Prof. Javier Esparza
Philipp Czerner, Martin Helfrich

Technische Universität München
Lehrstuhl für Theoretische Informatik

Einführung in die Theoretische Informatik

Sommersemester 2022 – Übungsblatt 11

- Das Übungsblatt ist in zwei Teile gegliedert: den Vorbereitungsteil, den Sie vor der Übung selbstständig bearbeiten sollen, und den Übungs-/Nachbereitungsteil, der Aufgaben enthält, die in der Übung besprochen werden und von Ihnen anschließend zur Nachbereitung verwendet werden können.
- Das ist nicht das Hausaufgabenblatt! Die Hausaufgaben finden Sie auf einem separaten Blatt.

Vorbereitung (→ vor der Übung selbständig zu bearbeiten)

Individualaufgabe Ü11.1. (*Wichtige Begriffe*)

Überprüfen Sie, dass Sie die folgenden Begriffe oder Notationen korrekt definieren können.

- Postsche Korrespondenzproblem (PCP)
- Reduktion
- berechenbar
- totale Funktion
- entscheidbar / unentscheidbar
- semi-entscheidbar (bzw. rekursiv aufzählbar)

Individualaufgabe Ü11.2. (H_{NEQ})

Sei Σ ein beliebiges Alphabet. Zeigen Sie: Es ist unentscheidbar zu prüfen, ob für zwei als Eingabe gegebene Turing-Maschinen die eine auf allen Eingaben genau dann hält, wenn die andere nicht hält. Formaler: Zeigen Sie, die Menge

$$\mathcal{H}_{NEQ} := \{w_1\#w_2 \mid w_1, w_2 \in \{0,1\}^* \text{ und } \forall x \in \Sigma^*.\ M_{w_1}[x]\downarrow \iff \neg M_{w_2}[x]\downarrow\}$$

ist unentscheidbar.

Lösungsskizze. Für diese Aufgabe gibt es eine Videolösung: Link

Wir reduzieren das Halteproblem auf leerem Band $\mathcal{H}_0$ auf $\mathcal{H}_{NEQ}$. Dabei gilt zu beachten, dass die Reduktion total, berechenbar und korrekt sein muss.

Reduktion von $\mathcal{H}_0$: Sei $w_\perp$ die Kodierung einer Turing-Maschine, die auf keiner Eingabe hält. Sei $w \in \{0,1\}^*$ beliebig. Wir berechnen zunächst die Kodierung w' einer Turing-Maschine, die bei jeder Eingabe das Band löscht und dann $M_w[\epsilon]$ ausführt. Anschließend geben wir $w'\#w_\perp$ zurück.

Die Reduktion ist total: Für jede Eingabe w wird die Ausgabe $w'\#w_\perp$ erzeugt.

Die Reduktion ist berechenbar: Die En- und Dekodierungsfunktionen für Turing-Maschinen sind berechenbar. Außerdem kann eine TM alle Zeichen auf dem Band, die nicht dem

⟨1⟩ Ist $L(G_1) \cap L(G_2) = \emptyset$?

⟨2⟩ Ist $|L(G_1) \cap L(G_2)| < \infty$? → Ja: $|L(G_1) \cap L(G_1)| = n \geq 0$

Nein: $|L(G_1) \cap L(G_2)| = \infty > 0$

Z: $\langle 1 \rangle \leq \langle 2 \rangle$

- Gegeben G_1, G_2 CFG
- Manipuliere $G_1 \mapsto G_1'$, $G_2 \mapsto G_2'$

sodass $|L(G_1') \cap L(G_2')| < \infty$

$\iff L(G_1) \cap L(G_2) = \emptyset$

$$\underbrace{L(G_1') \cap L(G_2')}_{L} = \underbrace{A\,(L(G_1) \cap L(G_2))}_{R}$$

* wenn $R = \emptyset$, dann $L = \emptyset \Rightarrow |L| = 0 < \infty$
* wenn $R \neq \emptyset$, dann wollen wir $|A| = \infty$

Man könnte wählen: Σ^* $\Rightarrow |L| = \infty$

↳ $R \neq \emptyset \Rightarrow |L(G_1') \cap L(G_2')| = \infty$

Wir wählen $A = L(a^* d)$ $\Sigma^* \{\#\}$

G_1' so definiert, dass: $L(G_1') = L(a^* d) L(G_1)$

G_2' —— " ——: $L(G_2') = L(a^* d) L(G_2)$

Dann gilt:

$$L(G_1') \cap L(G_2') = L(a^* d) L(G_1) \cap L(a^* d) L(G_2)$$
$$= L(a^* d)\,(L(G_1) \cap L(G_2))$$

