

Übung 11 zur Vorlesung Formale Sprachen und Komplexität

Hinweis:

Die letzte Aufgabe auf diesem Blatt ist eine Aufgabe zur Klausurvorbereitung. Diese Aufgabe orientiert sich in Form und inhaltlichen Schwerpunkten an den Klausuraufgaben. Dies bedeutet jedoch nicht, dass die anderen Aufgaben nicht klausurrelevant sind.

Die Lösungen der Klausurvorbereitungs-Aufgaben werden am Ende der Bearbeitungszeit gesondert veröffentlicht, aber **nicht** im Tutorium besprochen. Die Lösungen der anderen Aufgaben werden bereits zu Beginn der Bearbeitungszeit veröffentlicht und können Ihnen bei der Bearbeitung helfen.

Wenn Sie Ihre Lösung innerhalb der Bearbeitungszeit über Moodle abgeben, erhalten Sie eine individuelle Korrektur. Die Abgabe ist freiwillig (aber nachdrücklich empfohlen).

FSK11-1 Satz von Rice

Sei $M = (Z, \{a, b\}, \Gamma, \delta, z_0, \square, E)$ eine deterministische Turingmaschine. Welche der folgenden Fragestellungen zu M sind entscheidbar?

Zeigen Sie die Unentscheidbarkeit unentscheidbarer Fragestellungen mittels des Satzes von Rice. Bei entscheidbaren Fragestellungen erläutern Sie, wie die charakteristische Funktion berechnet wird.

- a) Die Funktion die von M berechnet wird hat mindestens 3 Fixpunkte. Ein Fixpunkt einer Funktion $f : D \rightarrow D$ ist ein $x \in D$, sodass $f(x) = x$.
- b) Totalitätsproblem: Es gibt ein Wort $w \notin L(M)$.
- c) Hat M einen Müllzustand? Formal definieren wir einen Müllzustand hier als einen Zustand z , der kein Endzustand ist und für den gilt $\delta(z, a) \subseteq \{z\}$ für alle a .
- d) M terminiert auf jeder Eingabe nach zwischen 50 und 55 Schritten.

FSK11-2 Beweise prüfen

In den folgenden Teilaufgaben betrachten wir jeweils einen Beweis, der einen Fehler enthält. Identifizieren Sie diesen Fehler (mit kurzer Begründung).

- a) Beweisen oder widerlegen Sie: Die Sprache

$$D = \{w \in \{0,1\}^* \mid M_w \text{ akzeptiert } w \text{ nicht}\}$$

ist semi-entscheidbar.

Beweis:

D ist semi-entscheidbar. Um das zu zeigen, konstruieren wir eine DTM M , die D semi-entscheidet. Das heißt, dass M für alle Eingaben $w \in D$ hält und für alle Eingaben $w \notin D$ nicht hält.

Angenommen, es gäbe so eine Turingmaschine M . Betrachte ein Wort $w \in \{0,1\}^*$.

- Wenn $w \in D$ ist, dann akzeptiert M_w die Eingabe w . Somit akzeptiert auch M das Wort w .
- Wenn $w \notin D$ ist, dann hält M_w mit Eingabe w nicht. Somit akzeptiert auch M das Wort w nicht.

M semi-entscheidet also D .

- b) Sei $L_u = \{w\#x \mid w, x \in \{0,1\}^* \text{ und } x \in L(M_w)\}$. Diese Sprache ist semi-entscheidbar, aber nicht entscheidbar.

Zeigen Sie: Die Sprache $L_r = \{w \in \{0,1\}^* \mid L(M_w) \text{ ist regulär}\}$ ist unentscheidbar.

Beweis:

Wir reduzieren L_u auf L_r . Da L_u unentscheidbar ist, folgt daraus, dass L_r unentscheidbar ist.

Sei $v \in \{0,1,\#\}^*$. Wir definieren die Reduktionsfunktion f durch

$$f(v) = \begin{cases} w_{M_3} & \text{falls } v = w\#x \text{ und } M_w \text{ akzeptiert } x \\ w_{M_4} & \text{sonst} \end{cases}$$

Dabei ist M_3 eine Turingmaschine, die die reguläre Sprache $\{0,1\}^*$ akzeptiert. M_4 ist eine Turingmaschine, die die nichtreguläre Sprache $\{0^n 1^n \mid n \in \mathbb{N}\}$ akzeptiert. w_{M_i} ist die Binärkodierung der jeweiligen Turingmaschine.

M_3 , M_4 und die Binärkodierungen sind offensichtlich berechenbar, also ist f berechenbar (und offensichtlich total). Weiterhin gilt:

$$\begin{array}{ll} v \in L_u & \\ \text{g.d.w. } v = w\#x \text{ und } x \in L(M_w) & \\ \text{g.d.w. } M_{f(v)} \text{ akzeptiert eine reguläre Sprache} & \\ \text{g.d.w. } f(v) \in L_r & \end{array}$$

Somit ist f eine valide Reduktionsfunktion und $L_u \leq L_r$.

FSK11-3 PCP und DFA

Betrachten Sie das folgende Verfahren V :

V berechnet für eine PCP-Instanz K über einem Alphabet Σ (d.h. $K = ((x_1, y_1), \dots, (x_k, y_k))$ mit $x_i, y_i \in \Sigma^+$) und einen deterministischen endlichen Automaten $A = (Z, \Sigma, \delta, z_0, E)$ einen Wahrheitswert.

