

Lösungsvorschlag zur Übung 11 zur Vorlesung  
Theoretische Informatik für Medieninformatiker

TIMI11-1 *Entscheidbarkeit und Reduktionen*

(2 Punkte)

a) Betrachten Sie die Sprache

$$L_1 = \{z \in \{0,1\}^* \mid \text{TM } M_z \text{ hält für Eingabe } 01\}$$

Welche der folgenden Aussagen ist korrekt? Beweisen Sie Ihre Antwort.

- $L_1$  ist entscheidbar.
- $L_1$  ist semi-entscheidbar, aber nicht entscheidbar.
- $L_1$  ist weder entscheidbar noch semi-entscheidbar.

Um zu zeigen, dass  $L_1$  (semi-)entscheidbar ist, beschreiben Sie kurz die Funktionsweise einer deterministischen Turingmaschine, die  $L_1$  (semi-)entscheidet. Um zu zeigen, dass  $L_1$  nicht (semi-)entscheidbar ist, reduzieren Sie ein geeignetes Problem auf  $L_1$ .

**LÖSUNGSVORSCHLAG:**

$L_1$  ist semi-entscheidbar: für eine Eingabe  $z$  simulieren wir die Maschine  $M_z$  mit Eingabe 01. Wenn die Simulation hält, geben wir 1 aus.

$L_1$  ist nicht entscheidbar. Wir beweisen das durch Reduktion von  $H_0$  auf  $L_1$ .

Um  $H_0 \leq L_1$  zu zeigen, müssen wir eine totale, berechenbare Funktion  $f: \{0,1\}^* \rightarrow \{0,1\}^*$  angeben, sodass für alle  $z$  gilt:  $z \in H_0 \iff f(z) \in L_1$ . Wir wählen

$$f(z) = \begin{cases} \langle M' \rangle & \text{falls } z = \langle M \rangle \\ z & \text{sonst} \end{cases}$$

Dabei ist

- $M'$  eine Turingmaschine, die zunächst ihre Eingabe löscht und dann  $M$  ausführt.
- $\langle M \rangle$  und  $\langle M' \rangle$  die Binärkodierung der jeweiligen Turingmaschine.

Die Funktion  $f$  ist offensichtlich total. Sie ist auch berechenbar, denn:

- Wir können Turingmaschinen binär kodieren und dekodieren (und erkennen, ob ein binäres Wort eine Turingmaschine kodiert).
- Wir können  $M'$  aus  $M$  konstruieren.

Für  $M'$  gilt:

- Wenn  $M$  mit Eingabe  $\varepsilon$  hält, dann hält  $M'$  mit jeder beliebigen Eingabe und also insbesondere mit Eingabe  $01$ .
- $M'$  hält mit einer beliebigen Eingabe, und also insbesondere mit Eingabe  $01$ , nur dann, wenn  $M$  mit Eingabe  $\varepsilon$  hält.

Somit gilt:

$$\begin{aligned} & z \in H_0 \\ \text{g.d.w. } & z = \langle M \rangle \text{ und } M \text{ hält mit Eingabe } \varepsilon \\ \text{g.d.w. } & f(z) = \langle M' \rangle \text{ und } M' \text{ hält mit Eingabe } 01 \\ \text{g.d.w. } & f(z) \in L_1 \end{aligned}$$

Somit ist  $H_0 \leq L_1$  und da  $H_0$  unentscheidbar ist, ist auch  $L_1$  unentscheidbar.

- b) Prüfen Sie, ob die folgenden Behauptungen wahr oder falsch sind. Begründen Sie Ihre Antworten wie folgt: Wenn eine Sprache  $L$  (semi-)entscheidbar ist, beschreiben Sie die Funktionsweise einer deterministischen Turingmaschine, die die charakteristische Funktion  $\chi_L$  bzw.  $\chi'_L$  berechnet. Wenn  $L$  nicht (semi-)entscheidbar ist, leiten Sie einen Widerspruch ab.
- i) Wenn  $A$  und  $B$  entscheidbare Sprachen sind, dann ist  $A \cap B$  entscheidbar.