$G_1': \quad S' \to aS' \mid d\underline{\underline{S}}$ $\qquad S$ Startsymbol von G_1

Am Ende gilt:

$$|L(G_1') \cap L(G_2')| < \infty \iff L(G_1) \cap L(G_2) = \emptyset$$

$$L(G_1') = \Sigma^* L(G_1)$$

$$L(G_2') = \Sigma^* L(G_2)$$

$$L(G_1') \cap L(G_2') = \overbrace{\Sigma^*}^{a} \overbrace{L(G_1)}^{ab} \cap \overbrace{\Sigma^*}^{aa} \overbrace{L(G_2)}^{b}$$

$$= \Sigma^* \underbrace{(L(G_1) \cap L(G_2))}_{\emptyset}$$

$$A(B \cap C) \neq AB \cap AC$$

$\langle 4 \rangle$ Ist $L(G_1) \subseteq L(G_2)$?

$\langle 5 \rangle$ Ist $L(G_1) = L(G_2)$?

- Gegeben G_1, G_2 CFGs
- Konstruiere G_1', G_2' sodass

$$L(G_1) \subseteq L(G_2)$$

$$\Leftrightarrow L(G_1') = L(G_2')$$

Hinweis: $\boxed{A \subseteq B \Leftrightarrow A \cup B = B}$

Definiere G_1', sodass $L(G_1') = L(G_1) \cup L(G_2)$

" G_2', sodass $L(G_2') = L(G_2)$

$\hookrightarrow G_2' := G_2$

$\hookrightarrow G_1'$: $S \rightarrow S_1 \mid S_2$

wobei S_1 St. symbol von G_1

S_2 —"— von G_2

Dann gilt:

$$L(G_1) \subseteq L(G_2) \Leftrightarrow \underline{L(G_1) \cup L(G_2)} = L(G_2)$$

$$\Leftrightarrow L(G_1') = L(G_2')$$

Leerzeichen entsprechen, durch geeignete Übergänge anfangs überschreiben.

Die Reduktion ist korrekt:

$$\begin{aligned}
w \in \mathcal{H}_0 &\iff M_w[\epsilon]{\downarrow} && \text{(Def. } \mathcal{H}_0\text{)}\\
&\iff \forall x \in \Sigma^*.\ M_{w'}[x]{\downarrow} && (M_{w'} \text{ führt immer } M_w[\epsilon] \text{ aus})\\
&\iff \forall x \in \Sigma^*.(M_{w'}[x]{\downarrow} \text{ und } \neg M_{w_\perp}[x]{\downarrow}) && (M_{w_\perp} \text{ hält nie})\\
&\iff w'\#w_\perp \in \mathcal{H}_{NEQ} && \text{(Def. } \mathcal{H}_{NEQ}\text{)}
\end{aligned}$$

□

Übung und Nachbereitung

Übungsaufgabe Ü11.3. (*Unentscheidbare Typ-2 Probleme*)

In der Vorlesung haben wir folgende Probleme für kontextfreie Grammatiken G_1, G_2 über einem Alphabet Σ kennengelernt:

$\langle 1 \rangle$ Ist $L(G_1) \cap L(G_2) = \emptyset$?

$\langle 2 \rangle$ Ist $|L(G_1) \cap L(G_2)| < \infty$?

$\langle 3 \rangle$ Ist $L(G_1) \cap L(G_2)$ kontextfrei?

$\langle 4 \rangle$ Ist $L(G_1) \subseteq L(G_2)$?

$\langle 5 \rangle$ Ist $L(G_1) = L(G_2)$?

Von $\langle 1 \rangle$ wurde in der Vorlesung gezeigt, dass es unentscheidbar ist. In der Übung und den Hausaufgaben dieser Woche wollen wir auch die anderen Probleme behandeln. Zeigen Sie:

(a) $\langle 1 \rangle \leq \langle 2 \rangle$

(b) $\langle 4 \rangle \leq \langle 5 \rangle$

(c) $\langle 1 \rangle \leq \langle 3 \rangle$

(d) $\langle 1 \rangle \leq \langle 4 \rangle$

Hinweise: Zur Vereinfachung dürfen Sie $\Sigma = \{a, b\}$ jeweils für das zu reduzierende Problem annehmen (ohne das Problem, auf das Sie reduzieren, einzuschränken). Bei der (d) ist es sinnvoll, die Sprache $L(G_1)\{\$\}L(G_2)^R$ zu betrachten.

Übungsaufgabe Ü11.4. (*Überblick: Entscheidbarkeit*)

Vervollständigen Sie das folgende Diagramm.

