Abschlussklausur Theoretische Informatik II
18. Juli 2017

Prof. Dr. Roland Meyer
Peter Chini, Sebastian Muskalla
TU Braunschweig
Sommersemester 2017

1. **Bitte ausfüllen:**

   Vorname:

   Nachname:

   Matrikelnummer:

   Unterschrift:

2. Achten Sie darauf, dass Ihre Klausur vollständig ist und getackert bleibt (17 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 zweiseitig **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 den Termin für die Klausureinsicht auf unserer Website bekanntgeben: `tcs.cs.tu-bs.de/teaching/ThInf_SS_2017.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 $\Sigma = \{a, b\}$ und $w \in \Sigma^*$ ein Wort. Wir bezeichnen mit $|w|_a$ die Anzahl der a's in w. Analog verwenden wir $|w|_b$ für die Anzahl der b's in w.

Konstruieren Sie eine deterministische Turing-Maschine M, welche die Sprache

$$\mathcal{L} = \{ w \in \Sigma^* \mid |w|_b \leqslant |w|_a \leqslant 2 \cdot |w|_b \}$$

entscheidet. Geben Sie dabei eine formale Beschreibung von M als Tupel, sowie eine ausführliche Erklärung der Arbeitsweise von M an.

Hinweis: Ihre Turing-Maschine darf mehrere Bänder verwenden.

2. Reaching Definitions

$\boxed{2 + 3 + 5 = 10 \text{ Punkte}}$

Führen Sie für das folgende Programm P eine Reaching-Definitions-Analyse durch. Beachten Sie, dass es sich hierbei um eine Vorwärts-May-Analyse handelt.

```
[x := 2]^1
while [x < 5]^2 do
    if [y = 7]^3 then
        [y := y + 2]^4
    else
        [x := x - 2]^5
    end if
end while
[skip]^6
```

a) Zeichnen Sie den Kontrollflussgraphen G von P.

b) Betrachten Sie den Verband

$$(D, \leqslant) = \left(\mathcal{P}\left(\{x, y\} \times \left(\{1, \ldots, 6\} \uplus \{?\}\right)\right), \subseteq\right) .$$

Intuitiv bedeutet (x, i), dass der letzte Schreibzugriff auf x in Block i erfolgte (bzw. x noch nicht initialisiert ist, falls i =?). Geben Sie für die Blöcke 1 - 6 geeignete monotone Transferfunktionen über diesem Verband an.

c) Betrachten Sie das Datenflusssystem

$$\left(G, (D, \leqslant), \{(x, ?), (y, ?)\}, TF\right) ,$$

wobei TF die Transferfunktionen aus Teil b) sind. Geben Sie das induzierte Gleichungssystem an und bestimmen Sie seine kleinste Lösung mit dem Satz von Kleene.

3. Berechenbarkeit

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

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

$$\mathit{faster}(w_1, w_2, x, c) = \begin{cases} w_1, & \text{falls } \mathrm{Time}_{M_{w_1}}(x) \leqslant c,\ \ \mathrm{Time}_{M_{w_1}}(x) < \mathrm{Time}_{M_{w_2}}(x)\,, \\ w_2, & \text{falls } \mathrm{Time}_{M_{w_2}}(x) \leqslant c,\ \ \mathrm{Time}_{M_{w_2}}(x) < \mathrm{Time}_{M_{w_1}}(x), \\ 1, & \text{falls } \mathrm{Time}_{M_{w_1}}(x) = \mathrm{Time}_{M_{w_2}}(x) \leqslant c \\ 0, & \text{falls beide Maschinen nicht innerhalb von } c \text{ Schritten halten.} \end{cases}$$

Dabei sind $w_1, w_2 \in \Sigma^*$ Kodierungen von Turing-Maschinen, $x \in \Sigma^*$ eine Eingabe und $c \in \Sigma^*$ die binär kodierte Anzahl an Schritten, die höchstens erlaubt ist. Für $i \in \{1, 2\}$ ist $\mathrm{Time}_{M_{w_i}}(x)$ die Anzahl der Schritte, die M_{w_i} auf Eingabe x zum Halten braucht.

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

b) Was ist die Zeitkomplexität Ihres Algorithmus?

