Prof. Dr. Roland Meyer
René Maseli

Probeklausur
Theoretische Informatik 2
6. Juli 2022

TU Braunschweig
Sommersemester 2022

1. **Bitte am Anfang ausfüllen:**

   Vorname:

   Nachname:

   Matrikelnummer:

   Nummer des Sitzplatzes:

   Unterschrift:

2. Die Nummer Ihrer Klausur ist `# 0`. Bitte merken Sie sich die Nummer. Wir werden Ihr Klausurergebnis anonymisiert unter Verwendung dieser Nummer bekanntgeben.
3. Achten Sie darauf, dass Ihre Klausur **vollständig ist** und **getackert bleibt** (20 Blätter).
4. Benutzen Sie **nur das an dieses Blatt angeheftete Papier**. Bei Bedarf können wir weitere Leerblätter austeilen. Wenn der Platz auf der Vorderseite des jeweiligen Aufgabenblatts nicht ausreicht, **machen Sie kenntlich**, wo Sie die Bearbeitung der Aufgabe fortsetzen.
5. Als Hilfsmittel sind **ausschließlich** Sprachwörterbücher sowie ein beidseitig **handschriftlich beschriebenes DIN A4-Blatt** erlaubt. Elektronische Geräte müssen während der Klausur ausgeschaltet bleiben. Täuschungsversuche werden als nicht bestanden gewertet und dem Prüfungsamt gemeldet.
6. Schreiben Sie leserlich und bearbeiten Sie Ihre Klausur mit einem **dokumentenechten Stift** (nicht mit Bleistift, kein Tipp-Ex, kein Tintenkiller) und **nicht in roter oder grüner Farbe**.
7. Wir werden die **Klausurergebnisse** auf unserer Website bekanntgeben: `tcs.cs.tu-bs.de/teaching/TheoInf2_SS_2022.html`.
8. Die **Bearbeitungszeit** beträgt **120 Minuten** (+ ggf. Zeit zum Lüften).
9. Mit **28 Punkten** ist die Klausur **sicher bestanden.**

Bepunktung: (wird von den Korrektoren ausgefüllt)

Aufgabe	1	2	3	4	5	6	7	Σ
Max.	10	10	10	10	10	10	10	**70**
Punkte								

1. Konstruktion einer DTM

10 Punkte

Konstruieren Sie eine **deterministische** Turingmaschine M, welche die Sprache

$$L = \left\{ a^m b^n \;\middle|\; m, n > 0 \text{ UND } m^2 < 3n \right\}$$

entscheidet. Beispielsweise sind $ab, aabb \in L$, aber $aab, aaabbb \notin L$.

- Erklären Sie die Arbeitsweise der Maschine ausführlich. Geben Sie insbesondere die Aufgabe jedes Kontrollzustands der Maschine an.
- Geben Sie die Transitionen der Maschine explizit an, z.B. in Form einer Tabelle oder als Zustandsgraph. Im Zustandsgraphen brauchen Sie Transitionen nach q_{rej} nicht zu zeichnen.
- Sie können wahlweise annehmen, dass das Band auf beiden Seiten der Eingabe mit ␣-Symbolen gefüllt ist, oder dass das Band auf der linken Seite durch ein \$-Symbol beschränkt ist. Geben Sie an, wofür Sie sich entschieden haben und geben Sie an, auf welches Symbol der Lese-/Schreibkopf initial zeigt.

Hinweis: Die Turingmaschine darf mehrere Bänder verwenden.

2. NL-Vollständigkeit

6 + 4 Punkte

Sei Σ eine endliche Menge. Ein Σ-gelabelter Graph $G = \langle V, E, \ell \rangle$ besteht aus einem gerichteten Graphen $\langle V, E\rangle$ mit $E \subseteq V \times V$ und einer Kantenlabel-Funktion $\ell\colon E \to \Sigma$, welche adjazente Knotenpaare $\langle u, v\rangle \in E$ auf ein Label $\ell(u, v) \in \Sigma$ abbildet.

Jeder Pfad p in G, bestehend aus einer Knotenfolge $p = \langle s, v_1, v_2, \dots, v_k, t\rangle$, beschreibt ein Wort über dem Alphabet Σ, genauer $\ell^*(p) := \ell(s, v_1)\ell(v_1, v_2)\dots\ell(v_k, t) \in \Sigma^*$.

Betrachten Sie das folgende Problem.

Pfadproblem mit Endlichem Automat (DFA-PATH)	
Gegeben:	Ein $\{0, 1\}$-gelabelter Graph $G = \langle V, E, \ell \rangle$, Knoten $s, t \in V$ und einen DFA A über dem Alphabet $\{0, 1\}$.
Entscheide:	Gibt es in G einen s-t-Pfad p mit $\ell^*(p) \in \mathcal{L}(A)$?

Zeigen Sie, dass DFA-PATH NL-vollständig (bzgl. logspace-many-one-Reduktionen) ist:

a) „Membership“: DFA-PATH $\in$ NL.

b) „Hardness“: DFA-PATH ist NL-schwer (bzgl. logspace-many-one-Reduktionen).

3. NP-Vollständigkeit

$4 + 6 = 10$ Punkte

Betrachten Sie das folgende Problem.

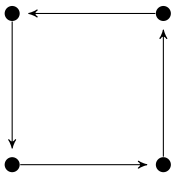
2-Hamilton-Pfade (2HP)

Gegeben: Ein gerichteter Graph $G = \langle V, E \rangle$.

