ETH zürich
Departement Informatik

Theoretische Informatik

Prof. Dr. Juraj Hromkovič
Dr. Hans-Joachim Böckenhauer
http://www.ita.inf.ethz.ch/theoInf18

Lösungsvorschläge – Blatt 7

Zürich, 9. November 2018

Lösung zu Aufgabe 20

(a) Um zu zeigen, dass $L_{\text{union}} \in \mathcal{L}_{\text{RE}}$ gilt, konstruieren wir eine 2-Band-Turingmaschine A für L_{union}. Dies reicht aus, da Mehrband-Turingmaschinen und Turingmaschinen nach Satz 4.1 im Buch äquivalent sind. Die Maschine A überprüft zunächst, ob die Eingabe eine gültige Form hat, also aus den Kodierungen von zwei Turingmaschinen M und M' und einem Wort w besteht. Falls dies nicht der Fall ist, dann verwirft A. Sonst verwendet A ihre zwei Arbeitsbänder, um die Arbeit von M und M' auf dem Wort w parallel zu simulieren.

Falls eine der beiden simulierten Maschinen das Wort w akzeptiert, dann akzeptiert auch A ihre Eingabe. Falls sowohl M als auch M' das Wort w verwerfen, verwirft auch A. Falls eine der beiden Maschinen M und M' unendlich lange läuft und die andere verwirft oder auch unendlich lange läuft, dann läuft auch A unendlich lange. Damit ist A eine Turingmaschine, die die Sprache L_{union} akzeptiert.

Man beachte, dass eine sequentielle Simulation der beiden Maschinen M und M' nicht möglich ist, da es sein kann, dass M auf w unendlich lange läuft, M' aber w akzeptiert.

(b) Um $L_{\text{U}} \leq_{\text{EE}} L_{\text{union}}$ zu zeigen, müssen wir zeigen, wie man alle Eingaben für L_{U} so algorithmisch in Eingaben für L_{union} umwandeln kann, dass genau die zu akzeptierenden Eingaben auf zu akzeptierende Eingaben abgebildet werden. Jedes Wort über $\{0, 1, \#\}$, das nicht die Form $\text{Kod}(M)\#w$ hat für eine Turingmaschine M, wird auf λ abgebildet, da $\lambda \notin L_{\text{union}}$. Jedes Wort der Form $\text{Kod}(M)\#w$ wird auf $\text{Kod}(M)\#\text{Kod}(M)\#w$ abgebildet und ist damit eine gültige Eingabe für L_{union}. Offenbar gilt

$$\text{Kod}(M)\#\text{Kod}(M)\#w \in L_{\text{union}} \iff \text{Kod}(M)\#w \in L_{\text{U}}\,.$$

Damit ist $L_{\text{U}} \leq_{\text{EE}} L_{\text{union}}$ gezeigt.

(c) Für den Beweis von $L_{\text{union}} \leq_{\text{EE}} L_{\text{U}}$ geben wir eine berechenbare Abbildung an, die gültige Eingaben für L_{union} in gültige Eingaben für L_{U} umwandelt. Ungültige Eingaben (also Eingaben, die nicht die Form $\text{Kod}(M)\#\text{Kod}(M')\#w$ haben) werden wieder auf $\lambda \notin L_{\text{U}}$ abgebildet. Jede Eingabe der Form $\text{Kod}(M)\#\text{Kod}(M')\#w$ wird abgebildet auf $\mathrm{Kod}(A)\#w$, wobei A eine Turingmaschine ist, die M und M' parallel simuliert wie im Aufgabenteil (a) beschrieben. Dann gilt

$$\begin{aligned}\mathrm{Kod}(M)\#\mathrm{Kod}(M')\#w \in L_{\mathrm{union}} &\iff w \in L(M) \cup L(M') \\ &\iff w \in L(A) \\ &\iff \mathrm{Kod}(A)\#w \in L_{\mathrm{U}}.\end{aligned}$$

Damit ist $L_{\mathrm{union}} \leq_{\mathrm{EE}} L_{\mathrm{U}}$ gezeigt.

Lösung zu Aufgabe 21

(a) Wir zeigen, dass $L_{\mathrm{H}} \leq_{\mathrm{EE}} L_{\mathrm{U}}$ gilt, daraus folgt dann sofort $L_{\mathrm{H}} \leq_{\mathrm{R}} L_{\mathrm{U}}$. Wir konstruieren also eine TM A, die eine Funktion $f\colon \{0,1,\#\}^* \to \{0,1,\#\}^*$ zur Umwandlung der Eingaben berechnet. Die TM A arbeitet auf einer Eingabe x nun wie folgt:

Falls x nicht von der Form $\mathrm{Kod}(M)\#w$ ist, so wird λ ausgegeben. Offenbar gilt $\lambda \notin L_{\mathrm{U}}$.

