Übungsstunde 7 - Should have stayed with 6

Sonntag, 6. November 2022 22:05

Serie 6, 7

Das wichtigste zu Turingmaschinen: (TM)

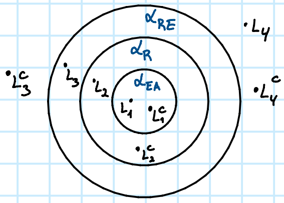
Was ist eine TM? Eine TM beschreibt eine abstrakte Maschiene, welche in der Lage ist, jede Aufgabe (unabhängig von dessen Sinnhaftigkeit) auszuführen

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

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

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

Sprachenpuzzel



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

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

Reduktionen

- $L_1 \leq_R L_2$: "L_1 auf L_2 rekursiv reduzierbar", falls $L_2 \in \mathcal{L}_R \Rightarrow L_1 \in \mathcal{L}_R$

  "Wenn ich L_2 lösen kann, dann kann ich auch L_1 lösen"
- $L_1 \leq_{EE} L_2$: Eingabe - zu - Eingabe Reduktion: Gib eine TM M an, welche ein Wort in L_1 zu einem äquivalenten Wort in L_2 umformt. $\forall x \in \Sigma_1^*: x \in L_1 \Leftrightarrow f_M(x) \in L_2$, wobei $f_M: \Sigma_1^* \to \Sigma_2^*$ 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 L_2 besser / direkt bestimmt werden bzw. "$\in L_1 \Leftrightarrow \neg \in L_2$"? R-Reduktion

Diagonalisierungsmethode

Beh.: $L_2 := \{w \in \{0,1\}^* \mid w = w_i \text{ für ein } i \in \mathbb{N}_{\geq 1} \text{ und } M_{\lceil i/2 \rceil} \text{ akzeptiert } w \text{ nicht}\} \notin \mathcal{L}_{RE}$

Beweis: Per Widerspruch

Angenommen $L_2 \in \mathcal{L}_{RE}$. Dann $\exists$ TM s.d. $L(M) = L_1$. Insbesondere $\exists i \in \mathbb{N}_{\geq 1}$ s.d. $M_i = M$.

Also $L_2 = L(M) = L(M_i)$. Betrachte das Wort w_{2i}. Dann gilt: Wort in Abhängigkeit von TM in Sprache wählen!

<u>Fall 1:</u> $w_{2i} \in L(M_i) \Rightarrow M_i$ akzeptiert $w_{2i} \Rightarrow w_{2i} \notin L_2$, da $M_{\lceil 2i/2 \rceil} = M_i$ akzeptiert w_{2i}

$\Rightarrow w_{2i} \notin L_2 = L(M_i)$ ↯

<u>Fall 2</u>: $w_{2i} \notin L(M_i) \Rightarrow M_i$ akzeptiert w_{2i} nicht $\Rightarrow w_{2i} \in L_2 = L(M_i)$ ↯

Somit war unsere Annahme falsch und $L_2 \notin \mathcal{L}_{RE}$. □

Der Beweis basiert auf der Idee, dass in der Liste jede TM erfasst ist und daher auch die angenommene Maschiene für die Sprache (i.e. $\mathbb{N} \to \mathbb{N}$, $i \mapsto \lceil i/2 \rceil$ ist surjektiv). Daher führt dasselbe Argument nicht zum Widerspruch wenn wir nicht-surjektive Abbildungen haben).

Reduktionen

<u>Beh.:</u> $L_H \leq_{EE} L_{uu,\lambda} = \{ Kod(M_1)\#Kod(M_2) \mid M_1 \,\&\, M_2 \text{ akzeptieren } \lambda \}$

<u>Beweis:</u> 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. $w \in L_H \iff M(w) \in L_{uu,\lambda}$

M arbeitet auf $w \in \{0,1,\#\}^*$ wie folgt: M prüft ob $w = Kod(A)\#x$ für $x \in \{0,1\}^*$ und $Kod(A) \in$ KodTM. Falls nicht, so hält M mit Bandinhalt λ, i.e. $M(w) = \lambda$

Andernfalls, hält M mit Bandinhalt $Kod(B)\#Kod(B)$ $(= M(w))$, 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.

<u>Beh.:</u> $w \in L_H \iff M(w) \in L_{uu,\lambda}$

<u>Beweis:</u> $w \in L_H \iff w = Kod(A)\#x \wedge A$ hält auf x

$\iff B$ akzeptiert jede Eingabe

$\iff B$ akzeptiert λ

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

□

Aliter: Wir geben einen Algo. F an, der eine Eingabe x für L_H in eine Eingabe $f(x)$ für $L_{uu,\lambda}$ transformiert. Zunächst entscheidet F, ob x von der Form $Kod(M)\#w$ für eine TM M und ein Wort $w \in \{0,1\}^*$ ist. Falls nicht, dann gibt F die Ausgabe $f(x) = \lambda$ aus.

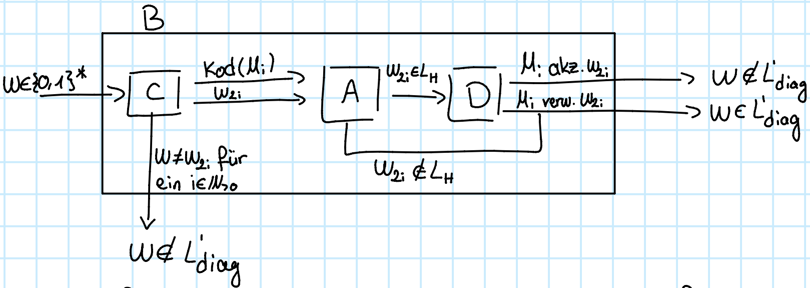
Ansonsten berechnet F eine modifizierte Version M' der in x eincodierten TM M' wie folgt:

Alle Transitionen zu q_r werden in q_a umgeleitet und M' ersetzt die Eingabe, welche zu beginn auf dem Band steht durch x.

<u>Beh.:</u> $x \in L_H \iff f(x) \in L_{uu,\lambda}$

Beh.: $L'_{diag} \le_R L_H$, $L'_{diag} = \{w \in \{0,1\}^* \mid w = w_{2i}, i \in \mathbb{N}$ und M_i akzeptiert w 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 Indizeabfrage ist, ist klar, dass C in endlicher Zeit terminiert. Falls nicht, so verwirft B auf w. Falls doch, gibt C $Kod(M_i)\#w_{2i}$ an A weiter. D simuliert M_i auf w_{2i}. Dies hält immer, da nur Eingaben simuliert werden, welche also auch in L_H sind, also halten. Somit ist klar, dass B immer hält.

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

Beweis: Direkt

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

Aufgabe:

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

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

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

Lösung:

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

Wir konstruieren hierfür eine TM S die immer hält, 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 hält, falls A oder B halten.

Fall 1: A akzep. $w \Rightarrow S$ akzep. Eingabe w.

Fall 2: B akzept. $w \Rightarrow S$ verweigert w.

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

Klar gilt: $L(S) = L_H$ ↯ $\square$

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

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

Falls $w = Kod(M)\#Kod(M')\#x$, 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 $L(A) = L_{union}$ $\square$

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