Dafür wird zunächst die Menge $M \subseteq Z \times Z$ folgendermaßen iterativ bestimmt:

- Anfangs ist $M = \emptyset$
- In jedem Schritt wird für alle $(z_i, z_j) \in M$, sowie für (z_0, z_0) (selbst, wenn es nicht in M enthalten ist) und jedes Wortpaar/Spielstein $T = (x_l, y_l)$ aus K bestimmt, welcher Zustand $z_{i'}$ von z_i aus mit dem oberen Wort von T erreicht wird (d.h. $z_{i'} = \hat{\delta}(z_i, x_l)$) und welcher Zustand $z_{j'}$ von z_j aus mit dem unteren Wort von T erreicht wird (d.h. $z_{j'} = \hat{\delta}(z_j, y_l)$).

Das Tupel $(z_{i'}, z_{j'})$ wird zu M hinzugefügt, wenn es nicht bereits in M vorhanden ist.

- Wenn sich die Menge M nicht mehr ändert, ist dies die engültige Gestalt von M und die Berechnung von M ist beendet.

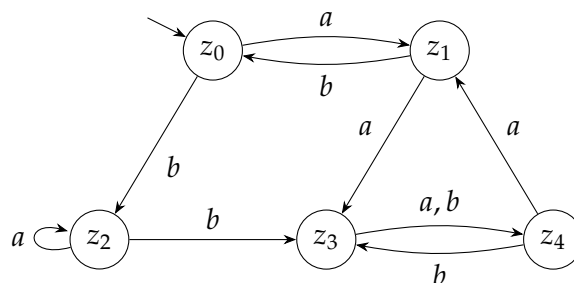
Gibt es nun einen Zustand $z \in Z$ mit $(z, z) \in M$, so wird „wahr“ zurückgegeben, ansonsten „falsch“.

Hinweis: Die Endzustandsmenge E wird vom Verfahren V nicht verwendet.

a) Berechnen Sie die Menge M für die PCP-Instanz K :

$$\left\{ \begin{bmatrix} bba \\ bb \end{bmatrix}, \begin{bmatrix} abba \\ abbab \end{bmatrix}, \begin{bmatrix} aba \\ bab \end{bmatrix}, \begin{bmatrix} abaaba \\ bbb \end{bmatrix}, \begin{bmatrix} ba \\ bbaabbab \end{bmatrix} \right\}$$

und den Automaten A :



Was gibt das Verfahren zurück?

- b) Wenn V auf einer PCP-Instanz K gelaufen ist, was kann man dann aus einer „wahr“- und was aus einer „falsch“-Antwort für die PCP-Instanz schlussfolgern?

Bestimmen Sie mit Begründung getrennt für den „wahr“- und „falsch“-Fall die richtige Antwortkategorie:

- K hat sicher eine Lösung
 - K hat sicher keine Lösung
 - Man kann damit noch nicht sicher sagen, ob K eine Lösung hat
- c) Zeigen Sie, dass es PCP-Instanzen K gibt, die keine Lösung haben, bei denen es aber dennoch keinen Automaten A gibt, sodass $V(K, A)$ mit „falsch“ antwortet.

Hinweis: Die Unentscheidbarkeit von PCP kann Ihnen beim Beweis helfen

Klausurvorbereitung FSK-11-K

- a) Wir betrachten das EVENPCP-Problem, eine Variante von LPCP, bei der die Wörter auf den Spielsteinen gerade Länge haben müssen. Eine Instanz von EVENPCP ist also eine endliche Folge von Paaren $(x_1, y_1), \dots, (x_n, y_n)$ mit $x_i, y_i \in \{w \mid w \in \Sigma^* \text{ und } |w| \text{ gerade}\}$ für $i = 1, \dots, n$. Beispielsweise ist $(ab, aaba)$ ein erlaubter Spielstein; (ab, aab) aber nicht.

Zeigen Sie durch Reduktion von PCP auf EVENPCP, dass EVENPCP unentscheidbar ist.

- b) Sei $A = \{w \in \{0,1\}^* \mid M_w \text{ hält bei Eingabe 36 mit Ausgabe 42 an}\}$. Der folgende Beweis zur Unentscheidbarkeit von A ist falsch. Lokalisieren Sie den Fehler und begründen Sie, wieso der Beweis falsch ist.

Beweis:

Wir zeigen $H_0 \leq A$. Da H_0 unentscheidbar ist, folgt daraus, dass auch A unentscheidbar ist.

Wir definieren die Reduktionsfunktion $f : \{0,1\}^* \rightarrow \{0,1\}^*$ wie folgt. Für $w \in \{0,1\}^*$ berechnet f zunächst die Turingmaschine M_w , erstellt daraus eine Turingmaschine T und berechnet anschließend deren Binärcodierung w_T . Dabei verhält sich T wie folgt:

- Prüfe, ob auf dem Band die Zahl 36 (in Binärdarstellung) steht. Falls nein, gehe in eine Endlosschleife über.
- Führe M_w aus.
- Falls M_w anhält, schreibe die Zahl 42 (in Binärdarstellung) auf das Band und akzeptiere.

Die Funktion f ist offensichtlich total. Sie ist auch berechenbar, da T konstruierbar ist. Weiterhin gilt:

- $w \in H_0$
- g.d.w. M_w hält auf leerem Band
- g.d.w. $M_{f(w)}$ hält für Eingabe 36 mit Ausgabe 42
- g.d.w. $f(w) \in A$

Somit ist f eine valide Reduktionsfunktion und $H_0 \leq A$.