Abschlussklausur
Theoretische Informatik 2
31. März 2022

Prof. Dr. Roland Meyer
Thomas Haas
TU Braunschweig
Wintersemester 2021/2022

1. Lesen und unterschreiben Sie die **Erklärung nach Abschluss der Online-Prüfung**, welche Sie im Dateiordner finden. Laden Sie diese unbedingt in den dafür vorgesehenen **Abgabeordner** hoch, Ihre Klausur gilt sonst als **nicht bestanden**. Dieser Abgabeordner wird den ganzen Tag zur Verfügung stehen also können Sie ihre Erklärung auch nach Beendigung der Klausur hochladen.

   Falls Sie die Erklärung weder händisch noch digital unterschreiben können, schreiben Sie eine E-Mail an *t.haas@tu-bs.de* und *a.soleinsky@tu-bs.de*. Geben Sie in dieser E-Mail **alle Informationen an**, die in der Erklärung abgefragt werden.

2. Schreiben Sie **leserlich** und **nummerieren** Sie die Seiten Ihrer Abgabe.

3. **Laden** Sie bis zum Ende der Klausur Ihre **Abgabe** in den dafür vorgesehenen **Abgabeordner hoch**. Sie können Ihre **Abgabe** dazu **abfotografieren**, **einscannen** oder **direkt als PDF** per Tablet oder ähnlichem erstellen. Abgaben als **.pdf, .jpg** oder sonstigen **Standardformaten** sind möglich.

4. Der Dateiname Ihrer Klausur soll MatNr_Nachname sein. Der Dateiname Ihrer Erklärung nach Abschluss der Online-Prüfung soll MatNr_Nachname_Erklaerung sein.

   Beispiel: 4444444_Mustermann_Max.pdf und
   4444444_Mustermann_Max_Erklaerung.pdf

   Falls Sie mehrere Dateien abgeben, fügen Sie entsprechende Suffixe in der Form _Suffix hinzu (z.B. 4444444_Mustermann_Max_2.pdf für die zweite Seite der Abgabe).

5. Sie dürfen das **Skript** und ihre **eigenen Aufzeichnungen** verwenden. Das Heranziehen **fremder Hilfe** (z.B. andere Studenten oder Internetforen) ist **untersagt**.

6. Wenn Sie im Laufe der Klausur **Fragen** haben, steht Ihnen folgender BBB-Raum zur Verfügung:
   `https://webconf.tu-bs.de/tho-6n6-t7e`
   Stellen Sie Ihre Frage über den öffentlichen Chat. **Nach Beantwortung** Ihrer Frage, **verlassen** Sie bitte den BBB-Raum wieder.

7. Im Falle von Auftreten **technischer Probleme**, machen Sie bitte **Beweisfotos** und melden Sie anschließend die Probleme telefonisch unter +49 531-391-9522.

8. Wir werden den Termin für die Klausureinsicht auf unserer Website bekanntgeben:
   `tcs.cs.tu-bs.de/teaching/TheoInf2Klausur_WS_20212022.html`.

9. Die **Bearbeitungszeit** beträgt **240 Minuten**. 60 Minuten Extrazeit für den technischen Zusatzaufwand sind bereits eingerechnet. Laden Sie die Klausur **rechtzeitig** hoch!

10. Mit **40 Punkten** ist die Klausur **sicher bestanden.**

1. TM-Konstruktion

10 Punkte

Betrachten Sie die Sprache $\mathcal{L} = \{a^n.b^m.c^k \mid n, m, k > 0 \text{ und } n - m \leqslant k < n + m\} \subseteq \{a, b, c\}^*$.

Konstruieren Sie eine DTM M, die diese Sprache akzeptiert.

- 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.
- 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.

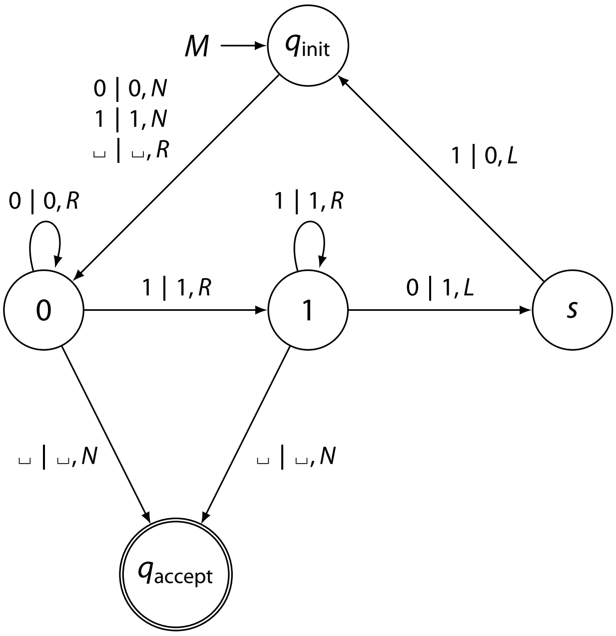
Hinweis: Sie dürfen auch mehrere Bänder verwenden.

