# 1. Klausur zur Vorlesung
# Theoretische Grundlagen der Informatik
# Wintersemester 2017/2018

**Hier Aufkleber mit Name und Matrikelnummer anbringen**

Vorname: ____________________

Nachname: ____________________

Matrikelnummer: ____________________

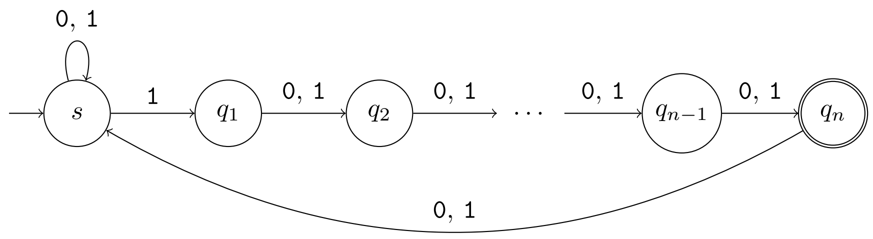
**Beachten Sie:**

- 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.
- Es sind keine Hilfsmittel zugelassen.
- You may write in English.

| | Mögliche Punkte | | | | Erreichte Punkte | | | |
|---|---|---|---|---|---|---|---|---|
| | a | b | c | $\Sigma$ | a | b | c | $\Sigma$ |
| Aufg. 1 | 2 | 5 | 1 | 8 | | | | |
| Aufg. 2 | 2 | 2 | 4 | 8 | | | | |
| Aufg. 3 | 4 | 3 | 3 | 10 | | | | |
| Aufg. 4 | 4 | 5 | – | 9 | | | – | |
| Aufg. 5 | 5 | 5 | – | 10 | | | – | |
| Aufg. 6 | 1 | 3 | 5 | 9 | | | | |
| Aufg. 7 | $6 \times 1$ | | | 6 | | | | |
| $\Sigma$ | | | | 60 | | | | |

## **Problem 1:** Endliche Automaten $2 + 5 + 1 = 8$ Punkte

Sei für $n = \{1, 2, \ldots\}$ folgender nichtdeterministischer endlicher Automat $\mathcal{A}_n$ mit $n+1$ Zuständen gegeben, der eine Sprache über dem Alphabet $\Sigma = \{\mathtt{0}, \mathtt{1}\}$ erkennt.




(a) Geben Sie die Sprache, die $\mathcal{A}_n$ erkennt, in Mengenschreibweise an.

(b) Zeigen Sie, dass der Index der Neroderelation von $L(\mathcal{A}_n)$ mindestens $2^n$ ist.

(c) Was folgt aus dem Zusammenhang aus Teilaufgabe (b) für die Anzahl an Zuständen eines zu $\mathcal{A}_n$ äquivalenten deterministischen endlichen Automaten?

## **Problem 2:** Kontextfreie Grammatiken $2 + 2 + 4 = 8$ Punkte

Zu einer Sprache $L$ sei die Spiegelsprache $L^{\mathsf{R}}$ definiert als

$$L^{\mathsf{R}} = \left\{ w^{\mathsf{R}} \mid w \in L \right\}.$$

Dabei ist $w^{\mathsf{R}}$ die Spiegelung von $w$, d.h. $(w_1 w_2 \dots w_k)^{\mathsf{R}} = w_k w_{k-1} \dots w_1$.

(a) Sei $G = (\Sigma = \{a, b\}, V = \{S, A, B\}, S, R)$ mit folgender Produktionsmenge $R$

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

Geben Sie eine Grammatik $G^{\mathsf{R}}$ an, die $L(G)^{\mathsf{R}}$ erzeugt.

(b) Sei $G$ eine kontextfreie Grammatik. Beschreiben Sie eine allgemeine Vorgehensweise, um mithilfe von $G$ eine kontextfreie Grammatik $G^{\mathsf{R}}$ zu bestimmen, die die Sprache $L(G)^{\mathsf{R}}$ erzeugt.

(c) Beweisen Sie die Korrektheit Ihrer Vorgehensweise aus (b).

## **Problem 3:** Entscheidbarkeit 4 + 3 + 3 = 10 Punkte

Aus Vorlesung und Übung wissen Sie, dass die universelle Sprache

$$L_{\mathsf{u}} = \{\langle M, w\rangle \mid M \text{ akzeptiert } w\}$$

zwar semi-entscheidbar, aber nicht entscheidbar ist.

In dieser Aufgabe sollen Sie zeigen, dass weder die Sprache

$$L = \{\langle M\rangle \mid L(M) = \Sigma^*\}\,,$$

noch ihr Komplement $L^{\mathsf{c}}$ semi-entscheidbar ist.

