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 7

Zürich, 19. November 2021

Lösung zu Aufgabe 19

(a) Wir führen einen Widerspruchsbeweis und nehmen dazu $L_{\mathrm{H}}^{\complement} \in \mathcal{L}_{\mathrm{RE}}$ an. Es gibt also eine Turingmaschine $M_{\mathrm{H}}^{\complement}$, die $L_{\mathrm{H}}^{\complement}$ akzeptiert.

Zudem existiert folgende einfache Turingmaschine M_{H}, die L_{H} akzeptiert: Für eine Eingabe $x \in \{0, 1, \#\}$ prüft sie, ob $x = \mathrm{Kod}(M)\#w$ für ein $w \in \{0, 1\}^*$. Falls nicht, dann verwirft sie die Eingabe x und sonst simuliert sie M auf w. Falls diese Simulation endet, akzeptiert M_{H} die Eingabe x.

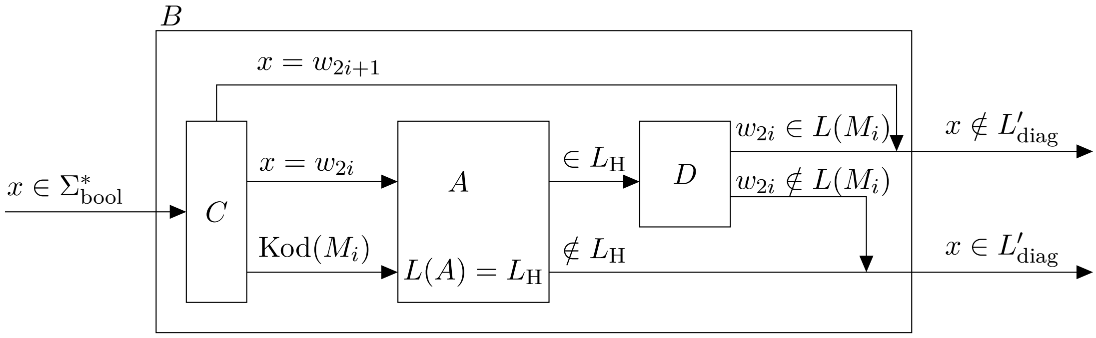
Wir beweisen nun, dass $L(M_{\mathrm{H}}) = L_{\mathrm{H}}$. Falls $x \in L_{\mathrm{H}}$, dann ist $x = \mathrm{Kod}(M)\#w$ und M_H simuliert M auf w und diese Simulation endet, weshalb M_{H} akzeptiert, somit gilt $x \in L(M_{\mathrm{H}})$. Falls hingegen $x \notin L_{\mathrm{H}}$, dann gibt es zwei Möglichkeiten. Entweder hat x nicht die Form $\mathrm{Kod}(M)\#w$, in diesem Fall verwirft M_{H} die Eingabe x sofort. Andernfalls simuliert M_{H} wieder M auf w, doch dieses Mal endet die Simulation nicht, sonst wäre $x \in L_{\mathrm{H}}$. Weil die Simulation nie endet, wird M_{H} also x auch nie akzeptieren, somit gilt $x \notin L(M_{\mathrm{H}})$.

Nun konstruieren wir einen Algorithmus S, der die angenommene Turingmaschine $M_{\mathrm{H}}^{\complement}$ benutzt, um L_{H} zu entscheiden. Auf der Eingabe x simuliert S die beiden Turingmaschinen M_{H} auf x und $M_{\mathrm{H}}^{\complement}$ auf x parallel. Falls in der Simulation M_{H} irgendwann die Eingabe x akzeptiert, dann akzeptiert S sofort die eigene Eingabe. Falls hingegen $M_{\mathrm{H}}^{\complement}$ in der Simulation x irgendwann akzeptiert, dann verwirft S sofort die eigene Eingabe.

Für jedes Wort x gilt entweder $x \in L_{\mathrm{H}}$ oder $x \in L_{\mathrm{H}}^{\complement}$. Im ersten Fall wird M_{H} die Eingabe x in endlicher Zeit akzeptieren und somit S die eigene Eingabe x akzeptieren. Im zweiten Fall wird $M_{\mathrm{H}}^{\complement}$ auf x in endlicher Zeit akzeptieren und S somit die eigene Eingabe x verwerfen.

Damit ist bewiesen, dass S auf jeder Eingabe terminiert und x genau dann akzeptiert, wenn $x \in L_{\mathrm{H}}$, und genau dann verwirft, wenn $x \in L_{\mathrm{H}}^{\complement}$, also $x \notin L_{\mathrm{H}}$. Somit ist S ein Algorithmus, der L_{H} entscheidet, und damit gilt $L_{\mathrm{H}} \in \mathcal{L}_{\mathrm{R}}$. Wir wissen aber aus der Vorlesung, dass $L_{\mathrm{H}} \notin \mathcal{L}_{\mathrm{R}}$, wir haben also den gewünschten Widerspruch.