Zu Aufgabe 1:

2. TM-Analyse

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

Betrachten Sie die TM M mit Eingabealphabet $\Sigma = \{0, 1\}$, Bandalphabet $\Gamma = \{0, 1, \sqcup\}$ und Zustandsmenge $Q = \{q_{\text{init}}, q_{\text{accept}}, 0, 1, s\}$. Hierbei ist q_{init} der Startzustand und q_{accept} der akzeptierende Zustand.



a) Beschreiben Sie die Arbeitsweise von M in Worten und geben Sie diese als Pseudo-Code an.

b) Interpretieren Sie M nun als Entscheider. Was ist die von M akzeptierte Sprache?

c) Interpretieren Sie M nun als Berechner. Was ist die von M berechnete Funktion f?

Hinweise: Bei Eingabe w startet die Maschine in der Konfiguration $\ldots\sqcup q_{\text{init}}\, w\sqcup\ldots$, d.h. links und rechts von der Eingabe befinden sich $\sqcup$-Symbole und der Lese-/Schreibkopf zeigt auf den ersten Buchstaben der Eingabe. Der Funktionswert $f(w)$ ist definiert als w', wenn M bei Eingabe w mit Bandinhalt w' akzeptiert. Alle fehlenden Transitionen führen dazu, dass die Maschine die Berechnung beendet und keine Ausgabe erzeugt.

Zu Aufgabe 2:

3. NL-Vollständigkeit

$5 + 5 = 10$ Punkte

Betrachten Sie das folgende Problem.

Double reachability (DR)

Gegeben: Ein gerichteter Graph $G = (V, E)$ und Knoten s_1, s_2, t mit $s_1 \neq s_2$.

Entscheide: Gibt es in G einen Pfad von s_1 nach t und einen von s_2 nach t?

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

a) „Membership“: DR $\in$ NL.

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

4. Entscheidbarkeit

10 Punkte

Betrachten Sie die Sprache aller Kodierungen von Turing-Maschinen, die das NP-vollständige Problem "Hamiltonian Cycle" (HC) lösen.

$$\mathcal{L} = \left\{ w \in \{0,1\}^* \mid \mathcal{L}(M_w) = \mathrm{HC} \right\}.$$

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

Hinweis: Sie dürfen ein zweites Band oder Pseudocode verwenden um die Reduktion bzw. das Ergebnis der Reduktion zu beschreiben.

5. NP-Vollständigkeit

$\boxed{4 + 6 = 10 \text{ Punkte}}$

Sei $G = (V, E)$ ein gerichteter Graph. Ein induzierter Subgraph von G ist ein Graph $G_s = (V_s, E_s)$ mit $V_s \subseteq V$ und $E_s = E \cap (V_s \times V_s)$, d.h. G_s enthält nur einen Teil der Knoten von G, aber behält alle Kanten zwischen diesen Knoten bei. Eine k-Partitionierung von G ist eine Zerlegung von G in k-viele disjunkte Subgraphen G_1 bis G_k, sodass $V = V_1 \cup \cdots \cup V_k$ und $V_i \cap V_j = \emptyset$ für alle $i \neq j$.

Betrachten Sie das folgende Problem.

Hamilton-Cycle Decomposition (HCD)

Gegeben: Ein gerichteter Graph $G = (V, E)$

Entscheide: Gibt es eine 3-Partitionierung von G, sodass G_1, G_2 und G_3 jeweils einen Hamiltonkreis haben?

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

a) „Membership“: HCD $\in$ NP.

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

6. Berechenbarkeit

$\boxed{6 + 2 + 2 = 10 \text{ Punkte}}$

Es sei $\Sigma = \{0, 1\}$. Wir sagen, dass ein Wort $x \in \Sigma^*$ **lexikographisch kleiner** ist als ein Wort $y \in \Sigma^*$, wenn $|x| \leqslant |y|$ und entweder sind x und y gleich oder an der ersten Stelle p, wo sie sich unterscheiden, ist $x_p < y_p$, wobei $0 < 1$ gilt.

