ETH zürich
Departement Informatik

Theoretische Informatik

Prof. Dr. Juraj Hromkovič
Dr. Hans-Joachim Böckenhauer
https://courses.ite.inf.ethz.ch/theo_inf_21

Lösungsvorschläge – Blatt 6

Zürich, 5. November 2021

Lösung zu Aufgabe 16

(a) Wir nehmen an, dass die Gäste wie folgt durchnummeriert sind: Sei $G_{0,i}$ der bereits im Hotel wohnende Gast im Zimmer Z_i, für $i \in \mathbb{N}$, sei $G_{j,i}$ der i-te Gast im Bus j, für $j \in \{1, 2, 3\}$ und $i \in \mathbb{N}$.

Um alle Wünsche der Gäste zu berücksichtigen, kann der Portier zum Beispiel die folgende Aufteilung wählen. Er teilt die Folge der Zimmer in Blöcke der Grösse 6 auf und vergibt die Zimmer mit Nummern 0 mod 6 an die bereits anwesenden Gäste, die Zimmer mit Nummern 1 mod 6 an die Gäste aus dem ersten Bus, die Zimmer mit Nummern 2 mod 6 und 3 mod 6 an die Gäste aus dem zweiten Bus und die Zimmer mit Nummern 4 mod 6 und 5 mod 6 an die Gäste aus dem dritten Bus.

Formal lässt sich diese Zimmerbelegung für alle $i \in \mathbb{N}$ durch eine Funktion f angeben mit

$$
\begin{array}{ll}
f(G_{0,i}) = Z_{6i}, & \\
f(G_{1,i}) = Z_{6i+1}, & \\
f(G_{2,i}) = Z_{6\cdot\frac{i}{2}+2} = Z_{3i+2}, & \text{falls } i \text{ gerade,} \\
f(G_{2,i}) = Z_{6\cdot\frac{i-1}{2}+3} = Z_{3i}, & \text{falls } i \text{ ungerade,} \\
f(G_{3,i}) = Z_{6\cdot\frac{i}{2}+4} = Z_{3i+4}, & \text{falls } i \text{ gerade,} \\
f(G_{3,i}) = Z_{6\cdot\frac{i-1}{2}+5} = Z_{3i+2}, & \text{falls } i \text{ ungerade.}
\end{array}
$$

(b) Der Portier teilt zunächst die Zimmer in unendlich viele Gruppen unendlicher Grösse ein. Dann kann er für jede eintreffende Gruppe von Gästen (unabhängig davon, ob diese endlich oder unendlich ist) die nächste noch nicht belegte Zimmergruppe verwenden.

Um die Zimmergruppen zu bestimmen, verwenden wir die folgende Idee: Wir wissen, dass es unendlich viele Primzahlen gibt. Sei p_i die i-te Primzahl in aufsteigender Reihenfolge. Dann können wir die i-te Gruppe von Zimmernummern definieren als $\text{Gruppe}(i) = \{p_i^j \mid j \in \mathbb{N}^+\}$. Hierbei wird kein Zimmer mehreren Gruppen zugeordnet, weil jede Zahl eine eindeutige Primfaktorzerlegung besitzt.

Lösung zu Aufgabe 17

Wir zeigen zunächst mit Hilfe eines indirekten Beweises, dass L_2 nicht rekursiv aufzählbar ist. Angenommen, L_2 sei rekursiv aufzählbar. Dann gibt es eine Turingmaschine M mit

$L(M) = L_2$. Weil M eine der Turingmaschinen in der kanonischen Aufzählung aller Turingmaschinen sein muss, existiert ein $j \in \mathbb{N} - \{0\}$, so dass $M_j = M$. Wir betrachten nun das Wort $w_i = w_{2j}$. Dann gilt

$$\begin{aligned} w_i = w_{2j} \in L_2 &\iff M_{\lceil i/2 \rceil} = M_j \text{ akzeptiert } w_{2j} \text{ nicht} \quad \text{(nach Definition von } L_2\text{)} \\ &\iff M \text{ akzeptiert } w_{2j} \text{ nicht} \quad \text{(nach Definition von } M\text{)} \\ &\iff w_{2j} \notin L(M) \\ &\iff w_{2j} = w_i \notin L_2. \end{aligned}$$

Dies ist ein Widerspruch, also war die Annahme falsch und L_2 ist nicht rekursiv aufzählbar.

