Benotete Hausaufgaben zur Vorlesung **WiSe 2023/2024**
Berechenbarkeit und Komplexität **TU Berlin**
Weller/Froese/Kellerhals/Kunz **11.03.2024**

Hausaufgabenblatt (Wiederholung)

Hinweise:

- Die Hausaufgabe kann ab dem **25.03.2024, 12:00 Uhr** bis zum **02.04.2024, 23:59 Uhr** auf ISIS hochgeladen werden.
- Die Hausaufgabe sollte möglichst in Dreiergruppen bearbeitet werden. Bitte tragen Sie sich in ISIS bis zum **25.03.2024, 11:00 Uhr** in der Gruppenwahl ein. Die Hausaufgabe kann nur von eingetragenen Gruppen abgegeben werden.
- Bitte verwenden Sie die LaTeX-Vorlage auf ISIS für Ihre Abgabe.
- Plagiate werden nicht toleriert und werden scharf geahndet.
- Es können bis zu **25 Portfoliopunkte** erreicht werden.
- Alle Antworten sind zu begründen. Antworten ohne Begründung erhalten **0 Punkte**. Einzige Einschränkungen:
  - Um zu zeigen, dass eine Funktion (Sprache) von einer Turing-Maschine berechnet (akzeptiert) werden kann, reicht es aus, das Verhalten der Maschine algorithmisch zu beschreiben. Das Gleiche gilt für WHILE- und GOTO-Programme.
  - Sätze, die in der Vorlesung oder Modulkonferenz *bewiesen* wurden (auch skizzenhaft) dürfen verwendet werden, aber unbewiesene Mitteilungen und Lösungen zu Tutoriumsaufgaben dürfen nicht verwendet werden (bzw. Beweis muss erbracht werden).
  - Sie können die Existenz einer universellen Turing-Maschine (eine Maschine, die bei Eingabe $w\#x$ die Maschine M_w auf Eingabe x simuliert) annehmen.
  - Sie können verwenden, dass das allgemeine Halteproblem H (Definition siehe unten) semi-entscheidbar ist.
- Wir behalten uns vor, pro Aufgabe mit x erreichbaren Punkten nicht mehr als $x/2$ Seiten zu lesen.

Erinnerungen:

- Alle in den Aufgaben vorkommenden Turing-Maschinen sind deterministisch.
- Σ ist ein beliebiges, endliches Alphabet. Das Symbol $\#$ ist ein Trennzeichen.
- Für jede Sprache $L \subseteq \Sigma^*$ ist $\overline{L} := \Sigma^* \setminus L$ ihr Komplement.
- Das allgemeine Halteproblem ist $H := \{w\#x \mid w, x \in \{0,1\}^* \text{ und } M_w \text{ hält bei Eingabe } x\}$.
- Das spezielle Halteproblem ist $K := \{w \in \{0,1\}^* \mid w\#w \in H\}$.
- Das Halteproblem auf leerem Band ist $H_0 := \{w \in \{0,1\}^* \mid w\# \in H\}$.

Aufgabe 1. *(Semi-)Entscheidbarkeit von Sprachen* 10 P.

Zeigen oder widerlegen Sie für jede der folgenden Sprachen jeweils Semi-Entscheidbarkeit und Entscheidbarkeit.

- $A := \{w \in \{0,1\}^* \mid M_w \text{ hält auf genau einem Eingabewort nicht}\}$
- $B := \{w\#x \mid w, x \in \{0,1\}^* \text{ und bei Eingabe } x \text{ besucht } M_w \text{ den Startzustand mehrmals}\}$
- $C := \{w\#x \mid w, x \in \{0,1\}^* \text{ und bei Eingabe } x \text{ besucht } M_w \text{ jeden ihrer Zustände genau einmal, bevor sie einen Zustand zum zweiten Mal besucht}\}$

Hinweis: Eine Turing-Maschine $M = (Z, \Sigma, \Gamma, \delta, z_0, \Box, E)$ *besucht* einen Zustand $z \in Z$ bei Eingabe x, falls es $k \in \mathbb{N}$, sowie $a, b \in \Gamma^*$ gibt, sodass $z_0 x \vdash_M^k azb$.

Lösungsskizze:

- A ist nicht semi-entscheidbar und damit nicht entscheidbar

  Beweis. Laut Vorlesung ist das Halteproblem auf leerem Band, H_0 nicht entscheidbar, laut Deckblatt ist H semi-entscheidbar und laut Vorlesung ist $H \leq H_0$, also ist auch H_0 semi-entscheidbar. Ergo ist $\overline{H}$ nicht semi-entscheidbar.

