Abschlussklausur Theoretische Informatik 2
15. März 2018

Prof. Dr. Roland Meyer
Sebastian Muskalla
TU Braunschweig
Wintersemester 2017/18

1. **Bitte ausfüllen:**

   Vorname:

   Nachname:

   Matrikelnummer:

   Unterschrift:

2. Achten Sie darauf, dass Ihre Klausur vollständig ist und getackert bleibt (16 Blätter)!
3. 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.
4. Als Hilfsmittel ist **ausschließlich** ein beidseitig **handgeschriebenes DIN A4-Blatt** erlaubt. Elektronische Geräte müssen während der Klausur ausgeschaltet bleiben. Täuschungsversuche werden mit 0 Punkten gewertet und dem Prüfungsamt gemeldet.
5. Schreiben Sie leserlich und bearbeiten Sie Ihre Klausur **nicht mit Bleistift** und auch **nicht in roter oder grüner Farbe!**
6. Wir werden das Deckblatt während der Klausur auf korrekte Daten überprüfen. Legen Sie dazu Ihren **Studierendenausweis** und einen **amtlichen Lichtbildausweis** bereit.
7. Wir werden Ergebnisse und Termin der Klausureinsicht auf unserer Website bekanntgeben: `tcs.cs.tu-bs.de/teaching/ThInfKlausur_WS_20172018.html`.
8. Die **Bearbeitungszeit** beträgt **180 Minuten**.
9. Mit **40 Punkten** ist die Klausur **sicher bestanden.**

Punkteverteilung: (wird von den Korrektoren ausgefüllt)

Aufgabe	1	2	3	4	5	6	7	8	9	10	Σ
Max.	10	10	10	10	10	10	10	10	10	10	**100**
Punkte											

1. Konstruktion einer DTM

$\boxed{\text{10 Punkte}}$

Es sei $\Sigma = \{a, b, c\}$. Für ein Wort $w \in \Sigma^*$ bezeichnen wir mit $|w|_a$ die Anzahl der a's in w, analog $|w|_b$ und $|w|_c$.

Konstruieren Sie eine deterministische Turing-Maschine $\mathcal{M}$, welche die Sprache

$$\mathcal{L} = \left\{ w \in \Sigma^* \;\middle|\; |w|_a - |w|_c \leqslant |w|_b \leqslant |w|_a + |w|_c \right\}$$

entscheidet. Geben Sie dabei eine formale Beschreibung von $\mathcal{M}$ als Tupel, sowie eine ausführliche Erklärung der Arbeitsweise von $\mathcal{M}$ an.

Hinweis: Ihre Turing-Maschine darf mehrere Bänder verwenden.

2. Berechenbarkeit

$\boxed{8 + 2 = 10 \text{ Punkte}}$

Es sei $\Sigma = \{0, 1\}$. Betrachten Sie die Funktion *benchmark* $: (\Sigma^*)^4 \to \Sigma^*$, die wie folgt definiert ist:

$$\mathit{benchmark}(w, x, t, s) = \begin{cases} w, & \text{falls } \mathrm{Time}_{M_w}(x) \leqslant t,\ \mathrm{Space}_{M_w}(x) \leqslant s\ , \\ 0, & \text{falls } \mathrm{Time}_{M_w}(x) > t\ , \\ 1, & \text{falls } \mathrm{Time}_{M_w}(x) \leqslant t, \mathrm{Space}_{M_w}(x) > s\ . \end{cases}$$

Dabei ist $w \in \Sigma^*$ die Kodierung einer deterministischen Turing-Maschine, $x \in \Sigma^*$ eine Eingabe und $t, s \in \Sigma^*$ binär kodierte Zahlen. Wie üblich ist $\mathrm{Time}_{M_w}(x)$ die Anzahl der Schritte, die M_w auf Eingabe x zum Halten braucht und $\mathrm{Space}_{M_w}(x)$ die maximale Anzahl Zellen, die während der Berechnung auf dem Band belegt sind.

a) Beweisen Sie, dass *benchmark* berechenbar ist.
Geben Sie hierzu einen Algorithmus (als Pseudo-Code) an.

b) Geben Sie Zeit- und Platzkomplexität Ihres Algorithmus in der Größe der Eingabe an.