Für die Sprache L_1 kann man keinen analogen Beweis führen wie für L_2. Der Beweis für L_2 basiert darauf, dass in der Liste jede Turingmaschine erfasst ist und daher auch die angenommene Maschine für die betrachtete Sprache. Dies gilt zwar für L_1 auch, führt aber nicht zum Widerspruch, da diese Maschine auch eine ungerade Nummer haben kann, also nicht von der Form $2i$ für ein $i \in \mathbb{N} - \{0\}$ ist.

Man bemerke, dass man aus dem Fehlschlagen dieses Beweisansatzes nicht folgern kann, dass L_1 rekursiv aufzählbar wäre.

Lösung zu Aufgabe 18

(a) Wir können unser neues Modell beispielsweise wie folgt definieren. Eine Turingmaschine des neuen Modells ist ein 7-Tupel $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$, wobei

  (i) $Q = \{q_0, q_{\text{accept}}, q_{\text{reject}}\}$,
  (ii) Σ ein Alphabet mit $¢, \sqcup \notin \Sigma$ ist,
  (iii) Γ ein Alphabet mit $¢, \sqcup \in \Gamma$, $\Sigma \subseteq \Gamma$ und $\Gamma \cap Q = \emptyset$ ist,
  (iv) $\delta\colon \{q_0\} \times \Gamma^3 \to Q \times \Gamma^3 \times \{\mathrm{L}, \mathrm{R}, \mathrm{N}\}$ mit der Eigenschaft

$$\forall s, s' \in \Gamma\colon\ \exists q \in Q, \tilde{s}, \tilde{s}' \in \Gamma, d \in \{\mathrm{R}, \mathrm{N}\}\colon \qquad \delta(q_0, (¢, s, s')) = (q, (¢, \tilde{s}, \tilde{s}'), d).$$

Wir können die Definition einer Konfiguration unverändert vom Standardmodell übernehmen. Die Definition eines Berechnungsschrittes muss hingegen an die neue Übergangsfunktion δ angepasst werden. Dies ist wie folgt möglich:

  (i) $x_1 \ldots x_{i-1} q x_i \ldots x_n \underset{M}{\vdash} x_1 \ldots x_{i-2} y_{\mathrm{L}} p y y_{\mathrm{R}} x_{i+2} \ldots x_n$,
  falls $\delta(q_0, (x_{i-1}, x_i, x_{i+1})) = (p, (y_{\mathrm{L}}, y, y_{\mathrm{R}}), \mathrm{N})$
  (ii) $x_1 \ldots x_{i-1} q x_i \ldots x_n \underset{M}{\vdash} x_1 \ldots x_{i-2} p y_{\mathrm{L}} y y_{\mathrm{R}} x_{i+2} \ldots x_n$,
  falls $\delta(q_0, (x_{i-1}, x_i, x_{i+1})) = (p, (y_{\mathrm{L}}, y, y_{\mathrm{R}}), \mathrm{L})$
  (iii) $x_1 \ldots x_{i-1} q x_i \ldots x_n \underset{M}{\vdash} x_1 \ldots x_{i-2} y_{\mathrm{L}} y p y_{\mathrm{R}} x_{i+2} \ldots x_n$,
  falls $\delta(q_0, (x_{i-1}, x_i, x_{i+1})) = (p, (y_{\mathrm{L}}, y, y_{\mathrm{R}}), \mathrm{R})$, für $i < n - 1$ und
  (iv) $x_1 \ldots x_{n-2} q x_{n-1} x_n \underset{M}{\vdash} x_1 \ldots x_{n-2} y_{\mathrm{L}} y p y_{\mathrm{R}} \sqcup$,
  falls $\delta(q_0, (x_{n-2}, x_{n-1}, x_n)) = (p, (y_{\mathrm{L}}, y, y_{\mathrm{R}}), \mathrm{R})$.

