

Übungsstunde 7

Sonntag, 5. November 2023 10:06

Das wichtigste zu Turingmaschinen: (TM)

Was ist eine TM? Eine TM beschreibt eine abstrakte Maschine, welche in der Lage ist, jede Aufgabe (unabhängig von dessen Sinnhaftigkeit) auszuführen

Was ist ein Algorithmus? Ein Algorithmus ist eine TM welche immer hält (i.e. jede Eingabe entweder akzeptiert oder verwirft)

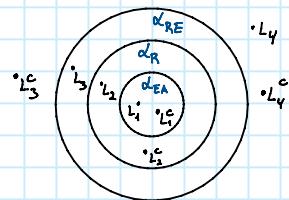
Was kann eine TM? $\begin{cases} \text{Halten (akzeptieren / verwirfen)} \\ \text{nicht Halten} \end{cases}$

Die Lösungsmenge einer Sprache wird von einer TM immer in endlicher Zeit verifiziert!

Sprachenpuzzel

$$L_{RE} = \{ L(M) \mid M \text{ ist TM} \}$$

$$L_R = \{ L(M) \mid M \text{ ist TM, die immer hält} \}$$



Reduktionen

• $L_1 \leq_R L_2$: "L₁ auf L₂ rekursiv reduzierbar", falls $L_2 \in L_R \Rightarrow L_1 \in L_R$

"Wenn ich L₂ lösen kann, dann kann ich auch L₁ lösen"

• $L_1 \leq_{EE} L_2$: Eingabe - zu - Eingabe Reduktion: Gib eine TM M an, welche ein Wort in L₁ zu einem äquivalenten Wort in L₂ umformt. $\forall x \in \Sigma_1^* : x \in L_1 \Leftrightarrow f_M(x) \in L_2$, wobei:

$f_M : \Sigma_1^* \rightarrow \Sigma_2^*$ die von M berechnete Abbildung ist

• Lemma 5.3: $L_1 \subseteq \Sigma_1^*, L_2 \subseteq \Sigma_2^* : L_1 \leq_{EE} L_2 \Rightarrow L_1 \leq_R L_2$

• Wann verwende ich was?

Reicht es die Eingabe zu transformieren bzw. kann die Lösung übernommen werden? **EE-Reduktion**

Kann die Lösung durch die Verwendung von L₂ besser / direkt bestimmt werden bzw. " $\in L_1 \Leftrightarrow \in L_2$ "?

R-Reduktion

Reduktionen

Beh.: $L_H \leq_E L_{\text{uu}, \lambda} = \{ \text{Kod}(M_1) \# \text{Kod}(M_2) \mid M_1 \text{ & } M_2 \text{ akzeptieren } \lambda \}$

Beweis: Direkt. Wir beschreiben eine TM M welche immer hält (Algo.), die für jede Eingabe $w \in \{0,1,\#\}^*$ eine Eingabe $M(w) \in \{0,1,\#\}^*$ generiert, s.d. $w \in L_H \iff M(w) \in L_{\text{uu}, \lambda}$
 M arbeitet auf $w \in \{0,1,\#\}^*$ wie folgt: M prüft ob $w = \text{Kod}(A) \# x$ für $x \in \{0,1\}^*$ und $\text{Kod}(A) \in \text{KodTM}$. Falls nicht, so hält M mit Bandinhalt λ , i.e. $M(w) = \lambda$.
Andernfalls, hält M mit Bandinhalt $\text{Kod}(B) \# \text{Kod}(B)$ ($= M(w)$), wobei B wie folgt arbeitet:
 B simuliert unabhängig von seiner Eingabe A auf x .

Falls x von A akzeptiert wird, so akzeptiert auch B .

Falls x von A verworfen wird, so akzeptiert B .

Falls A nicht auf x hält, so hält, so hält auch B nicht.