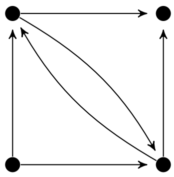
Entscheide: Gibt es zwei unterschiedliche Pfade in G, die jeweils alle Knoten besuchen?

Dabei sind zum Beispiel $G_1 \in$ 2HP und $G_2 \in$ 2HP, aber $G_3 \notin$ 2HP.

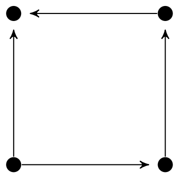
G_1 :



G_2 :



G_3 :



Zeigen Sie, dass 2HP NP-vollständig (bzgl. Polynomialzeit-Reduktionen) ist:

a) „Membership": 2HP $\in$ NP.

b) „Hardness": 2HP ist NP-schwer (bzgl. Polynomialzeit-Reduktionen).

4. Entscheidbarkeit II

10 Punkte

Betrachten Sie das folgende Problem.

Strict-Regular-Upward-Boundedness (SRUB)

Gegeben: DTM M und DFA A mit Eingabealphabet $\{0, 1\}$.

Entscheide: Ist $\mathcal{L}(M) \neq \mathcal{L}(A)$, aber liegt jedes $x \in \mathcal{L}(M)$ auch in $\mathcal{L}(A)$?

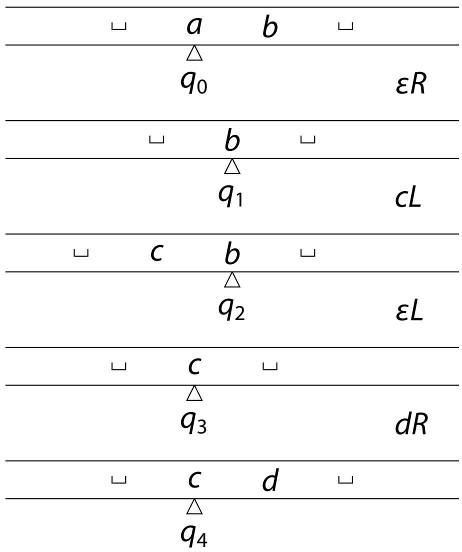
Beweisen Sie, dass SRUB weder semi-entscheidbar, noch co-semi-entscheidbar ist.

5. Insert-Delete-Maschinen

5+5=10 Punkte

Wir definieren eine alternative Turingmaschine, die statt Band-Zellen zu überschreiben und sich zu benachbarten Zellen zu bewegen, mittels vier Operationen Zellen einfügen und entfernen kann. Hierzu benutzen wir ε als Markierung für Lösch-Transitionen.

Zum Beispiel entfernt $\delta(q_0, a) = \langle \varepsilon, R, q_1 \rangle$ die aktuelle Zelle und geht nach rechts, $\delta(q_1, b) = \langle c, L, q_2 \rangle$ fügt eine Zelle mit c links neben der aktuellen Zelle ein, $\delta(q_2, b) = \langle \varepsilon, L, q_3 \rangle$ löscht die aktuelle Zelle und geht nach links und $\delta(q_3, c) = \langle d, R, q_4 \rangle$ fügt d in eine neue Zelle rechts daneben hinzu.



Auch hier gilt, Insert-Delete-Maschinen definieren eine Sprache aus genau den Wörtern, auf die eine Berechnung in einen akzeptieren Zustand möglich ist.

Beweisen Sie:

a) Jede durch einen Insert-Delete-Maschine akzeptierte Sprache ist semientscheidbar.

b) Jede semientscheidbare Sprache wird durch eine Insert-Delete-Maschine akzeptiert.

6. Quiz

2+2+3+3=10 Punkte

Bantworten Sie die folgenden Fragen. Begründen Sie Ihre Entscheidung mit einem kurzen Beweis oder einem Gegenbeispiel.

a) Ist die Menge der unentscheidbaren Sprachen abzählbar?

b) Falls das Problem A in NP liegt, aber B unentscheidbar ist, muss $A \cap B$ dann auch unentscheidbar sein?

c) Ist die folgende Schlussfolgerung korrekt? Weil das Circuit Value Problem P-hart bzgl. logspace-many-one-Reduktionen ist, aber mittels eines Linear-Platz-beschränkten Algorithmus gelöst werden kann, gibt es für jedes Problem in P einen LBA, der das Problem entscheidet.

d) Sei $f : \Sigma^* \to \Gamma^*$ eine Funktion mit $f(x) \in$ SELF-ACCEPT genau dann, wenn $x \in$ UNIVERSALITY für jedes $x \in \Sigma^*$. Muss f unberechenbar sein?

7. Berechenbarkeit

10 Punkte

Es sei $\Sigma = \{0, 1\}$. Betrachten Sie die Funktion *shallowWords* $: \Sigma^* \times \mathbb{N} \to (\mathbb{N} \cup \{\infty\})$, die für die Kodierung $w \in \Sigma^*$ einer deterministischen Turingmaschine M_w und eine Schranke n, die Anzahl der Wörter ausgibt, die von M_w in höchstens n Schritten akzeptiert werden. Dabei soll ∞ ausgegeben werden, falls es unendlich viele solcher Eingaben gibt.

Beweisen Sie, dass die Funktion *shallowWords* berechenbar ist.

Geben Sie hierzu einen Algorithmus (als Pseudo-Code) an.

Hinweis: Die Kodierung der Zahlen in der Ein- und Ausgabe der Funktion (z.B. unär oder binär) ist für die Bearbeitung der Aufgabe unerheblich.