Wir gehen hierbei davon aus, dass der Lese-/Schreibkopf zu Beginn nicht auf dem Feld mit Symbol ¢ steht, sondern auf dem Feld rechts davon, die Startkonfiguration für ein Eingabewort x ist also $¢q_0x$ falls $|x| \geq 2$, $¢q_0x\sqcup$ falls $|x| = 1$ und $¢q_0\sqcup\sqcup$ falls $|x| = 0$. Die verbleibenden Definitionen – insbesondere der Begriff einer Berechnung und der Begriff der von einer Turingmaschine akzeptierten Sprache – lassen sich nun unverändert übernehmen.

(b) Sei eine Turingmaschine $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ des Standard-Modells vorgegeben. Wir geben eine äquivalente Turingmaschine M' des neuen Modells an. Die Idee ist, dass wir das Alphabet von Γ um $\Gamma \times Q$ erweitern, was es ermöglicht, den aktuellen Zustand von M jeweils in einem Symbol abzuspeichern, das im nächsten Schritt wieder ausgelesen werden kann.

Formal definieren wir

$$M' = (\{q_0, q_{\text{accept}}, q_{\text{reject}}\}, \Sigma, \Gamma \cup \Gamma \times Q, \delta', q_0, q_{\text{accept}}, q_{\text{reject}}),$$

wobei für alle $q, \tilde{q} \in Q,\ s, s_{\text{L}}, s_{\text{R}}, \tilde{s} \in \Gamma$ und $S \in \{\text{L}, \text{R}, \text{N}\}$

$$\delta'(q_0, \big(s_{\text{L}}, s, s_{\text{R}}\big)) = (q_0, \big(s_{\text{L}}, \begin{pmatrix} s \\ q_0 \end{pmatrix}, s_{\text{R}}\big), \text{N})$$

und, falls $\delta(q, s) = (\tilde{q}, \tilde{s}, S)$ mit $\tilde{q} \notin \{q_{\text{accept}}, q_{\text{reject}}\}$, dann

$$\begin{aligned}
\delta'(q_0, \big(s_{\text{L}}, \begin{pmatrix} s \\ q \end{pmatrix}, s_{\text{R}}\big)) &= (q_0, \big(\begin{pmatrix} s_{\text{L}} \\ \tilde{q} \end{pmatrix}, \tilde{s}, s_{\text{R}}\big), \text{L}), \text{ falls } S = \text{L}, \\
\delta'(q_0, \big(s_{\text{L}}, \begin{pmatrix} s \\ q \end{pmatrix}, s_{\text{R}}\big)) &= (q_0, \big(s_{\text{L}}, \tilde{s}, \begin{pmatrix} s_{\text{R}} \\ \tilde{q} \end{pmatrix}\big), \text{R}), \text{ falls } S = \text{R} \text{ und} \\
\delta'(q_0, \big(s_{\text{L}}, \begin{pmatrix} s \\ q \end{pmatrix}, s_{\text{R}}\big)) &= (q_0, \big(s_{\text{L}}, \begin{pmatrix} \tilde{s} \\ \tilde{q} \end{pmatrix}, s_{\text{R}}\big), \text{N}), \text{ falls } S = \text{N},
\end{aligned}$$

und, falls $\delta(q, s) = (\tilde{q}, \tilde{s}, S)$ mit $\tilde{q} \in \{q_{\text{accept}}, q_{\text{reject}}\}$, dann

$$\delta'(q_0, \big(s_{\text{L}}, \begin{pmatrix} s \\ q \end{pmatrix}, s_{\text{R}}\big)) = (\tilde{q}, \big(s_{\text{L}}, \tilde{s}, s_{\text{R}}\big), \text{S})$$

sei.