(b) Um $L'_{\mathrm{diag}} \leq_{\mathrm{R}} L_{\mathrm{H}}$ zu zeigen, nehmen wir an, A sei ein Algorithmus, der L_{H} entscheidet. Dann konstruieren wir einen Algorithmus B, der mit Hilfe von A die Sprache L'_{diag} entscheidet. Der Algorithmus B ist so strukturiert wie in der folgenden Abbildung gezeigt:



Für eine Eingabe $x \in \Sigma_{\text{bool}}^*$ berechnet das Teilprogramm C das $j \in \mathbb{N}$, so dass $x = w_j$ das j-te Wort über Σ_{bool} in kanonischer Ordnung ist. Falls j ungerade ist, falls also $j = 2i + 1$ für ein $i \in \mathbb{N}$, dann verwirft B die Eingabe x. Falls hingegen $j = 2i$ für ein $i \in \mathbb{N}$, dann berechnet C auch die Kodierung $\text{Kod}(M_i)$ der i-ten Turingmaschine in kanonischer Ordnung. Das Teilprogramm A für L_{H} bekommt dann $\text{Kod}(M_i)$ und $x = w_{2i}$ als Eingabe in der Form $\text{Kod}(M_i)\#x$.

Falls A die Eingabe $\text{Kod}(M_i)\#x$ verwirft, dann hält M_i nicht auf w_{2i}, also akzeptiert M_i das Wort w_{2i} auch nicht. Also gilt $w_{2i} \in L'_{\text{diag}}$ und B akzeptiert seine Eingabe $x = w_{2i}$.

Falls A die Eingabe $\text{Kod}(M_i)\#x$ akzeptiert, dann hält M_i auf w_{2i}. In diesem Fall simuliert das Teilprogramm D die Arbeit von M_i auf w_{2i}. Diese Simulation endet auf jeden Fall in endlicher Zeit. Falls die Simulation ergibt, dass M_i das Wort w_{2i} akzeptiert, dann gilt $w_{2i} \notin L'_{\text{diag}}$ und B verwirft seine Eingabe $x = w_{2i}$. Sonst verwirft M_i das Wort w_{2i}, es gilt also $w_{2i} \in L'_{\text{diag}}$ und B akzeptiert seine Eingabe $x = w_{2i}$.

Lösung zu Aufgabe 20

(a) Um $L_{\text{H}} \leq_{\text{EE}} L_{\text{UU},\lambda}$ zu beweisen, geben wir einen Algorithmus F an, der eine Eingabe x für L_{H} in eine Eingabe $f(x)$ für $L_{\text{UU},\lambda}$ transformiert. Zunächst entscheidet F, ob x von der Form $\text{Kod}(M)\#w$ für eine Turingmaschine M und ein Wort $w \in \{0,1\}^*$ ist. Falls nicht, dann gibt F die Ausgabe $f(x) = \lambda$ aus. Ansonsten berechnet F eine modifizierte Version M' der in x eincodierten Turingmaschine M wie folgt. Einerseits werden alle Transitionen, welche in M in den verwerfenden Zustand führen, in M' in den akzeptierenden Zustand umgeleitet. Andererseits ersetzt M' die Eingabe, die zu Beginn auf dem Band steht, immer durch w. Die Ausgabe von F ist dann $f(x) = \text{Kod}(M')\#\text{Kod}(M')$.

Wir beweisen nun die Korrektheit dieser Reduktion, also $x \in L_{\text{H}} \iff f(x) \in L_{\text{UU},\lambda}$ für alle $x \in \{0,1,\#\}^*$.

Nehmen wir zunächst an, dass $x \in L_{\text{H}}$. Dann gilt $x = \text{Kod}(M)\#w$ für eine Turingmaschine M und ein Wort $w \in \{0,1\}^*$ und M hält auf w. Entsprechend hält auch M' auf λ, da M' zuerst λ durch w ersetzt und dann so arbeitet, wie M es tun würde, ausser dass M' immer in den akzeptierenden Zustand geht, wenn M in den verwerfenden Zustand gehen würde. Insbesondere akzeptiert M' also λ und somit gilt $f(x) = \text{Kod}(M')\#\text{Kod}(M') \in L_{\text{UU},\lambda}$ nach der Definition von $L_{\text{UU},\lambda}$ mit $M_1 = M_2 = M'$.