Betrachten Sie die Funktion $cubeTest : \Sigma^* \times \mathbb{N} \to \Sigma^* \cup \{\text{success}\}$ definiert als

$$cubeTest(w, k) := \begin{cases} x, & \text{falls x das lexikographisch kleinste Wort ist, für das gilt:} \\ & |x| \leqslant k \text{ und } Time_w(x) > (|x| + 1)^3 \\ \text{success}, & \text{falls es solch ein Wort nicht gibt} \end{cases}$$

Hierbei ist $Time_w(x)$ die Anzahl der Schritte, die die TM M_w auf Eingabe x durchführt bevor sie hält. Falls w keine Turing-Maschine kodiert, ist $Time_w(x) = 0$ für alle Eingaben x.

a) Zeigen Sie, dass *cubeTest* berechenbar ist, indem Sie einen Algorithmus angeben (Pseudocode genügt).

b) Angenommen k sei unär kodiert. Geben Sie eine möglichst genaue Laufzeitabschätzung für ihren Algorithmus in O-Notation an.

c) Sei w die Kodierung einer Turingmaschine M, sodass $\forall k \in \mathbb{N} : cubeTest(w, k) = \text{success}$ gilt. Was können Sie dann über die Laufzeit von M aussagen?

7. Quiz

2 + 2 + 3 + 3 = 10 Punkte

Untersuchen Sie die folgenden Aussagen auf Korrektheit. Begründen Sie Ihre Antwort mit einem kurzen Beweis oder einem Gegenbeispiel.

a) Wenn $\mathcal{L}$ Teilmenge einer Sprache $\mathcal{M} \in \text{PSPACE}$ ist, dann ist $\mathcal{L}$ selber in PSPACE.

b) Sei $\mathcal{L}_1$ eine NP-schwere Sprache und sei $\mathcal{L}_2$ eine coNP-schwere Sprache, dann ist $\mathcal{L} = \mathcal{L}_1 \cap \mathcal{L}_2$ sowohl NP-schwer als auch coNP-schwer (jeweils bzgl. logspace-many-one-Reduktionen).

c) Es gibt eine semi-entscheidbare, aber unentscheidbare Sprache $\mathcal{L}$ mit $\mathcal{L} \leqslant \overline{\mathcal{L}}$ (d.h. $\mathcal{L}$ ist many-one-reduzierbar auf ihr Komplement).

d) Sei für jede reelle Zahl $r \in \mathbb{R}$ eine berechenbare Funktion $f_r : \Sigma^* \to \Gamma^*$ definiert. Dann muss es zwei reelle Zahlen $r \neq s$ geben, sodass $f_r = f_s$ gilt, d.h. $\forall w \in \Sigma^* : f_r(w) = f_s(w)$.

8. Offset DTMs

10 Punkte

Sei $k \in \mathbb{N}$. Eine k-Offset DTM ist wie eine übliche DTM definiert mit folgender Besonderheit. Sie hat einen Lesekopf und einen separaten Schreibkopf, wobei der Schreibekopf immer um k Stellen nach rechts verschoben ist im Vergleich zum Lesekopf. Es gibt zwei Arten von Transitionen:

- $\delta(q, s) = (q', s', D)$ bedeutet, dass die TM in Zustand q das Symbol s liest (unter dem Lesekopf) und dann in den Zustand q' geht, das Symbol unter dem Schreibkopf durch s' ersetzt und schließlich beide Köpfe simultan in Richtung $D \in \{N, L, R\}$ bewegt.
- $\delta(q, s) = (q', D)$ ist eine Transition wie oben, aber ohne dass eine Schreiboperation durchgeführt wird.

Wie üblich fängt die TM bei Eingabe w mit dem Lesekopf auf dem ersten Symbol von w an.

Zeigen Sie, dass für jedes $k \in \mathbb{N}$ die k-Offset DTMs **gleichmächtig** zu üblichen DTMs sind.

9. NP-hardness

10 Punkte

Sei $\mathcal{L}$ eine NP-schwere Sprache (bzgl. logspace-many-one-Reduktionen) und sei $\mathcal{M}$ eine endliche Sprache disjunkt von $\mathcal{L}$, d.h. $\mathcal{L} \cap \mathcal{M} = \emptyset$. Zeigen Sie, dass $\mathcal{L} \cup \mathcal{M}$ NP-schwer ist (bzgl. logspace-many-one-Reduktionen).

10. Berechenbare Bilder von Sprachen

10 Punkte

Sei $\mathcal{L} \subseteq \Sigma^*$ eine semi-entscheidbare Sprache und sei $f : \Sigma^* \to \Gamma^*$ eine berechenbare Funktion. Wir definieren die Sprache $f(\mathcal{L}) = \{f(w) \mid w \in \mathcal{L}\} \subseteq \Gamma^*$.

Zeigen Sie, dass $f(\mathcal{L})$ semi-entscheidbar ist.