivalenter Mealy-Automat konstruiert werden kann. Begründen Sie, warum Ihre Konstruktion korrekt ist!

(c) Geben Sie einen äquivalenten Moore-Automat zu folgendem Mealy-Automat an:



(d) Beschreiben Sie, wie für jeden Mealy-Automat ein äquivalenter Moore-Automat konstruiert werden kann. Begründen Sie, warum Ihre Konstruktion korrekt ist!

Problem 2: Reguläre Sprachen $3 + 3 = 6$ Punkte

Für ein endliches Alphabet Σ sei eine Sprache der Form

$$L = \{w_k \in \Sigma^* \mid k \in \mathbb{N}^+\}$$

gegeben, wobei für jedes $k \in \mathbb{N}^+$ gilt: $|w_{k+1}| - |w_k| \geq \sqrt[2020]{k}$. In der Wortfolge $w_1, w_2, \dots$ werden die Längenabstände zwischen zwei aufeinanderfolgenden Wörtern also immer größer.

(a) Zeigen Sie, dass für jedes $n \in \mathbb{N}^+$ ein $k \in \mathbb{N}^+$ existiert, sodass $|w_{k+1}| - |w_k| > n$ und $|w_k| > n$.

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

Problem 3: Kontextfreie Grammatiken $4 + 2 + 3 = 9$ Punkte

Gegeben sei die kontextfreie Grammatik $G = (\Sigma = \{\texttt{a}, \texttt{b}, \texttt{c}, \texttt{d}\}, V = \{A, B, C, D, E\}, A, R)$. Die Regelmenge R enthält folgende Regeln:

$$
\begin{aligned}
A &\to AB \mid EC \mid \texttt{a} \\
B &\to CA \\
C &\to CC \mid DC \mid \texttt{c} \\
D &\to \texttt{d} \\
E &\to CA \mid DA
\end{aligned}
$$

(a) Wenden Sie den CYK-Algorithmus auf das Wort `adcaca` an. Ist das Wort in $L(G)$ enthalten?

a	d	c	a	c	a

(b) Geben Sie einen Ableitungsbaum für das Wort `adcaca` in G an.

(c) Der Ableitungsbaum für ein Wort ist im Allgemeinen nicht eindeutig. Geben Sie ein Verfahren an, das mithilfe der vom CYK-Algorithmus erzeugten Tabelle *alle* möglichen Ableitungsbäume für das überprüfte Wort generiert.

Hinweis: Berechnen Sie rekursiv für jedes Symbol X in Zelle V_{ij} der Tabelle die Menge aller Ableitungsbäume, die mögliche Ableitungen von X auf das Teilwort w_{ij} repräsentieren.

Problem 4: Turing-Maschinen mit Lesezeichen 3 + 4 = 7 Punkte

Wir erweitern das Berechnungsmodell der Turing-Maschine (ein Kopf, ein Band, eine Spur) folgendermaßen. Zusätzlich zum Kopf verfügt die Turing-Maschine über zwei *Lesezeichen*. Anfangs befinden sich diese auf der Startposition des Kopfes. Neben den üblichen Operationen kann sich die Turing-Maschine nun in jedem Schritt entschließen, eins der Lesezeichen auf die aktuelle Kopfposition zu verschieben. Das Bandsymbol an dieser Position wird dadurch nicht gelöscht.

Außerdem ist es möglich, von der aktuellen Position zu einem beliebigen Lesezeichen zu springen. Sowohl das Verschieben eines Lesezeichens als auch das Springen zu einem Lesezeichen benötigen jeweils einen Zeitschritt. Durch das Springen zu einem Lesezeichen können also größere Distanzen in einem Zeitschritt überwunden werden. Wir bezeichnen dieses Modell als *Lesezeichen-Turing-Maschine* (LTM).

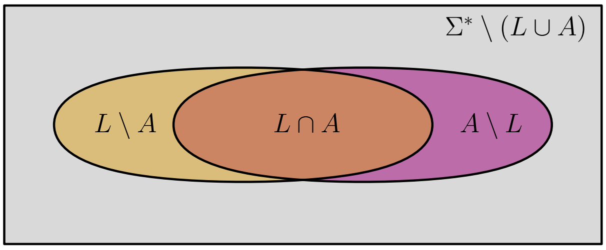
(a) Beschreiben Sie eine LTM, die die Palindromsprache in Linearzeit erkennt.