3. Turing-Maschinen mit Kosten

$\boxed{7 + 3 = 10 \text{ Punkte}}$

Eine Turing-Maschine mit Kosten (KTM) $M^K = (Q, \Sigma, \Gamma, \delta, q_0, B, k)$ besteht aus

- der nicht-deterministischen Turing-Maschine (NTM) $(Q, \Sigma, \Gamma, \delta, q_0)$,
- dem **initialen Budget** $B \in \mathbb{N}$,
- einer **Kostenfunktion** k, die jeder Transition t aus $\delta(Q \times \Gamma)$ ihre echt positiven Kosten $k(t) \in \mathbb{N}, k(t) > 0$ zuordnet.

Eine Konfiguration einer solchen Maschine ist von der Form $(u\ q\ av, b)$, wobei $u\ q\ av$ eine Konfiguration der NTM und $b \in \mathbb{N}$ das **verbleibende Budget** ist. Die Initiale Konfiguration zu Eingabe x ist $(\varepsilon\ q_0\ \$x, B)$. Das Ausführen einer Transition $t \in \delta(q, a)$ in Konfiguration $(u\ q\ av, b)$ ist nur möglich, wenn $b \geqslant k(t)$ erfüllt ist. In diesem Fall führt es zu Konfiguration $(c', b - k(t))$, wobei c' die wie üblich definierte Nachfolgekonfiguration der NTM ist. (Wenn das verbleibende Budget zu klein ist, um irgend eine Transition auszuführen, bleibt die Maschine stecken und weist ab.)

Die Sprache $\mathcal{L}\left(M^K\right)$ von M^K ist die Menge aller Wörter x, zu denen es eine Transitionsfolge von $(\varepsilon\ q_0\ \$x, B)$ zu $(u\ q_{accept}\ v, b)$ gibt (mit u, v, b beliebig).

a) Zeigen Sie, dass Turing-Maschinen mit Kosten von herkömmlichen Entscheidern simuliert werden können. Erklären Sie dazu, wie man zu einer gegebene M^K eine totale (immer haltende) Turing-Maschine (gegebenenfalls nicht-deterministisch und mit mehreren Bändern) konstruiert, die die selbe Sprache akzeptiert.

   Eine Erklärung der Simulation unter Verwendung von Pseudo-Code ist ausreichend, Sie müssen die Transitionen nicht formal angegeben. Begründen Sie, warum Ihre Maschine ein Entscheider ist und die gleiche Sprache akzeptiert.

b) Zeigen Sie: Es gibt ein entscheidbares Problem $\mathcal{L}$, so dass es **keine** Turing-Maschine mit Kosten M^K gibt mit $\mathcal{L}\left(M^K\right) = \mathcal{L}$.

Fortsetzung der Bearbeitung von Aufgabe 3

4. Konstantenweitergabe

$\boxed{3 + 1 + 6 = 10 \text{ Punkte}}$

Wir betrachten in dieser Aufgabe den vollständigen Verband $\mathcal{D} = (\mathbb{N} \cup \{\bot, \top\}, \leqslant)$, wobei $\leqslant$ durch die folgenden Regeln vollständig definiert ist:

$$\bot \leqslant \top \qquad \bot \leqslant n \;\; \forall n \in \mathbb{N} \qquad n \leqslant \top \;\; \forall n \in \mathbb{N}$$

a) Geben Sie die folgenden Joins und Meets an.

$\bot \sqcup \top =$ ________	$\bot \sqcup 3 =$ ________
$\top \sqcup 4 =$ ________	$1 \sqcup 2 =$ ________
$5 \sqcap 6 =$ ________	$\bigsqcup \mathbb{N} =$ ________

b) Der Verband $\mathcal{D}$ ist unendlich. Warum lässt er sich dennoch für Datenflussanalyse verwenden?

c) Betrachten Sie das folgende Program.

