ETH zürich
Departement Informatik

Theoretische Informatik

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

Lösungsvorschläge – Blatt 8

Zürich, 16. November 2018

Lösung zu Aufgabe 22

Sei $L \subseteq \{0,1\}^*$ eine unendliche Sprache. Mit Hilfe der Diagonalisierungsmethode können wir eine Teilsprache L' von L definieren, die nicht rekursiv aufzählbar ist. Sei M_i die i-te Turingmaschine in kanonischer Ordnung. Dann definieren wir eine Diagonalsprache relativ zu L wie folgt:

$$L_{\text{diag},L} = \{w \in \{0,1\}^* \mid w = w_i \text{ ist das } i\text{-te Wort in } L \text{ für ein } i \in \mathbb{N} \\ \text{und } M_i \text{ akzeptiert } w_i \text{ nicht}\}.$$

Offenbar gilt $L_{\text{diag},L} \subseteq L$.
Angenommen, $L_{\text{diag},L}$ sei rekursiv aufzählbar. Dann ist $L_{\text{diag},L} = L(M)$ für eine TM M. Weil M eine der Turingmaschinen in der Aufzählung aller Turingmaschinen sein muss, existiert ein $i \in \mathbb{N}$, so dass $M = M_i$. Die Sprache $L_{\text{diag},L}$ kann aber nicht gleich $L(M_i)$ sein, weil

$$w_i \in L_{\text{diag},L} \iff w_i \notin L(M_i)\,.$$

Dies ist ein Widerspruch, also gilt $L_{\text{diag},L} \notin \mathcal{L}_{\text{RE}}$.

Lösung zu Aufgabe 23

(a) Eine korrekt kodierte Eingabe des Halteproblems L_{H} enthält zwei Komponenten, die Kodierung einer TM M und ein Eingabewort x. Daraus konstruieren wir eine Maschine $N_{M,x}$, die auf *jeder* Eingabe wie folgt arbeitet:

> $N_{M,x}$ ignoriert die eigene Eingabe, und simuliert die Arbeit einer TM M' auf x. Die TM M' ist dabei gleich M mit der Modifikation, dass alle Transitionen, die in M nach q_{reject} führen, in M' nach q_{accept} gehen.

Aus $N_{M,x}$ erhält man das Wort $v(N_{M,x})$, das den Index von $N_{M,x}$ beschreibt, es gilt also $v(N_{M,x}) = w_i$, so dass $N_{M,x} = M_i$ in der bei der Definition von L_{diag} verwendeten Aufzählung gilt.

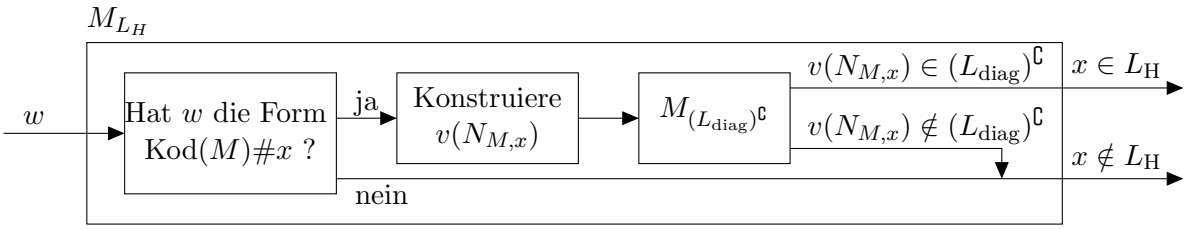
Dann gilt offenbar für alle korrekt kodierten Eingaben $\text{Kod}(M)\#x$, dass

$$\begin{aligned}
\text{Kod}(M)\#x \in L_{\text{H}} &\iff M \text{ hält auf } x \\
&\iff M' \text{ erreicht } q_{\text{accept}} \text{ auf } x \\
&\iff N_{M,x} \text{ akzeptiert jede Eingabe} \\
&\iff N_{M,x} \text{ akzeptiert } v(N_{M,x}) \\
&\iff v(N_{M,x}) \in (L_{\text{diag}})^{\complement}.
\end{aligned}$$