  Wir zeigen die nicht-semi-Entscheidbarkeit von A durch Reduktion von $\overline{H_0}$ auf A, mittels Reduktionsfunktion f, die bei Eingabe eines Wortes w die Kodierung einer Maschine M ausgibt, die bei Eingabe $x \in \{0,1\}^*$ wie folgt agiert:

  1. falls $x = \epsilon$, so gehe in eine Endlosschleife
  2. falls $x \neq \epsilon$, so simuliere M_w auf dem leeren Band für $BIN^{-1}(x)$ Schritte. Hält M_w in dieser Zeit, so gehe in eine Endlosschleife, sonst halte.

  f ist offenbar total und berechenbar. Wir beweisen die Reduktionseigenschaft von f, das heißt $w \in \overline{H_0} \iff f(w) \in A$.

  $$\begin{aligned} M_w \text{ hält nicht auf } \epsilon &\iff \forall_{k \in \mathbb{N}}\ M_w \text{ hält nicht (auf } \epsilon\text{) nach } k \text{ Schritten} \\ &\iff \forall_{x \in \{0,1\}^+}\ M_w \text{ hält nicht nach } BIN^{-1}(x) \text{ Schritten} \\ &\iff \forall_{x \in \{0,1\}^+}\ M \text{ hält auf } x \end{aligned}$$

  und, da M per Konstruktion auf ϵ nicht hält,

  $$\iff \langle M \rangle \in A \iff f(w) \in A \qquad \Box$$

- B ist semi-entscheidbar, aber nicht entscheidbar

  Beweis. Eine TM, die B semi-entscheidet simuliert M_w auf x und akzeptiert, falls der Startzustand ein zweites Mal besucht wird.

  Wir zeigen die Unentscheidbarkeit von B durch Reduktion von H_0, das nach Vorlesung unentscheidbar ist, mittels Reduktionsfunktion f, die bei Einfabe w die Kodierung der folgenden Modifikation von M_w, zusammen mit $x = \epsilon$, ausgibt:

  1. modifiziere M_w, so zu M', dass M' keinen akzeptierenden Endzustand hat (es geht nur um's Halten), und der Startzustand keine eingehenden Übergänge hat,
  2. jeder Übergang, der nicht definiert ist, schreibt ein neues Symbol q, das nicht im Arbeitsalphabet von M' vorkommt, auf das Band, bewegt den Kopf nicht, und geht in den Startzustand über (daraufhin hält die Maschine, da kein Übergang beim Lesen von q definiert ist)

  Wir beobachten Folgendes per Konstruktion:

  (a) M' besucht den Startzustand nie mehrmals

(b) M' hält genau dann wenn M_w hält

(c) die von $f(w)$ kodierte Maschine M besucht den Startzustand genau dann mehrmals, wenn M' hält

$$\begin{aligned} w \in H_0 &\iff M_w \text{ hält auf } \epsilon \\ &\overset{(b),(c)}{\iff} M \text{ besucht ihren Startzustand mehrfach bei Einfabe } \epsilon \\ &\iff f(w)\#\epsilon \in B. \end{aligned}$$

□

- C ist entscheidbar

Beweis. Bei Eingabe $w\#x$ müssen wir M_w nur für $|Z|$ Schritte auf x simulieren, wobei Z die Zustandsmenge von M_w ist. Besucht M_w in dieser Zeit einen Zustand mehr als einmal lehnen wir $w\#x$ ab, sonst akzeptieren wir. □

Aufgabe 2. *Berechenbarkeit* 9 P.

(a) Zeigen Sie, dass folgende Sprache unentscheidbar ist.

$$L \coloneqq \{w \in \{0,1\}^* \mid T(M_w) = \emptyset \text{ oder } T(M_w) = \{0,1\}^*\}$$

(b) Zeigen oder widerlegen Sie die Berechenbarkeit der Funktion $f \colon \{0,1\}^* \to \mathbb{N}$ mit

$$f(w) \coloneqq \min\{n \in \mathbb{N} \mid \text{Es existiert eine Turing-Maschine } M \text{ mit } n \text{ Zuständen, sodass } T(M) = T(M_w)\}.$$

Hinweis: Sie dürfen hierfür (a) verwenden.