(b) Zeigen Sie, dass Lesezeichen-Turing-Maschinen und (herkömmliche) Turing-Maschinen gleich mächtig sind.

Problem 5: Entscheidbarkeit $1 + 2 + 3 + 2 = 8$ Punkte

In dieser Aufgabe sollen Sie eine Eigenschaft von entscheidbaren Sprachen beweisen, die sich informell so ausdrücken lässt: Um zu zeigen, dass eine Sprache entscheidbar ist, reicht es, eine Turing-Maschine zu finden, die für „fast alle“ (d.h. alle bis auf endlich viele) Wörter die richtige Antwort gibt.

Sei dazu eine beliebige Sprache L gegeben und eine Turing-Maschine $\mathcal{M}$, die eine Sprache A entscheidet. Die Situation lässt sich wie folgt darstellen:



(a) Zeigen Sie: Die Menge der entscheidbaren Sprachen ist unter Komplementbildung abgeschlossen.

(b) Zeigen Sie *eine* der folgenden Aussagen:

(1) Die Menge der entscheidbaren Sprachen ist unter Vereinigung abgeschlossen.

(2) Die Menge der entscheidbaren Sprachen ist unter Schnitt abgeschlossen.

Geben Sie explizit an, welche der beiden Aussagen Sie zeigen. Sie dürfen beide Aussagen im Folgenden benutzen.

(c) Zeigen Sie: Wenn $A \subseteq L$ gilt (also $A \setminus L$ leer ist) und $L \setminus A$ endlich ist, dann ist L entscheidbar.

(d) Zeigen Sie: Wenn $L \setminus A$ und $A \setminus L$ endlich sind, dann ist L entscheidbar.

Problem 6: NP-Vollständigkeit $2 + 4 + 4 = 10$ Punkte

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

Gegeben:	Menge U von n Variablen Menge C von k Klauseln über U Jede Klausel enthält *höchstens* drei Literale
Frage:	Existiert eine erfüllende Wahrheitsbelegung für C?

Achtung: In der Vorlesung wurde gefordert, dass jede Klausel *genau* drei Literate 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 3,3-SAT:

Gegeben:	Menge U von n Variablen Menge C von k Klauseln über U Jede Klausel enthält höchstens drei Literale Jede Variable kommt in höchstens drei Klauseln vor
Frage:	Existiert eine erfüllende Wahrheitsbelegung für C?

Wir sagen, dass eine Variable x in einer Klausel *vorkommt*, wenn die Klausel x oder $\bar{x}$ enthält. Beachten Sie, dass eine Variable mehrfach in einer Klausel vorkommen kann. Gezählt wird dann nur die Klausel als Ganzes, nicht die Anzahl der Vorkommen innerhalb der Klausel.

(a) Gegeben seien j Variablen $x_1, x_2, \ldots, x_j$. Geben Sie eine Menge C von j Klauseln mit jeweils höchstens drei Literalen an, sodass jede Variable in genau zwei Klauseln vorkommt und C genau zwei erfüllende Belegungen hat: $x_1 = x_2 = ... = x_j = \texttt{wahr}$ und $x_1 = x_2 = ... = x_j = \texttt{falsch}$.

(b) Gegeben sei eine 3-SAT-Instanz (U, C), in der genau eine Variable x in mehr als drei Klauseln vorkommt. Konstruieren Sie daraus eine 3,3-SAT-Instanz (U', C') und zeigen Sie, dass diese genau dann erfüllbar ist, wenn (U, C) erfüllbar ist.

(c) Zeigen Sie, dass 3,3-Sat NP-vollständig ist. Geben Sie bei Ihrer Reduktion explizit an, von welchem Problem auf welches reduziert wird!