#### LÖSUNGSVORSCHLAG:

Wahr. Sei  $T_A$  eine DTM, die  $A$  entscheidet, und  $T_B$  eine DTM, die  $B$  entscheidet. Konstruiere eine DTM für  $A \cap B$ , die sich wie folgt verhält: Gegeben  $x$ , berechne  $T_A(x)$ . Falls  $T_A(x) = 0$ , lehne  $x$  ab, ansonsten berechne  $T_B(x)$ . Gilt  $T_B(x) = 0$ , lehne  $x$  ab, sonst akzeptiere  $x$ . Da  $T_A, T_B$  die jeweiligen Mengen entscheiden, terminieren beide DTM immer, womit auch die DTM zu  $A \cap B$  stets mit dem korrekten Ergebnis terminiert. Somit ist  $A \cap B$  entscheidbar.

- ii) Wenn  $A$  und  $A \cup B$  entscheidbar sind, dann ist  $B$  entscheidbar.

#### LÖSUNGSVORSCHLAG:

Falsch. Sei  $A = \{0, 1\}^*$  und  $B = H_0 \subseteq \{0, 1\}^*$ . Dann sind  $A$  und  $A \cup B = A$  entscheidbar, aber  $B$  nicht.

- iii) Das Problem, ob  $L(M) \neq \emptyset$  für eine gegebene Turingmaschine  $M$  gilt, ist semi-entscheidbar.

**LÖSUNGSVORSCHLAG:**

Wahr. Algorithmus: Für  $i = 0, 1, \dots$  schreibe nacheinander alle Wörter  $w \in \Sigma^*$  mit  $|w| \leq i$  auf das Band und simulierte  $M$  auf jedem  $w$  für  $i$  Schritte. Falls  $M$  in  $i$  Schritten akzeptiert, gib 1 aus. Ist  $L(M) \neq \emptyset$ , so testen wir  $M$  irgendwann für ausreichend viele Schritte auf Wörtern ausreichender Länge, um ein Wort aus  $L(M)$  zu finden.

- iv) Das Problem, ob  $L(M) = \emptyset$  für eine gegebene Turingmaschine  $M$  gilt, ist semi-entscheidbar.

**LÖSUNGSVORSCHLAG:**

Falsch. Wäre dieses Problem semi-entscheidbar, so wäre zusammen mit der vorigen Teilaufgabe entscheidbar, ob  $L(M) \neq \emptyset$  ist. Sei  $P$  die Menge der Turingmaschinen  $M$  (binär kodiert), für die  $L(M) \neq \emptyset$  ist. Wir zeigen:  $H_0 \leq P$  und somit wäre  $H_0$  entscheidbar, wenn  $P$  entscheidbar wäre.

Um  $H_0 \leq P$  zu zeigen, definieren wir die Funktion  $f$ , die die (binär kodierte) Turingmaschine  $M$  auf eine (binär kodierte) Turingmaschine  $M'$  abbildet. Dabei verhält sich  $M'$  wie folgt: Ist die Eingabe nicht das leere Wort, so akzeptiert  $M'$  nicht. Ist die Eingabe das leere Wort, so simuliert  $M'$  zunächst  $M$  auf dem leeren Wort und akzeptiert dann (sofern  $M$  gehalten hat).  $M'$  akzeptiert also höchstens das leere Wort, und das genau dann wenn  $M$  das leere Wort akzeptiert. Die Funktion ist offensichtlich total und berechenbar. Somit gilt:

$$\begin{aligned} & M \in H_0 \\ \text{g.d.w.} & \quad M \text{ hält auf } \varepsilon \\ \text{g.d.w.} & \quad L(f(M)) \neq \emptyset \\ \text{g.d.w.} & \quad f(M) \in P \end{aligned}$$

- c) Zeigen Sie, dass das folgende Problem für jede Turingmaschine  $T$  und natürliche Zahl  $n$  entscheidbar ist.

„ $T$  hält auf jeder Eingabe nach höchstens  $n$  Schritten.“

### LÖSUNGSVORSCHLAG:

Für jede Eingabe  $w$  können wir prüfen, ob  $T$  auf  $w$  in  $n$  Schritten hält, indem wir  $T$  für  $n$  Schritte simulieren. Weiterhin kann  $T$  in  $n$  Schritten nur höchstens  $n$  Zeichen lesen. Deshalb gilt für Wörter  $w$  mit  $|w| > n$ : Wenn  $T$  nicht in  $n$  Schritten auf  $w[0] \dots w[n]$  hält, dann hält  $T$  auch nicht in  $n$  Schritten auf  $w$ . Somit genügt es, nur Wörter mit Länge höchstens  $n$  zu testen. Da das endlich viele Wörter sind, ist das Problem entscheidbar.