$[x := 4]^1$
$[x := x + 2]^2$
while $[x > 0]^3$ **do**
  $[x := x - 2]^4$
end while
$[skip]^5$

Führen Sie eine Vorwärts-Datenflussanalyse unter Verwendung des Verbandes $\mathcal{D}$ durch, betrachten Sie also das Datenflusssystem

$$(G, \mathcal{D}, \bot, TF)\,.$$

Geben Sie zunächst den Kontrollflussgraphen G und die Transferfunktionen TF zum Program an, stellen Sie das induzierte Gleichungssystem auf und bestimmten Sie seine kleinste Lösung. Die Bedeutung der Datenflusswerte ist wie folgt:

- $\bot$: x ist am Eingang des Blocks nicht initialisiert,
- $n \in \mathbb{N}$: x ist am Eingang des Blocks konstant n (hat also garantiert Wert n),
- $\top$: x ist am Eingang des Blocks (potentiell) nicht konstant.

Bearbeitung von Aufgabe 4

5. Entscheidbarkeit

10 Punkte

Betrachten Sie die Sprache aller Kodierungen von Turing-Maschinen, deren Sprache aus genau einem Wort besteht,

$$\mathcal{L} = \left\{ w \in \{0,1\}^* \;\middle|\; \exists x \in \{0,1\}^* \colon \mathcal{L}(M_w) = \{x\} \right\}.$$

Beweisen Sie, dass $\mathcal{L}$ nicht entscheidbar ist. Verwenden Sie **nicht** den Satz von Rice.

6. NL-Vollständigkeit

$\boxed{5 + 5 = 10 \text{ Punkte}}$

Betrachten Sie das folgende Problem:

Double Path (DP)	
Gegeben:	Ein gerichteter Graph $G = (V, E)$ und Knoten $s, t \in V$.
Entscheide:	Gibt es in G zwei knotenverschiedene Pfade von s nach t?

Hierbei nennen wir zwei Pfade **knotenverschieden**, wenn es mindestens einen Knoten gibt, der in einem der Pfade vorkommt, im anderen jedoch nicht.

Zeigen Sie, dass DP NL-vollständig (bezüglich Logspace-Reduktionen) ist:

a) "Membership": DP $\in$ NL.

b) "Hardness": DP ist NL-schwer (bezüglich Logspace-Reduktionen).

7. NP-Vollständigkeit

4 + 6 = 10 Punkte

Betrachten Sie das folgende Problem:

Loopy Cycle (LC)

Gegeben: Ein gerichteter Graph $G = (V, R)$ und ein Knoten $v \in V$.

Entscheide: Gibt es in G einen Loopy Cycle in v?

Hierbei ist ein **Loopy Cycle** in v ein Pfad

$$v \to v_1 \to v_2 \to \ldots \to v_{\ell-1} \to v \to v_{\ell+1} \to v_{\ell+2} \to \ldots \to v_k \to v \,,$$

der in v beginnt und endet, v ein weiteres mal besucht und jeden anderen Knoten aus V genau ein einziges Mal beinhaltet.

Zeigen Sie, dass LC NP-vollständig (bezüglich Polynomialzeit-Reduktionen) ist:

a) “Membership”: LC $\in$ NP.

b) “Hardness”: LC ist NP-schwer (bezüglich Polynomialzeit-Reduktionen).

8. Quiz

$\boxed{2 + 2 + 2 + 2 + 2 = 10 \text{ Punkte}}$

Beweisen oder widerlegen Sie die folgenden Aussagen. Begründen Sie Ihre Antwort mit einem kurzen Beweis oder einem Gegenbeispiel.

a) Sei $\mathcal{M}$ eine Turing-Maschine über dem Eingabealphabet Σ. Es gibt immer eine Turing-Maschine $\mathcal{M}'$ mit $\mathcal{L}(\mathcal{M}') = \Sigma^* \setminus \mathcal{L}(\mathcal{M})$.

b) Das Komplement einer semi-entscheidbaren, unentscheidbaren Sprache ist nie semi-entscheidbar.

c) Aus PSPACE = NP würde PSPACE = coNP folgen.

d) Jedes PSPACE-vollständige Problem ist auch NP-schwer (jeweils bzgl. Polynomialzeit-Reduktionen).

e) Es gibt ein Problem aus NL, das PSPACE-vollständig (bzgl. Logspace-Reduktionen) ist.
Hinweis: NL $\subsetneq$ PSPACE.