(a) Zeigen Sie, dass $L^{\mathsf{c}}$ nicht semi-entscheidbar ist, indem Sie $L_{\mathsf{u}}$ auf $L$ reduzieren.

Aus einer gegebenen Turingmaschine $M$ zusammen mit einer Eingabe $w$ konstruieren wir folgendermaßen eine neue Turingmaschine $N$. Bei Eingabe von $x$ simuliert $N$ die Turingmaschine $M$ mit Eingabe $w$ für $|x|$ Schritte. Falls $M$ innerhalb von $|x|$ Schritten akzeptiert, lehnt $N$ die Eingabe $x$ ab. Ansonsten akzeptiert $N$ die Eingabe $x$.

(b) Zeigen Sie, dass $M$ die Eingabe $w$ genau dann nicht akzeptiert, wenn $L(N) = \Sigma^*$ gilt.

(c) Zeigen Sie, dass $L$ nicht semi-entscheidbar ist, indem Sie $L_{\mathsf{u}}^{\mathsf{c}}$ auf $L$ reduzieren.

## Problem 4: Kellerautomaten 4 + 5 = 9 Punkte

Sei

$$L = \{\alpha\#\beta \mid \alpha \text{ ist Teilwort von } \beta\}.$$

(a) Zeigen Sie, dass $L$ nicht kontextfrei ist.

Das Berechnungsmodell nichtdeterministischer Kellerautomat (PDA) lässt sich so ändern, dass sich der Lesekopf, der bei PDAs stets einen Schritt nach rechts geht, nun in zwei Richtungen bewegen kann, nämlich jeweils einen Schritt nach links oder nach rechts. Der Lesekopf kann außerdem das linke und rechte Ende des Eingabeworts erkennen. Ein LRPDA akzeptiert durch akzeptierende Zustände (und nicht durch leeren Stack). Ansonsten bleibt das Berechnungsmodell unverändert. Bezeichne solche Kellerautomaten als LRPDA.

(b) Beschreiben Sie das Verhalten eines LRPDAs, der $L$ erkennt. Gehen Sie insbesondere darauf ein, wie Sie die zusätzlichen Kopfbewegungen und den Nichtdeterminismus ausnutzen.

## Problem 5: NP-Vollständigkeit 5 + 5 = 10 Punkte

Das Entscheidungsproblem DEGREE-RESTRICTED SPANNING TREE ist wie folgt definiert.

| | |
|---|---|
| **Input:** | Graph $G = (V, E)$ |
| | Zahl $k \in \mathbb{N}$ |
| **Lösung:** | aufspannender Baum $T \subseteq G$ mit Maximalgrad höchstens $k$, |
| | d.h. $T$ ist zusammenhängend, kreisfrei und enthält alle Knoten von $G$ |
| | und jeder Knoten ist in höchstens $k$ Kanten von $T$ enthalten |

Für eine Zahl $k \in \mathbb{N}$ ist das Entscheidungsproblem DEGREE-$k$-RESTRICTED SPANNING TREE wie folgt definiert.

| | |
|---|---|
| **Input:** | Graph $G = (V, E)$ |
| **Lösung:** | aufspannender Baum $T \subseteq G$ mit Maximalgrad höchstens $k$, |
| | d.h. $T$ ist zusammenhängend, kreisfrei und enthält alle Knoten von $G$ |
| | und jeder Knoten ist in höchstens $k$ Kanten von $T$ enthalten |

(a) Ist das Problem DEGREE-RESTRICTED SPANNING TREE NP-vollständig? Beweisen Sie!
*Hinweis: Betrachten Sie den Fall $k = 2$.*

(b) Für welche $k \in \mathbb{N}$ ist ob das Problem DEGREE-$k$-RESTRICTED SPANNING TREE NP-vollständig? Beweisen Sie!
*Hinweis: Für geeignete Zahlen k können Sie* DEGREE-2-RESTRICTED SPANNING TREE *auf* DEGREE-$k$-RESTRICTED SPANNING TREE *reduzieren.*

Jedes der folgenden Entscheidungsprobleme ist NP-vollständig. Sie können diese Probleme für Ihre Reduktionen nutzen.

TRAVELING SALESMAN

**Input:** Graph $G = (V, E)$
Längenfunktion $\ell : E \to \mathbb{R}$
Zahl $k \in \mathbb{N}$

**Lösung:** geschlossene Tour $T$ durch alle Knoten von $G$ mit Gesamtlänge höchstens $k$,
d.h. $\sum_{e \in T} \ell(e) \leq k$.