4. GOTO-Maschinen

10 Punkte

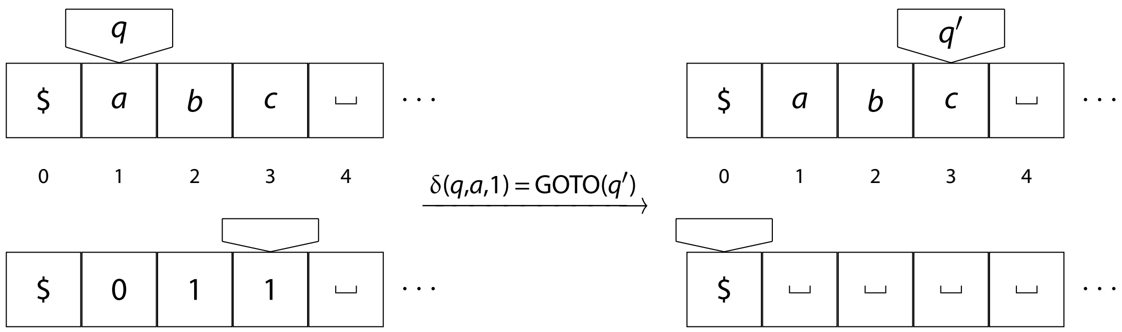
Eine GOTO-Turing-Maschine (GTM) M^G ist eine deterministische Turing-Maschine, die zusätzlich zu normalen Transitionen Sprungbefehle (GOTO) in beliebige Zellen auf dem Band ausführen kann. Formal hat eine GTM M^G zwei Bänder: Auf dem ersten Band (Eingabeband) steht initial die Eingabe und es kann wie gewöhnlich verwendet werden. Auf das zweite Band (Adressband) kann die Maschine die Binärkodierung einer Zahl i schreiben. Intuitiv ist dies die Zieladresse des Sprungbefehls.

$\delta(q, a, a')$ ist entweder eine gewöhnliche Transition einer Zwei-Band-DTM, oder eine GOTO-Transition von der Form $\delta(q, a, a') = \text{GOTO}(q')$. Wenn M^G eine solche GOTO-Transition benutzt, und in diesem Moment $\text{bin}(i)$ der Inhalt des Adressbands ist, dann

- wechselt M^G in den Kontrollzustand q',
- die Kopfposition auf dem Eingabeband wechselt zur i-ten Zelle, der Bandinhalt wird nicht geändert und
- der Inhalt des Adressbands wird gelöscht und der Kopf zeigt auf das \$-Symbol.

Konfigurationen und Sprache von M^G sind wie bei gewöhnlichen Turing-Maschinen definiert.

Beispiel:



Zeigen Sie, dass GOTO-Turing-Maschinen von herkömmlichen Turing-Maschinen simuliert werden können. Sei M^G eine gegebene GTM. Erklären Sie, wie eine DTM M (gegebenenfalls mit mehreren Bändern) mit $\mathcal{L}\left(M^G\right) = \mathcal{L}(M)$ konstruiert werden kann.

5. NL-Vollständigkeit

3 + 7 = 10 Punkte

Betrachten Sie das folgende Problem:

One-way-reachability (OWR)

Gegeben: Ein gerichteter Graph $G = (V, E)$ und Knoten $s, t \in V$.

Entscheide: Gibt es in G einen Pfad von s nach t oder einen Pfad von t nach s?

Zeigen Sie, dass OWR NL-vollständig (bezüglich logspace-many-one-Reduktionen) ist:

a) „Membership“: OWR $\in$ NL.

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

6. Entscheidbarkeit

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

Betrachten Sie die folgende Sprache über dem Alphabet $\Sigma = \{0, 1, \#\}$:

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

a) Beweisen Sie, dass $\mathcal{L}$ semi-entscheidbar ist.

b) Beweisen Sie, dass $\mathcal{L}$ nicht entscheidbar ist.

7. NP-Vollständigkeit

3 + 7 = 10 Punkte

Betrachten Sie das folgende Problem:

<u>*k*-**PATH**</u>

Gegeben: Ein gerichteter Graph $G = (V, E)$ und eine natürliche Zahl k.

Entscheide: Gibt es in G einen einfachen Pfad der Länge k?

Ein einfacher Pfad ist ein Pfad, der keine Knoten wiederholt.

Beachten Sie, dass Sie davon ausgehen müssen, dass die Zahl k binär kodiert ist.