Fortsetzung der Bearbeitung von Aufgabe 8

9. 2D-Rechenmaschinen

$\boxed{8 + 2 = 10 \text{ Punkte}}$

Eine **2D-Rechenmaschine** verwendet statt eines Bands ein in alle vier Richtungen (links, rechts, oben, unten) unendliches kariertes Papier als Speicher. Formal ist eine solche Maschine als Tupel $\mathcal{R} = (Q, \Gamma, \delta, q_0)$ definiert. Dabei ist Q die Menge der Kontrollzustände (mit $q_{accept} \in Q$), $q_0 \in Q$ der Initialzustand, Γ das Papieralphabet (mit $\sqcup \in \Gamma$). Die (deterministische) Transitionsfunktion ist vom Typ

$$\delta\colon Q \times \Gamma \to Q \times \Gamma \times \{L, R, O, U\}\ .$$

Eine Konfiguration ist von der Form $(q, (i, j), v)$, hierbei ist $q \in Q$ ein Kontrollzustand, $(i, j) \in \mathbb{Z} \times \mathbb{Z}$ die Kopfposition auf dem Papier und $v : \mathbb{Z} \times \mathbb{Z} \to \Gamma$ der Papierinhalt (das heißt, v ordnet jeder Zelle $(n, m) \in \mathbb{Z} \times \mathbb{Z}$ ihren Inhalt $v(n, m) \in \Gamma$ zu). Die Initialkonfiguration ist $(q_0, (0, 0), v_0)$ mit $v_0(n, m) = \sqcup$ für alle (n, m). Die Transitionsrelation $\to$ auf Konfigurationen wird durch δ induziert: Wenn der aktuelle Kontrollzustand q und der Papierinhalt an der Kopfposition a ist, und $\delta(q, a) = (q', b, d)$, dann überschreibt die Maschine a durch b, geht in Kontrollzustand q' und bewegt den Kopf eine Zelle nach links/rechts/oben/unten (für $d = L/R/O/U$).

2D-Rechenmaschinen erhalten keine Eingabe und starten immer mit leerem Papier.

a) Zeigen Sie, dass eine 2D-Rechenmaschine durch eine deterministische Turingmaschine simuliert werden kann: Erklären Sie, wie man zu einer gegebenen 2D-Rechenmaschine $\mathcal{R}$ eine DTM $\mathcal{M}$ (gegebenfalls mit mehreren Bändern) konstruiert, so dass $\mathcal{R}$ nach endlich vielen Schritten akzeptiert (eine Konfiguration mit Kontrollzustand q_{accept} erreicht), genau dann, wenn $\mathcal{M}$ die leeren Eingabe ε akzeptiert.

   Eine Erklärung der Simulation unter Verwendung von Pseudo-Code ist ausreichend, Sie müssen die Transitionen nicht formal angegeben.

b) In Aufgabtenteil a) haben Sie gezeigt, dass das Halteproblem für 2D-Rechenmaschinen semi-entscheidbar ist. Begründen Sie nun, dass das Halteproblem für 2D-Rechenmaschinen nicht entscheidbar ist. (Sie müssen keinen formalen Beweis führen.)

Fortsetzung der Bearbeitung von Aufgabe 9

10. Drei Probleme

$\boxed{3 + 7 = 10 \text{ Punkte}}$

Es sei Σ^* ein Alphabet und $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3 \subseteq \Sigma^*$ drei Sprachen mit $\Sigma^* = \mathcal{L}_1 \cup \mathcal{L}_2 \cup \mathcal{L}_3$ und $\mathcal{L}_1 \cap \mathcal{L}_2 = \mathcal{L}_2 \cap \mathcal{L}_3 = \mathcal{L}_1 \cap \mathcal{L}_3 = \varnothing$.

a) Angenommen, $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$ sind alle semi-entscheidbar. Beweisen Sie, dass dann $\mathcal{L}_1$ entscheidbar ist.

b) Angenommen, $\mathcal{L}_2$ ist NL-vollständig (bezüglich Logspace-Reduktionen) und $\mathcal{L}_3$ ist in L. Beweisen Sie, dass dann $\mathcal{L}_1$ NL-vollständig (bezüglich Logspace-Reduktionen) ist.

   Hinweis: Betrachten Sie das Komplementproblem zu $\mathcal{L}_1$.

Zusätzliches Blatt