HAMILTONPFAD

**Input:** Graph $G = (V, E)$

**Lösung:** Pfad $P$ durch alle Knoten von $G$

CLIQUE

**Input:** Graph $G = (V, E)$
Zahl $k \in \mathbb{N}$

**Lösung:** Menge $U \subseteq V$ von mindestens $k$ Knoten in der je zwei Knoten benachbart sind,
d.h. $|U| \geq k$ und für alle $u, v \in U$ mit $u \neq v$ gilt $\{u, v\} \in E$.

COLOR

**Input:** Graph $G = (V, E)$
Zahl $k \in \mathbb{N}$

**Lösung:** Knotenfärbung mit höchstens $k$ Farben in der keine gleichfarbigen Knoten benachbart sind,
d.h. $V = V_1 \cup \cdots \cup V_k$ und für $i = 1, \ldots, k$ und alle $u, v \in V_i$ gilt $\{u, v\} \notin E$.

## Problem 6: Approximationsalgorithmen $1 + 3 + 5 = 9$ Punkte

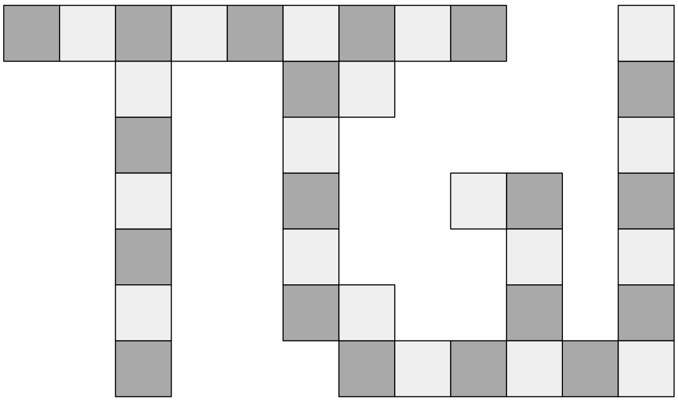
Das Optimierungsproblem NON-ATTACKING ROOK PLACEMENT ist wie folgt definiert.

**Input:** endlicher Ausschnitt $S$ des unendlichen Schachbretts in der Ebene

**Lösung:** Menge $T$ von Türmen auf $S$ die sich gegenseitig nicht bedrohen, d.h.
- jeder Turm in $T$ steht auf genau einem Feld in $S$,
- auf jedem Feld in $S$ steht höchstens ein Turm in $T$,
- kein Turm in $T$ kann einen anderen Turm in $T$ erreichen durch eine nur-horizontale oder nur-vertikale Bewegung entlang der Felder in $S$

**Ziel:** maximiere die Anzahl $k = |T|$ der Türme in $T$

(a) Zeichnen Sie in die folgende Instanz $S$ des NON-ATTACKING ROOK PLACEMENT Problems eine Lösung $T$ mit Wert $|T| = 9$ ein.



(b) Zeigen Sie, dass für jede Lösung $T$ des NON-ATTACKING ROOK PLACEMENT Problems jedes Feld von $S$ von höchstens zwei Türmen in $T$ bedroht wird.

(c) Zeigen Sie, dass der folgende Greedy-Algorithmus $\mathcal{A}$ ein polynomialer Approximationsalgorithmus für NON-ATTACKING ROOK PLACEMENT mit einer relativen Gütegarantie von 2 ist:

- Solange es ein Feld $f$ in $S$ gibt, dass von keinem bereits platzierten Turm bedroht wird, platziere einen Turm auf $f$.

## **Problem 7:** Gemischtes 6 Punkte

Sind die folgenden Aussagen korrekt? Begründen Sie jeweils kurz.

(a) Jede kontextfreie Sprache liegt in $P$.

(b) Eine Grammatik mit Startsymbol $S$ erzeugt genau dann das leere Wort, wenn $S \to \varepsilon$ eine Produktion ist.

(c) Wenn $\mathsf{P} = \mathsf{NP}$, dann sind alle Sprachen über dem Alphabet $\Sigma$ in $\mathsf{NP}$, bis auf $\Sigma^*$ und die leere Sprache, auch $\mathsf{NP}$-schwer.

(d) Jede kontextfreie Grammatik die nicht das leere Wort erzeugt kann in polynomieller Zeit in Greibach-Normalform umgewandelt werden.

(e) Kann ein Problem $\Pi$ in Polynomialzeit entschieden werden, wenn die Eingabe unär kodiert ist, so ist $\Pi$ nicht stark $\mathsf{NP}$-vollständig, solange $\mathsf{P} \neq \mathsf{NP}$.

(f) Der vom Huffman-Algorithmus berechnete Codierungsbaum ist eindeutig.