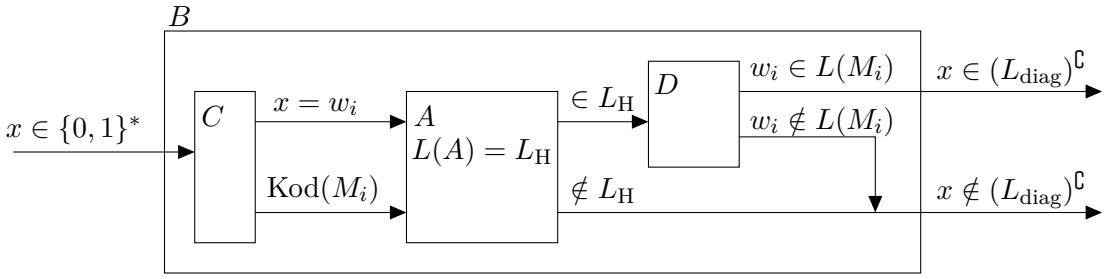
Also liefert die Antwort auf die Frage, ob $v(N_{M,x}) \in (L_{\text{diag}})^{\complement}$ ist, das gewünschte Ergebnis.

Falls die Eingabe w nicht korrekt kodiert ist, falls sie also nicht die Form $\text{Kod}(M)\#x$ hat, geben wir die Antwort "Nein" aus. Dies ist das gewünschte Ergebnis, da in diesem Fall $w \notin L_H$ gilt.

Diese Reduktion lässt sich graphisch wie folgt darstellen:



(b) Um $(L_{\text{diag}})^{\complement} \leq_{\text{R}} L_{\text{H}}$ 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 $(L_{\text{diag}})^{\complement}$ 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 ein $i \in \mathbb{N}$, so dass $x = w_i$, und die Kodierung $\text{Kod}(M_i)$ der i-ten TM. Das Teilprogramm A für L_{H} bekommt $\text{Kod}(M_i)\#x = w_i$ als Eingabe.

Falls A die Eingabe $\text{Kod}(M_i)\#x$ verwirft, dann hält M_i nicht auf w_i, also akzeptiert M_i das Wort w_i auch nicht. Also gilt $w_i \notin (L_{\text{diag}})^{\complement}$ und B verwirft seine Eingabe $x = w_i$.

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

Lösung zu Aufgabe 24

(a) Es gilt

$$(L_{\text{infinite}})^{\complement} = \{w \in \{0,1\}^* \mid w \neq \text{Kod}(M') \text{ für alle TM } M'\} \\ \cup \{\text{Kod}(M) \mid \text{es gibt eine Eingabe } x, \text{ so dass } M \text{ auf } x \text{ hält}\}\,.$$

Wir beschreiben im Folgenden eine nichtdeterministische Turingmaschine N mit $L(N) = (L_{\text{infinite}})^{\complement}$. Es gibt genau also für jedes Wort aus $w \in (L_{\text{infinite}})^{\complement}$ eine endliche, akzeptierende Berechnung von N auf w. Die NTM N überprüft für eine Eingabe w zunächst, ob $w = \text{Kod}(M)$ für irgendeine TM M. Falls nicht, dann akzeptiert N die Eingabe w. Falls $w = \text{Kod}(M)$ gilt für eine TM M, dann wählt N nichtdeterministisch ein Wort x über dem Eingabealphabet von M und simuliert M deterministisch auf x.

Falls M auf dem Wort x hält, dann akzeptiert N ihre Eingabe w. Falls M auf x unendlich lange rechnet, dann ist auch die Berechnung von N auf w unendlich.

Offenbar akzeptiert N alle Wörter, die nicht die Kodierung einer Turingmaschine sind. Falls die Eingabe w die Kodierung einer TM M ist, dann ist $w \in (L_{\text{infinite}})^{\complement}$ genau dann, wenn es ein Wort x gibt, so dass M auf x hält. Also gibt es dann eine akzeptierende Berechnung von N auf w, in der N genau dieses Wort x nichtdeterministisch wählt. Falls $w \in L_{\text{infinite}}$ gilt, dann gibt es keine akzeptierende Berechnung von N auf w. Also ist N eine NTM, die $(L_{\text{infinite}})^{\complement}$ akzeptiert, damit gilt $(L_{\text{infinite}})^{\complement} \in \mathcal{L}_{\text{RE}}$.