Lösungsskizze:

(a) Wir verwenden den Satz von Rice für Sprachen. Sei dazu $S := \{\emptyset, \{0,1\}^*\}$. Offenbar sind $\emptyset$ und $\{0,1\}^*$ beide semi-entscheidbar und S ist nicht trivial . Weiterhin ist $\mathcal{C}(S) = \{w \mid T(M_w) \in S\} = \{w \mid T(M_w) = \emptyset \vee T(M_w) = \{0,1\}^*\} = L$.

(b) *Beweis.* Wäre f berechenbar, so wäre auch χ_L berechenbar, da $\chi_L(w) = 1 \iff T(M_w) \in \{\emptyset, \{0,1\}^*\} \iff f(w) = 1$. Dazu zeigen wir die zweite Äquivalenz:

„$\Rightarrow$“: Zunächst gilt $f(w) \neq 0$ da jede Maschine einen Startzustand haben muss. Mit einem Zustand kann man sowohl $\emptyset$ (Startzustand nicht akzeptierend) als auch $\{0,1\}^*$ (Startzustand akzeptierend) akzeptieren.

„$\Leftarrow$“: Sei $f(w) = 1$. Nach Definition gibt es eine Maschine M mit einem einzigen Zustand z und $T(M) = T(M_w)$. Ist z akzeptierend, so hat z nach Definition keine ausgehenden Übergänge, d.h. M_w akzeptiert jedes Eingabewort, also $T(M_w) = \{0,1\}^*$. Ist z nicht akzeptierend, so hat M_w keine akzeptierenden Zustände, also $T(M_w) = \emptyset$. □

Aufgabe 3. *Postsches Korrespondenzproblem* 6 P.

Zeigen oder widerlegen Sie die Entscheidbarkeit der beiden folgenden Sprachen:

$$\begin{aligned} P_{\geq} \coloneqq \{\langle((x_1,y_1),\dots,(x_k,y_k))\rangle \mid\ & k \geq 1, x_i, y_i \in \Sigma^* \text{ für alle } i \in \{1,\dots,k\}, \\ & \text{und es existieren } n \geq k \text{ und } i_1,\dots,i_n \in \{1,\dots,k\}, \\ & \text{sodass } x_{i_1} \cdot \ldots \cdot x_{i_n} = y_{i_1} \cdot \ldots \cdot y_{i_n}\} \end{aligned}$$

$$P_{\leq} \coloneqq \{\langle((x_1, y_1), \ldots, (x_k, y_k))\rangle \mid k \geq 1, x_i, y_i \in \Sigma^* \text{ für alle } i \in \{1, \ldots, k\},$$
$$\text{und es existieren } n \leq \mathrm{ack}(k, k) \text{ und } i_1, \ldots, i_n \in \{1, \ldots, k\},$$
$$\text{sodass } x_{i_1} \cdot \ldots \cdot x_{i_n} = y_{i_1} \cdot \ldots \cdot y_{i_n}\}$$

Hinweis: ack bezeichnet hierbei die Ackermannfunktion (siehe Vorlesung).

Lösungsskizze:

$P_{\geq}$ $P_{\geq}$ ist unentscheidbar . Wir reduzieren das PCP auf $P_{\geq}$ mithilfe der folgenden Reduktionsfunktion f : Bei Eingabe einer (kodierten) PCP-Instanz $I = ((x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k))$,

1. erzeuge jede der k^k kombinationen der k Eingabedominos und teste, ob eine davon eine Lösung für die eingegebene PCP-Instanz ist
2. ist ein Test erfolgreich, gib eine triviale Ja-Instanz von $P_{\geq}$ aus (zB. $((0, 0))$), sonst gib I aus

f ist offenbar berechenbar. Wir zeigen, dass $I \in \mathrm{PCP} \iff f(I) \in P_{\geq}$.

Beweis. „$\Rightarrow$“ : Sei $I \in \mathrm{PCP}$. Dann gibt es eine kürzeste Lösung $(i_1, i_2, \ldots, i_n)$. Ist $n \leq k$, so wird diese Lösung von der Reduktionsfunktion in Schritt 1 gefunden und es gilt $f(I) \in P_{\geq}$. Ist $n > k$, so ist $f(I) = I$ und $I \in P_{\geq}$ nach Definition von $P_{\geq}$.

„$\Leftarrow$“ : Sei $I \in P_{\geq}$. Dann ist offenbar $I \in \mathrm{PCP}$ nach Definition von $P_{\geq}$. □

$P_{\leq}$ Eine Maschine, die $P_{\leq}$ entscheidet testet alle $k^{\mathrm{ack}(k,k)}$ vielen Kombinationen von Dominos.