### TIMI11-2 Satz von Rice

(0 Punkte)

Sei  $T = (Z, \{a, b\}, \Gamma, \delta, z_0, \square, E)$  eine Turingmaschine. Welche der folgenden Fragestellungen zu  $T$  sind entscheidbar?

Zeigen Sie die Unentscheidbarkeit unentscheidbarer Fragestellungen mittels des Satzes von Rice. Bei entscheidbaren Fragestellungen erläutern Sie, wie die charakteristische Funktion berechnet wird.

### LÖSUNGSVORSCHLAG:

Wiederholung:

**Satz von Rice:** Sei  $\mathcal{R}$  die Klasse aller Turingberechenbaren Funktionen. Sei  $\mathcal{S}$  eine beliebige Teilmenge, sodass  $\emptyset \subset \mathcal{S} \subset \mathcal{R}$ . Dann ist die Sprache

$$C(\mathcal{S}) = \{w \mid \text{die von } M_w \text{ berechnete Funktion liegt in } \mathcal{S}\}$$

unentscheidbar.

- a)  $T$  hat mindestens  $|\Gamma|!$  viele Zustände. (! bezeichnet die Fakultätsfunktion.)

**LÖSUNGSVORSCHLAG:** Das Problem ist entscheidbar, indem man aus der Turingmaschinenbeschreibung zunächst  $|\Gamma|$  und  $|Z|$  extrahiert, anschließend  $|\Gamma|!$  berechnet und anschließend die Zahlen vergleicht. All das ist von einer Turingmaschine berechenbar.

- b) Leerheitsproblem: Es gibt ein Wort  $w \in L(T)$

**LÖSUNGSVORSCHLAG:** Das Problem ist unentscheidbar:

Sei  $\mathcal{S} = \mathcal{R} \setminus \{\Omega\}$ , also die Menge aller berechenbaren Funktionen, bis auf die überall undefinierte Funktion  $\Omega$ .

$\mathcal{S}$  ist damit nicht leer und auch nicht die Menge aller berechenbaren Funktionen.

Dann ist  $C(\mathcal{S}) = \{w_M \mid M \text{ akzeptiert mindestens für eine Eingabe}\}$

Mit dem Satz von Rice ist  $C(\mathcal{S})$  unentscheidbar und damit ist das Leerheitsproblem (also ob eine Turingmaschine mindestens ein Wort akzeptiert) unentscheidbar.

- c) Wenn  $a \in L(T)$ , dann auch  $aa \in L(T)$

**LÖSUNGSVORSCHLAG:** Das Problem ist unentscheidbar:

Sei  $\mathcal{S} = \{f \in \mathcal{R} \mid f(a) \text{ definiert} \Rightarrow f(aa) \text{ definiert}\}$ . Dann sind totale (überall definierte) Funktionen wie  $f: \{a,b\}^* \rightarrow \{a,b\}^*, w \mapsto a$  in  $\mathcal{S}$  und die folgende (partielle) Funktion  $g$  ist nicht in  $\mathcal{S}$ :

$$g: \{a,b\}^* \rightarrow \{a,b\}^*, w \mapsto \begin{cases} a & \text{falls } w = a \\ \text{undefiniert} & \text{sonst} \end{cases}$$

Damit ist  $\mathcal{S}$  nicht leer und auch nicht die Menge aller berechenbaren Funktionen.

Mit dem Satz von Rice ist  $C(\mathcal{S}) = \{w_M \mid M \text{ akzeptiert } a \Rightarrow M \text{ akzeptiert } aa\}$  unentscheidbar.

- d) Wenn es ein Wort  $w \in L(T)$  gibt, dann gibt es auch ein Wort  $u$ , welches ein  $b$  enthält mit  $u \in L(T)$

**LÖSUNGSVORSCHLAG:** Das Problem ist unentscheidbar:

Sei  $\mathcal{S} = \{f \in \mathcal{R} \mid \exists w : f(w) \text{ definiert} \Rightarrow \exists u, u' \in \{a,b\}^* : f(ubu') \text{ definiert}\}$ . Dann sind alle totalen Funktionen in  $\mathcal{S}$  und die Funktion  $g$  aus c) ist wieder nicht in  $\mathcal{S}$ . Damit ist  $\mathcal{S}$  nicht leer und auch nicht die Menge aller berechenbaren Funktionen.

Mit dem Satz von Rice ist  $C(\mathcal{S}) = \{w_M \mid M \text{ akzeptiert eine Eingabe} \Rightarrow M \text{ akzeptiert eine Eingabe, die } b \text{ enthält}\}$  unentscheidbar.