(b) Wir zeigen $L_{\text{H},\lambda} \leq_{\text{EE}} (L_{\text{infinite}})^{\complement}$. Hierfür konstruieren wir eine TM A, die jede beliebige Eingabe w für $L_{\text{H},\lambda}$ in eine Eingabe $A(w)$ für $(L_{\text{infinite}})^{\complement}$ umwandelt, so dass

$$w \in L_{\text{H},\lambda} \iff A(w) \in (L_{\text{infinite}})^{\complement}\,.$$

Die TM A testet zunächst, ob $w = \text{Kod}(M)$ gilt für irgendeine TM M. Falls nicht, konstruiert A die Kodierung einer TM M_1, die auf keinem Wort hält.

Falls $w = \text{Kod}(M)$, dann konstruiert A die Kodierung einer TM N_M, die wie folgt arbeitet: N_M ignoriert ihre eigene Eingabe und simuliert M auf λ.

Wir zeigen nun, dass $w \in L_{\text{H},\lambda} \iff A(w) \in (L_{\text{infinite}})^{\complement}$ gilt.

Sei zunächst $w \notin L_{\text{H},\lambda}$. Falls w nicht die Kodierung einer TM ist, dann ist $A(w) = M_1$, also $A(w) \in L_{\text{infinite}}$ und damit $A(w) \notin (L_{\text{infinite}})^{\complement}$. Falls $w = \text{Kod}(M)$ für eine TM M, dann hält M nicht auf λ. Damit hält N_M auf keiner Eingabe, also ist $A(w) \notin (L_{\text{infinite}})^{\complement}$.

Sei nun $w \in L_{\text{H},\lambda}$. Dann gilt $w = \text{Kod}(M)$ für eine TM M, die auf der Eingabe λ hält. Damit hält N_M auf jeder Eingabe, weil N_M auf jeder Eingabe identisch arbeitet und M auf λ simuliert. Also gilt $A(w) = \text{Kod}(N_M) \in (L_{\text{infinite}})^{\complement}$.

(c) Wir zeigen zunächst die folgende Hilfsaussage, die bereits in der Vorlesung bewiesen und verwendet wurde, aber nicht im Buch bewiesen ist:

> Sei L eine Sprache über einem Alphabet Σ. Falls $L \in \mathcal{L}_{\text{RE}}$ und $L^{\complement} \in \mathcal{L}_{\text{RE}}$, dann gilt auch $L \in \mathcal{L}_{\text{R}}$.

Um diese Aussage zu beweisen, konstruieren wir aus zwei Turingmaschinen M und M_c, die L und $L^{\complement}$ akzeptieren, eine neue Turingmaschine A, die L entscheidet. Die Maschine A simuliert für eine Eingabe $w \in \Sigma^*$ einfach parallel die beiden Maschinen

M und M_c auf w. Falls $w \in L$, dann akzeptiert die Maschine M nach endlicher Zeit das Wort w und A akzeptiert ebenfalls. Falls $w \notin L$, dann akzeptiert M_c nach endlicher Zeit das Wort w und A verwirft die Eingabe. In beiden Fällen gibt A also nach endlich vielen Schritten die korrekte Antwort.

Wir können diese Hilfsaussage jetzt einfach verwenden, um $L_{\text{infinite}} \notin \mathcal{L}_{\text{RE}}$ zu zeigen. Wir führen einen indirekten Beweis: Angenommen $L_{\text{infinite}} \in \mathcal{L}_{\text{RE}}$. Nach Aufgabenteil (a) ist auch $(L_{\text{infinite}})^{\complement} \in \mathcal{L}_{\text{RE}}$. Mit der Hilfsaussage folgt damit, dass $(L_{\text{infinite}})^{\complement} \in \mathcal{L}_{\text{R}}$, im Widerspruch zu Aufgabenteil (b). Also ist die Annahme falsch und $L_{\text{infinite}} \notin \mathcal{L}_{\text{RE}}$.