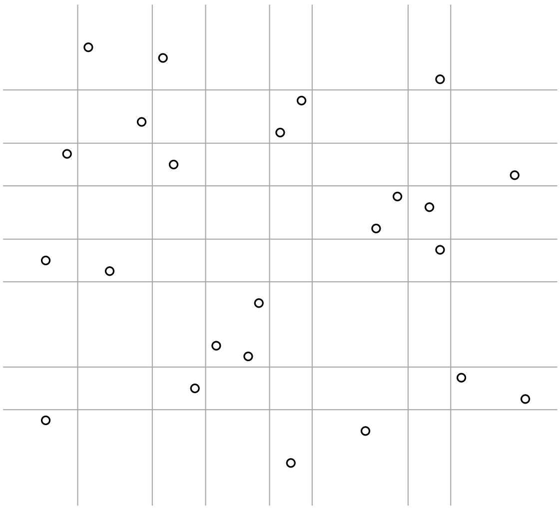
Problem 7: Approximation $2 + 3 + 2 + 3 = 10$ Punkte

Gegeben sei eine Menge $P \subset \mathbb{R}^2$ von n Punkten in der Ebene. Verschiedene Punkte in P haben stets verschiedene x-Koordinaten und verschiedene y-Koordinaten.

Das Optimierungsproblem MINIMAL-k-ENCLOSINGDISK (MkED) besteht darin, das kleinste $r \in \mathbb{R}$ zu bestimmen, sodass gilt: Es gibt einen Kreis mit Radius r, der mindestens k Punkte enthält.

Sei $m = \lceil 4n/k \rceil$. Platziere nun $m-1$ vertikale Geraden, die die Ebene so in m vertikale Streifen unterteilen, dass in jedem Streifen höchstens $k/4$ Punkte liegen. Platziere ebenso $m-1$ horizontale Geraden, die die Ebene in m horizontale Streifen unterteilen. Die Geraden sollen außerdem keinen Punkt schneiden.

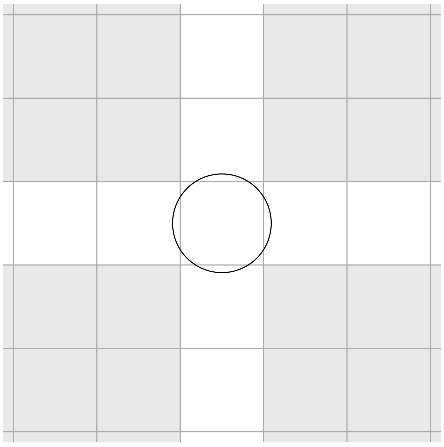
Hier ist ein Beispiel für $n = 24, k = 12$ und $m = 8$:



(a) Erklären Sie, wie die Lage der Geraden in Polynomialzeit berechnet werden kann.

Die Geraden ergeben ein Gitter, dessen Eckpunkte die Schnittpunkte der Geraden sind.

(b) Zeigen Sie, dass ein Kreis, der keinen Eckpunkt des Gitters enthält (siehe Abbildung), höchstens $k/2$ Punkte aus P enthalten kann.



Damit ist gezeigt, dass jeder Kreis, der mindestens k Punkte aus P enthält, auch einen Eckpunkt des Gitters enthält.

Unser Approximationsalgorithmus funktioniert folgendermaßen. Betrachte jeden Gitterpunkt g und berechne den Radius des kleinsten Kreises mit Mittelpunkt g, der mindestens k Punkte aus P enthält. Gib den kleinsten so berechneten Radius aus.

(c) Erklären Sie, wie für einen Gitterpunkt g der Radius des kleinsten Kreises mit Mittelpunkt g, der mindestens k Punkte aus P enthält, in Polynomialzeit berechnet werden kann.

Weil es $(m-1)^2 \in \mathcal{O}((n/k)^2)$ Gitterpunkte gibt, hat also auch der gesamte Approximationsalgorithmus polynomielle Laufzeit.

(d) Zeigen Sie, dass der obige Approximationsalgorithmus eine relative Gütegarantie von 2 hat. Beweisen Sie dazu, dass ein Kreis C mit minimalem Radius r stets vollständig in einem Kreis C' mit Radius $2r$ enthalten ist, wobei der Mittelpunkt von C' ein Eckpunkt des Gitters ist.

Hinweis: Dreiecksungleichung.