(a) Fügen Sie die Begriffe „entscheidbar“, „unentscheidbar“, „semi-entscheidbar“, „rekursiv aufzählbar“, „co-semi-entscheidbar“[1] und „nulli-entscheidbar“[2] sinnvoll zum Diagramm hinzu.

(b) Sei $\mathcal{V}_f = \{w \mid \varphi_w = f\}$ für eine totale berechenbare Funktion $f : \mathbb{N} \to \mathbb{N}$. Ordnen Sie diese Beispiel-Probleme richtig zu: $\{x \in \mathbb{N} \mid x = f(1)\}$, $\mathcal{H}$, $\overline{\mathcal{H}}$, $\mathcal{H}_0$, $\overline{\mathcal{H}_0}$, $\mathcal{H}_{\Sigma^*}$, $\overline{\mathcal{H}_{\Sigma^*}}$, $\mathcal{K}$, $\overline{\mathcal{K}}$, $\mathcal{H}_{NEQ}$, $\overline{\mathcal{H}_{NEQ}}$, PCP, $\overline{\text{PCP}}$, CFG Äquivalenz, $\mathcal{V}_f$, $\overline{\mathcal{V}_f}$ und „alle Sprachen $L \in P$“

$F_{\mathcal{V}_f} := \{ g \text{ berechenbar} \mid g = f \} \;\|\; = \{ f(1) \}$

$\rightarrow f \in F_{\mathcal{V}_f}$

$\rightarrow g(x) = f(x) + 1 \rightsquigarrow g \notin F_{\mathcal{V}_f}$

[1] Eine Sprache L ist *co-semi-entscheidbar*, wenn es eine TM gibt, die 0 ausgibt wenn die Eingabe nicht in L ist und nicht terminiert wenn die Eingabe in L ist.

[2] Eine Sprache ist *nulli-entscheidbar*, wenn sie nicht semi-entscheidbar ist und nicht co-semi-entscheidbar ist.

$\mathcal{H} = \{ w \# x \mid M_w[x]{\downarrow} \}$

$\mathcal{H}_{\Sigma^*} = \{ w \in \{0,1\}^* \mid \forall x \in \Sigma^*.\ M_w[x]{\downarrow} \}$

Wenn $x \notin A$ / Wenn $x \in A$	terminiert (0)	terminiert nicht ($\perp$)
terminiert (1)	Beispiele: $\{x \in \mathbb{N} \mid f(x)\}$, „alle Sprachen $L \in P$“ entscheidbar	Beispiele: H, H_0, K, PCP, $\overline{V_f}$ semi-entscheidbar rekursiv aufzählbar
terminiert nicht ($\perp$)	co-semi-entscheidbar ~~Beispiele~~: H, H_0, $\overline{K}$, $\overline{PCP}$, V_f, CFG $Ä_a$	nulli-entscheidbar Beispiele: H_{Σ^*}, $\overline{H_{\Sigma^*}}$, H_{NEA}, $\overline{H_{NEA}}$

unentscheidbar

Übungsaufgabe Ü11.5. (*Reductio ad absurdum*)

Sei $A := \{w \in \Sigma^* \mid \exists i \in \mathbb{N}_0.\ |w| = 5i + 3\}$ mit $\Sigma = \{a, b\}$. Erklären Sie, warum die angegebenen Funktionen keine Reduktionen gemäß Vorlesungsdefinition sind.

s.e. für A

(a) Behauptung: $\mathcal{H}_0 \leq A$

Reduktion: Definiere $f : \mathcal{H}_0 \to A$ mit $f(w) := aaa$.

(b) Behauptung: $\mathcal{H}_0 \leq A$

Reduktion:

$$f(w) = \begin{cases} aaa & \text{falls } w \in \mathcal{H}_0 \\ b & \text{sonst} \end{cases}$$

w → Ent. für A → Ja → Ja; Nein

(c) Behauptung: $A \leq \mathcal{H}_0$

Reduktion: f bildet jedes Element $x \in \Sigma^*$ auf die Kodierung einer TM M_x ab, die wie folgt definiert ist: Die TM M_x löscht die Eingabe und schreibt x aufs Band, bestimmt dann die Länge von x, zieht 3 ab und prüft anschließend, ob das Ergebnis durch 5 teilbar ist. Dementsprechend gibt die Maschine „Ja“(1) oder „Nein“(0) aus.

(d) Behauptung: $\overline{\mathcal{H}_0} \leq \mathcal{H}_0$

Reduktion: f bildet jedes $w \in \{0,1\}^*$ auf die Kodierung $f(w)$ einer TM $M_{f(w)}$ ab, die $M_w[\epsilon]$ simuliert. Falls $M_w[\epsilon]$ hält, geht $M_{f(w)}$ in eine Endlosschleife. Falls $M_w[\epsilon]$ nicht hält, hält $M_{f(w)}$.