Sonst gilt $x = \mathrm{Kod}(M)\#w$. Dann konstruiert A aus M eine Turingmaschine M', bei der nur einige Transitionen geändert sind: Für die Transitionsfunktion δ' von M' gilt $\delta'(q,a) = q_{\mathrm{accept}}$, wenn $\delta(q,a) = q_{\mathrm{reject}}$ in M gilt. M hält genau dann, wenn sie einen der beiden Zustände q_{accept} oder q_{reject} erreicht. In beiden Fällen (und nur dann) erreicht M' den Zustand q_{accept}. Es gilt also

$$M \text{ hält auf } w \iff w \in L(M') \tag{1}$$

für alle $w \in \Sigma_{\mathrm{bool}}^*$. Die TM A gibt $f(x) = \mathrm{Kod}(M')\#w$ aus.

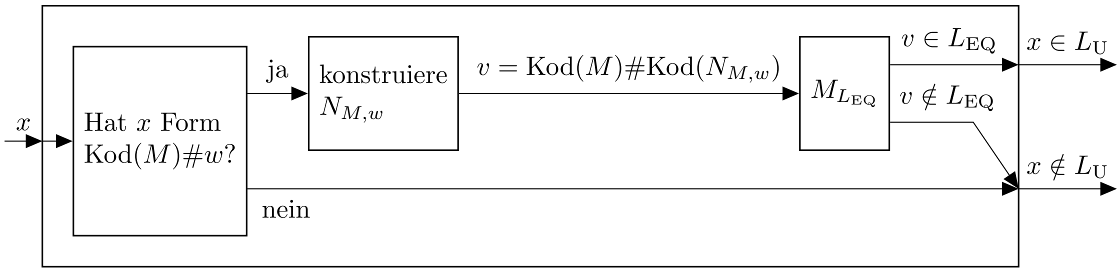
Wir zeigen nun $x \in L_{\mathrm{H}} \iff f(x) \in L_{\mathrm{U}}$.

Falls $x \in L_{\mathrm{H}}$, so ist $x = \mathrm{Kod}(M)\#w$, und M hält auf w. Nach (1) folgt $w \in L(M')$, also $f(\mathrm{Kod}(M)\#w) = \mathrm{Kod}(M')\#w \in L_{\mathrm{U}}$.

Falls $x \notin L_{\mathrm{H}}$, so gibt es zwei Fälle. Wenn x nicht von der Form $\mathrm{Kod}(M)\#w$ ist, so ist $f(x) = \lambda \notin L_{\mathrm{U}}$. Andernfalls ist $x = \mathrm{Kod}(M)\#w$, und M hält nicht auf w. Nach (1) folgt $w \notin L(M')$, also $f(\mathrm{Kod}(M)\#w) = \mathrm{Kod}(M')\#w \notin L_{\mathrm{U}}$.

(b) Die Aufgabe besteht darin, L_{U} zu entscheiden unter der Annahme, wir könnten L_{EQ} entscheiden. Wir wollen also wissen, ob eine Maschine ein bestimmtes Wort akzeptiert, können aber nur zwei Maschinen auf Äquivalenz testen. Also richten wir es so ein, dass wir zwei Maschinen erhalten, deren Sprachen sich nur hinsichtlich des einen Wortes unterscheiden können.

Das nachstehende Diagramm stellt eine Turingmaschine dar, die L_{U} entscheidet unter der Annahme, dass die darin verwendete Maschine $M_{L_{\mathrm{EQ}}}$ eine TM ist, welche L_{EQ} entscheidet.



Zu Beginn wird die Eingabe auf ihre syntaktische Gültigkeit überprüft und sofort verworfen, wenn sie nicht die richtige Form hat. Sonst wird die Maschine $N_{M,w}$ wie folgt konstruiert: $N_{M,w}$ überprüft für ihre Eingabe y, ob $y = w$ gilt. Wenn ja, so akzeptiert sie. Sonst arbeitet sie wie M auf y.

Es folgt unmittelbar für alle $y \neq w$, dass $y \in L(M) \iff y \in L(N_{M,w})$. Also unterscheiden sich $L(M)$ und $L(N_{M,w})$ höchstens um w. Weil stets $w \in L(N_{M,w})$ ist, folgt

$$L(M) = L(N_{M,w}) \iff w \in L(M).$$

Das bedeutet, dass auch

$$\mathrm{Kod}(M)\#\mathrm{Kod}(N_{M,w}) \in L_{\mathrm{EQ}} \iff \mathrm{Kod}(M)\#w \in L_{\mathrm{U}}$$

gilt, womit die Korrektheit der Reduktion nachgewiesen ist.