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

a) „Membership“: k-PATH $\in$ NP.

b) „Hardness“: k-PATH ist NP-schwer (bzgl. Polynomialzeit-Reduktionen).

8. Quiz

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

a) Beantworten Sie die folgenden Fragen. Begründen Sie Ihre Antwort mit einem kurzen Beweis oder einem Gegenbeispiel.

(1) Es seien $\mathcal{L}_1$ und $\mathcal{L}_3$ entscheidbar und $\mathcal{L}_2$ eine Sprache mit $\mathcal{L}_1 \subseteq \mathcal{L}_2 \subseteq \mathcal{L}_3$.
Ist $\mathcal{L}_2$ dann auch entscheidbar?

(2) Wenn $\mathcal{L}$ entscheidbar und $\mathcal{L}'$ semi-entscheidbar, ist $\mathcal{L} \cap \mathcal{L}'$ dann entscheidbar?

(3) Wenn $\mathcal{L}$ semi-entscheidbar und $\overline{\mathcal{L}}$ semi-entscheidbar, ist $\mathcal{L}$ dann entscheidbar?

b) Die Komplexitätsklasse EXP der Probleme, die sich deterministisch in exponentieller Zeit entscheiden lassen, war definiert als: EXP $= \bigcup_{k\in\mathbb{N}} \mathrm{DTIME}\left(2^{\mathcal{O}\left(n^k\right)}\right)$. Wir erweitern diese Definition. Für $m \geqslant 1$ ist

$$\mathrm{mEXP} = \bigcup_{k\in\mathbb{N}} \mathrm{DTIME}(\underbrace{2^{2^{\cdot^{\cdot^{\cdot^{2^{\mathcal{O}\left(n^k\right)}}}}}}}_{m\text{-fach exponentiell}})$$

die Klasse aller Probleme, die sich deterministisch in m-fach exponentieller Zeit entscheiden lassen. Analog werden auch mNEXP und mEXPSPACE definiert.

Betrachten Sie nun die Klasse:

$$\mathrm{ELEMENTARY} = \bigcup_{m\geqslant 1} \mathrm{mEXP}.$$

Beweisen Sie folgende Eigenschaften:

(1) ELEMENTARY $= \bigcup_{m\geqslant 1}$ mNEXP.

(2) ELEMENTARY $= \bigcup_{m\geqslant 1}$ mEXPSPACE.

9. Kürzeste Pfade

5 + 5 = 10 Punkte

Betrachten Sie die folgenden beiden Probleme.

Kein Pfad der Länge *k* (NO-LENGTH-*k*-PATH)

Gegeben: Ein gerichteter Graph $G = (V, E)$, Knoten $s, t \in V$ und eine natürliche Zahl k.

Entscheide: Gibt es in G keinen Pfad von s nach t der Länge höchstens k?

Kürzester Pfad (SHORTEST-PATH)

Gegeben: Ein gerichteter Graph $G = (V, E)$, Knoten $s, t \in V$ und eine natürliche Zahl k.

Entscheide: Hat ein kürzester Pfad von s nach t in G Länge *genau* k?

Ein kürzester Pfad von s nach t ist ein Pfad $s = v_0 \to v_1 \ldots \to v_m = t$ der Länge m, so dass es keinen Pfad der Länge $\leqslant m$ - 1 von s nach t gibt.

Beachten Sie, dass Sie bei beiden Problemen davon ausgehen müssen, dass die Zahl k binär kodiert ist.

a) Beweisen Sie: NO-LENGTH-*k*-PATH ist in NL.

b) Beweisen Sie: SHORTEST-PATH ist in NL.

10. Spezielles PCP

3 + 7 = 10 Punkte

Betrachten Sie die folgende Variante des PCP.

Spezielles Postsches Korrespondenzproblem (SPCP)

Gegeben: Eine endliche Sequenz von Tupeln aus Wörtern $(x_1, y_1), \ldots, (x_k, y_k)$

Entscheide: Gibt es eine endliche, nicht-leere Sequenz von Indizes $i_1 \ldots i_n$ mit $x_{i_1} x_{i_2} \ldots x_{i_n} = y_{i_1} y_{i_2} \ldots y_{i_n}$ und $i_n = 1$?

a) Beweisen Sie, dass das SPCP semi-entscheidbar ist.

b) Beweisen Sie, dass das SPCP nicht entscheidbar ist.