Beh.: $w \in L_H \iff M(w) \in L_{\text{uu}, \lambda}$

Beweis: $w \in L_H \iff w = \text{Kod}(A) \# x \wedge A \text{ hält auf } x$

$\iff B \text{ akzeptiert jede Eingabe}$

$\iff B \text{ akzeptiert } \lambda$

$\iff \text{Kod}(B) \# \text{Kod}(B) \in L_{\text{uu}, \lambda}$

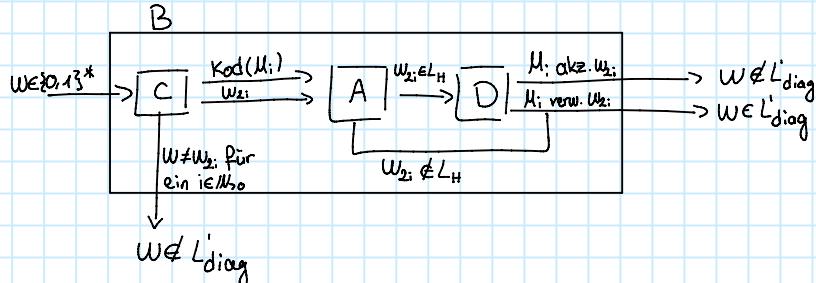
□

Aliter: Wir geben einen Algo. 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 TM 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 einderten TM M wie folgt:
Alle Transitionen zu q_f werden in q_0 umgeleitet und M' ersetzt die Eingabe, welche zu Beginn auf dem Band steht durch x .

Beh.: $x \in L_H \iff f(x) \in L_{\text{uu}, \lambda}$

Beh.: $L'_{\text{diag}} \leq_R L_H$, $L'_{\text{diag}} = \{w \in \{0,1\}^* \mid w = w_{2i}, i \in \mathbb{N} \text{ und } M_i \text{ akzeptiert } w \text{ nicht}\}$

Beweis: Direkt. Angenommen \exists Algorithmus A, der L_H entscheidet. Wir konstruieren hieraus einen Algo. B, der mit Hilfe von A die Sprache L'_{diag} entscheidet.



C prüft ob $w = w_{2i}$ für ein $i \in \mathbb{N}_{>0}$. Da dies nur eine Indizabfrage ist, ist klar, dass C in endlicher Zeit terminiert. Falls nicht, so verwirft B auf w. Falls doch, gibt C $\text{Kod}(M_i) \# w_{2i}$ an A weiter. D simuliert M_i auf w_{2i} . Dies hölt immer, da nur Eingaben simuliert werden, welche also auch in L_H sind, also halten. Somit ist klar, dass B immer hölt.

Beh.: $L(B) = L'_{\text{diag}}$

Beweis: Direkt

$$w \in L'_{\text{diag}} \iff w = w_{2i} \wedge M_i \text{ verwirft } w_{2i} \iff w \in L(B)$$

□

□

Aufgabe:

a) $L_H^C \notin \mathcal{L}_{\text{RE}}$

$$L_{\text{union}} = \{\text{Kod}(M) \# \text{Kod}(M') \# w \mid w \in L(M) \cup L(M')\}$$

b) $L_{\text{union}} \in \mathcal{L}_{\text{RE}}$

Lösung:

a) Angenommen $L_H^C \in \mathcal{L}_{\text{RE}} \wedge L_H \in \mathcal{L}_{\text{RE}}$ (VL). Dann folgt $L_H \in \mathcal{L}_{\text{R}}$ \Leftrightarrow

Wir konstruieren hierfür eine TM S die immer hölt, s. d. $L_H = L(S)$.

S arbeitet auf einer Eingabe $w \in \{0,1,\#\}^*$ wie folgt: Seien A, B TM, s. d. $L_H = L(A)$ und $L_H^C = L(B)$. S simuliert parallel A und B auf w und hölt, falls A oder B halten.

Fall 1: A akzept. w \Rightarrow S akzept. Eingabe w.

Fall 2: B akzept. w \Rightarrow S verwirft w.

Da für jedes $x \in \{0,1,\#\}^*$ entweder $x \in L_H$ oder $x \in L_H^C$ gilt, hölt S immer.

Klar gilt: $L(S) = L_H$ \Leftrightarrow

□

b) Wir konstruieren eine TM A s.d. $L_{\text{union}} = L(A)$. A arbeitet auf einer Eingabe $w \in \{0,1,\#\}^*$
wie folgt: A prüft die Form von w.

Falls $w \neq \text{Kod}(M) \# \text{Kod}(M') \# x$, so verwirft A auf w.

Falls $w = \text{Kod}(M) \# \text{Kod}(M') \# x$, so werden $M \& M'$ auf x simuliert.

Falls x von M oder M' akzeptiert wird, so akzeptiert auch A auf w.

Falls x von $M \& M'$ verworfen wird, so verwirft auch A.

Falls M oder M' auf x nicht hält, so hält auch A nicht.

Es ist klar, dass $L(A) = L_{\text{union}}$

□

Songempfehlung: Gasoline - I Prevail
All around us - Neelix, Durs