(e) Behauptung: $\mathcal{H}_{\Sigma^*} \leq \mathcal{H}_0$ mit $\mathcal{H}_{\Sigma^*} := \{w \in \{0,1\}^* \mid \forall x \in \Sigma^*.\ M_w[x]\downarrow\}$.

Reduktion: f bildet jedes $w \in \{0,1\}^*$ auf die Kodierung $f(w)$ einer TM $M_{f(w)}$ ab, die erst die Eingabe löscht, dann nichtdeterministisch $x \in \Sigma^*$ erzeugt und dann $M_w[x]$ simuliert.

Wenn $x \notin A$ / Wenn $x \in A$	terminiert (0)	terminiert nicht ($\perp$)
terminiert (1)	Beispiele:	Beispiele:
terminiert nicht ($\perp$)	Beispiele:	Beispiele:

Übungsaufgabe Ü11.5. (*Reductio ad absurdum*)

Sei $A := \{w \in \Sigma^* \mid \exists i \in \mathbb{N}_0.\ |w| = 5i + 3\}$ mit $\Sigma = \{a, b\}$. Erklären Sie, warum die angegebenen Funktionen keine Reduktionen gemäß Vorlesungsdefinition sind.

(a) Behauptung: $\mathcal{H}_0 \leq A$

Reduktion: Definiere $f : \mathcal{H}_0 \to A$ mit $f(w) := aaa$. nicht total

(b) Behauptung: $\mathcal{H}_0 \leq A$

Reduktion:

$$f(w) = \begin{cases} aaa & \text{falls } w \in \mathcal{H}_0 \\ b & \text{sonst} \end{cases}$$

total, nicht berechenbar (da $w \in \mathcal{H}_0$? unentscheidbar)

$ab \notin A$, aber $f(ab) \in \mathcal{H}_0$

(c) Behauptung: $A \leq \mathcal{H}_0$ total berechenbar

Reduktion: f bildet jedes Element $x \in \Sigma^*$ auf die Kodierung einer TM M_x ab, die wie folgt definiert ist: Die TM M_x löscht die Eingabe und schreibt x aufs Band, bestimmt dann die Länge von x, zieht 3 ab und prüft anschließend, ob das Ergebnis durch 5 teilbar ist. Dementsprechend gibt die Maschine „Ja"(1) oder „Nein"(0) aus.

(d) Behauptung: $\overline{\mathcal{H}_0} \leq \mathcal{H}_0$

Reduktion: f bildet jedes $w \in \{0,1\}^*$ (total) auf die Kodierung $f(w)$ einer TM $M_{f(w)}$ ab, die $M_w[\epsilon]$ simuliert. Falls $M_w[\epsilon]$ hält, geht $M_{f(w)}$ in eine Endlosschleife. Falls $M_w[\epsilon]$ nicht hält, hält $M_{f(w)}$. $\to$ unmöglich; f nicht wohldefiniert

(e) Behauptung: $\mathcal{H}_{\Sigma^*} \leq \mathcal{H}_0$ mit $\mathcal{H}_{\Sigma^*} := \{w \in \{0,1\}^* \mid \forall x \in \Sigma^*.\ M_w[x]\downarrow\}$.

Reduktion: f bildet jedes $w \in \{0,1\}^*$ (total) auf die Kodierung $f(w)$ einer TM $M_{f(w)}$ ab, die erst die Eingabe löscht, dann nichtdeterministisch $x \in \Sigma^*$ erzeugt und dann $M_w[x]$ simuliert.

Können wir $w \notin \mathcal{H}_{\Sigma^*}$ finden,

aber $f(w) \in \mathcal{H}_0$

Sei w eine TM mit $M_w[\epsilon]\downarrow$ und $M_w[1]\uparrow$.

Dann rät die TM $M_{f(w)}$ $x = \epsilon$.

Somit gilt $M_{f(w)}[\epsilon]\downarrow \iff M_w[x]\downarrow \iff M_w[\epsilon]\downarrow$ ✓

$\to$ $f(w) \in \mathcal{H}_0$

$f: \Sigma^* \to \Gamma^*$ ist eine Reduktion von A auf B gdw.

$A \subseteq \Sigma^*$, $B \subseteq \Gamma^*$

1. f total
2. f berechenbar
3. $w \in A \iff f(w) \in B$

$$\left.\begin{array}{ll} \text{i)} & w \in A, \quad f(w) \notin B \\ \text{ii)} & w \notin A, \quad f(w) \in B \end{array}\right\} \text{Violation}$$