Übungsstunde 7&8

Sonntag, 5. November 2023 10:06

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)

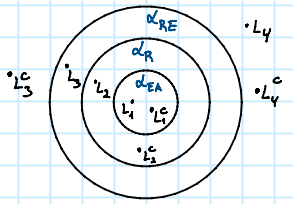
Was kann eine TM? { Halten (akzeptieren / verwerfen) ; nicht Halten

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 rekursir 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

Reduktionen

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

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

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.

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

Beweis: $w \in L_H \iff w = Kod(A)\#x \wedge A$ 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.

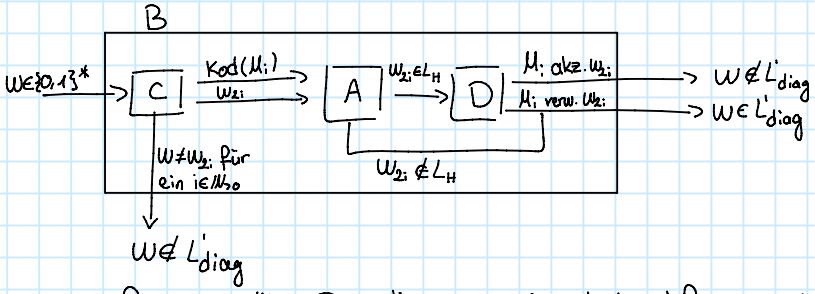
Ansonsten berechnet F eine modifizierte Version M' der in x eincodierten TM M' wie folgt:

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

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

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

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 $w = w_i$ 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_i an A weiter. D simuliert M_i auf w_i. 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

$w \in L'_{diag} \iff w = w_i \wedge M_i \text{ verwirft } w_i \iff w \in L(B)$ □ ■

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: $L(S) = L_H$ ↯ □

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 $L(A) = L_{union}$ □

Frage: Kann es $L_1 \notin \mathcal{L}_{RE}$ und $L_2 \in \mathcal{L}_{RE}$ geben, s.d. $L_1 \leq_R L_2$? JA, z.B. $L_2 = (L_{diag})^c$ und $L_1 = L_{diag}$

⚠ Zu jeder NTM M existiert eine TM M' s.d. $L(M) = L(M')$. Wenn wir also $L \in \mathcal{L}_{RE}$ zeigen sollen, können wir einfach eine NTM angeben! Dies ist häufig einfacher als eine TM zu finden

Aufgabe: Sei $L_{all} = \{Kod(M) \mid L(M) = \Sigma^*_{bool}\}$. Dann ist $L_{all} \notin \mathcal{L}_R$

Lösung: Wir benutzen den Satz von Rice. (üblicherweise steht in der Aufgabe "sie dürfen alle in der VL gezeigten Resultate verwenden) Wir zeigen also, dass L_{all} ein nicht triviales Entscheidungsproblem ist. Bemerke zunächst, dass $L_{all} \subseteq KodTM$. Wir prüfen die Bedingungen:

i) Die TM M welche seine Eingabe löscht und dann akzeptiert ist in $L_{all} \Rightarrow L_{all} \neq \emptyset$

ii) Sei M eine TM s.d. $\lambda \notin L(M) \Rightarrow Kod(M) \notin L_{all} \Rightarrow L_{all} \neq KodTM$

iii) Per Def. sind alle TM, welche Σ^*_{bool} akzeptieren in L_{all} enthalten, also gilt

$$L(A) = L(B) \Rightarrow [Kod(A) \in L_{all} \Leftrightarrow Kod(B) \in L_{all}]$$

Nach Rice gilt also: $L_{all} \notin \mathcal{L}_R$ □

Aufgabe: Alternativ Lösung zu $L_\infty^c \in \mathcal{L}_{RE}$

Lösung: Wir benutzen ein Diagonalisierungsargument und konstruieren eine det. TM A s.d. $L(A) = L_\infty^c$. A arbeitet auf einer Eingabe w wie folgt

1. A prüft ob $w \neq Kod(M)$ für eine TM M. Dann akzeptiert A.
2. $w = Kod(M)$. A konstruiert systematisch (wie Diagonale) alle Paare $(i,j) \in \mathbb{N}_{>0}^2$. Für jedes Paar (i,j) generiert A das kanonisch i-te Wort $w_i \in \{0,1\}^*$ und simuliert j Berechnungsschritte von M auf w_i. Falls für ein Paar (k,L) die TM w_k in L Schritten akzeptiert, so akzept. M und A akzeptiert w. Ansonsten akzeptiert A die Eingabe w nicht, da A ∞-Lange arbeitet

□

Es ist trivialerweise offensichtlich, dass $L(A) = L_\infty^c$

Beh.: $L = \{ Kod(M_1) \# Kod(M_2) \# k \mid M_1, M_2 \text{ TM über } \Sigma \text{ und } \Sigma^k \notin L(M_1) \cup L(M_2) \} \notin \mathcal{L}_{RE}$.

Beweis: Wir benutzen, dass $L^c \in \mathcal{L}_{RE}$. Dies kann man via eines Diagonalisierungsarguments ähnlich zur vorigen Aufgabe zeigen und ist dem emsigen Leser überlassen (Sessionsprüfung HS2020 A4).

Wir zeigen $L_u^c \leq_{EE} L$, da dann $L_u^c \leq_R L$ und $L_u^c \notin \mathcal{L}_{RE}$ gilt. Wir beschreiben eine TM S, die eine Funktion $f_S : \{0,1,\#\}^* \to (\{0,1,\#\} \cup \mathbb{N})^*$ berechnet, s.d.

$$x \in L_u^c \iff f_S(x) \in L$$

Sei $x \in \{0,1,\#\}^*$ die Eingabe. S prüft ob $x = Kod(M) \# w$ für eine TM M und ein $w \in \{0,1\}^*$. Falls nicht, so wird $f_S(x) = Kod(M_\emptyset) \# Kod(M_\emptyset) \# 1$ ausgegeben, wobei $M_\emptyset$ eine TM s.d. $L(M_\emptyset) = \emptyset$. Falls doch, so wird $f_S(x) = Kod(M_w) \# Kod(M_w) \# 1$ ausgegeben, wobei M_w eine TM ist, welche ihre eigene Eingabe ignoriert, M auf w simuliert und die Ausgabe übernimmt.

Beh.: $x \in L_u^c \iff f_S(x) \in L$

Beweis: Fall 1: $w \notin KodTM \cdot \{\#\} \cdot \{0,1\}^*$ (i.e. $w \in L_u^c$) $\Rightarrow f_S(w) = Kod(M_\emptyset) \# Kod(M_\emptyset) \# 1 \in L$, da

$$\Sigma \notin \emptyset = \emptyset \cup \emptyset$$

Fall 2: $x = Kod(M) \# w$ für $Kod(M) \in KodTM$, $w \in \{0,1\}^*$.

$$x \in L_u^c \iff w \notin L(M) \iff L(M_w) = \emptyset \iff \Sigma \notin L(M_w) \cup L(M_w)$$
$$\iff f_S(x) \in L$$

□

Da für jede Sprache $\tilde{L}$: $\tilde{L}, \tilde{L}^c \in \mathcal{L}_{RE} \iff \tilde{L} \in \mathcal{L}_R$ gilt, folgt $L \notin \mathcal{L}_{RE}$

□