Nehmen wir nun an, dass $x \notin L_{\text{H}}$. Dann ist entweder x nicht von der Form $\text{Kod}(M)\#w$ für eine Turingmaschine M und ein Wort $w \in \{0,1\}^*$, was $f(x) =$

$\lambda \notin L_{\mathrm{UU},\lambda}$ impliziert, oder aber x ist von dieser Form und M hält nicht auf w. Weil M' auf λ zuerst λ durch w ersetzt und dann wie M arbeitet, solange nicht der akzeptierende oder der verwerfende Zustand erreicht wird, hält auch M' nicht und es gilt $f(x) = \mathrm{Kod}(M')\#\mathrm{Kod}(M') \notin L_{\mathrm{UU},\lambda}$ nach der Definition von $L_{\mathrm{UU},\lambda}$ mit $M_1 = M_2 = M'$.

(b) Um $L_{\mathrm{U}}^{\complement} \leq_{\mathrm{EE}} L_{\mathrm{diag}}$ zu beweisen, geben wir einen Algorithmus F an, der eine Eingabe x für $L_{\mathrm{U}}^{\complement}$ zu einer Eingabe $f(x)$ für L_{diag} transformiert. Zunächst entscheidet F, ob x von der Form $\mathrm{Kod}(M)\#w$ für eine Turingmaschine M und ein Wort $w \in \{0,1\}^*$ ist. Falls nicht, dann bestimmt F die Kodierung einer Turingmaschine M^*, die jede Eingabe sofort verwirft, berechnet den Index j dieser Maschine, also die Zahl j, so dass M^* die j-te Turingmaschine ist, und gibt dann das kanonisch j-te Wort über $\{0,1\}$ aus. Die Ausgabe von F ist dann also $f(x) = w_j$. Falls hingegen x von der Form $\mathrm{Kod}(M)\#w$ für eine Turingmaschine M und ein Wort $w \in \{0,1\}^*$ ist, dann bestimmt F die Kodierung einer Turingmaschine M', die unabhängig von der eigenen Eingabe immer M auf w simuliert, berechnet den Index i von M' und gibt dann das kanonisch i-te Wort über $\{0,1\}$ aus. Die Ausgabe von F ist dann also $f(x) = w_i$.

Wir beweisen nun die Korrektheit dieser Reduktion, also $x \in L_{\mathrm{U}}^{\complement} \iff f(x) \in L_{\mathrm{diag}}$ für alle $x \in \{0,1,\#\}^*$.

Nehmen wir zunächst an, dass $x \in L_{\mathrm{U}}^{\complement}$ gilt und unterscheiden zwei Teilfälle. Falls x nicht von der Form $\mathrm{Kod}(M)\#w$ für eine Turingmaschine M und ein Wort $w \in \{0,1\}^*$ ist, dann haben wir $f(x) = w_j$, wobei M_j eine Turingmaschine ist, die alle Wörter sofort verwirft. Nach Definition von L_{diag} gilt also $f(x) \in L_{\mathrm{diag}}$. Falls hingegen $x = \mathrm{Kod}(M)\#w$ für eine Turingmaschine M und ein Wort $w \in \{0,1\}^*$, dann akzeptiert M das Wort w nicht Dies impliziert wiederum, dass M' keine Eingabe akzeptiert, da M' immer M auf w simuliert. Insbesondere akzeptiert $M' = M_i$ auch w_i nicht und somit gilt $f(x) = w_i \in L_{\mathrm{diag}}$.

Nehmen wir nun an, dass $x \notin L_{\mathrm{U}}^{\complement}$, also $x \in L_{\mathrm{U}}$. Entsprechend ist x von der Form $\mathrm{Kod}(M)\#w$ für eine Turingmaschine M und ein Wort $w \in \{0,1\}^*$ und M akzeptiert w. Dann wird M' jede Eingabe akzeptieren, da immer M auf w simuliert wird. Insbesondere akzeptiert $M' = M_i$ das Wort w_i. Somit gilt $f(x) = \mathrm{Kod}(M') \notin L_{\mathrm{diag}}$.

Lösung zu Aufgabe 21

(a) Um zu zeigen, dass $L_{\mathrm{union}} \in \mathcal{L}_{\mathrm{RE}}$ gilt, konstruieren wir eine 3-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 kopiert A zunächst die Eingabe auf das zweite Arbeitsband, bringt anschliessend den Lesekopf und den Kopf auf dem zweiten Arbeitsband zum Anfang des jeweiligen Bands und simuliert schliesslich parallel die Arbeit von M auf w auf dem ersten Arbeitsband und die Arbeit von M' auf w auf dem zweiten und dritten Arbeitsband, wobei das zweite Arbeitsband als Eingabeband von M' verwendet wird.

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 für L_{U} auf zu akzeptierende Eingaben für L_{union} abgebildet werden. Jedes Wort über $\{0, 1, \#\}$, das nicht die Form $\text{Kod}(M)\#w$ hat für eine Turingmaschine M und ein Wort w, 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 $\text{Kod}(A)\#w$, wobei A eine Turingmaschine ist, die M und M' parallel simuliert wie im Aufgabenteil (a) beschrieben. Dann gilt

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

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