

Lösungsvorschlag zur Übung 10 zur Vorlesung Formale Sprachen und Komplexität

FSK10-1 PCP-Varianten

- a) Wir betrachten das PCP4-Problem, eine Variante von PCP, bei der die Spielsteine aus vier Wörtern bestehen. Eine Instanz von PCP4 ist also eine endliche Folge von 4-Tupeln $(x_1, y_1, z_1, u_1), \dots, (x_n, y_n, z_n, u_n)$ mit $x_i, y_i, z_i, u_i \in \Sigma^+$ für $i = 1, \dots, n$. Eine Lösung der Instanz K ist ähnlich zu PCP eine endliche Folge von Indizes $i_1, \dots, i_m \in \mathbb{N}$, sodass $x_{i_1} \cdots x_{i_m} = y_{i_1} \cdots y_{i_m} = z_{i_1} \cdots z_{i_m} = u_{i_1} \cdots u_{i_m}$. Sind die folgenden Instanzen K_1, K_2 von PCP4 lösbar? Wenn ja, geben Sie eine Lösung (also eine geeignete Folge von Indizes) an. Wenn nein, beweisen Sie, dass die Instanz keine Lösung hat.

$$K_1 = \left(\begin{bmatrix} bbc \\ bca \\ cab \\ bbca \end{bmatrix}, \begin{bmatrix} a \\ ab \\ abb \\ a \end{bmatrix}, \begin{bmatrix} cca \\ ca \\ a \\ ca \end{bmatrix}, \begin{bmatrix} abc \\ bcc \\ ccc \\ bcc \end{bmatrix} \right)$$

$$K_2 = \left(\begin{bmatrix} c \\ b \\ bab \\ cc \end{bmatrix}, \begin{bmatrix} acb \\ baa \\ aaa \\ acba \end{bmatrix}, \begin{bmatrix} bca \\ caaa \\ b \\ c \end{bmatrix}, \begin{bmatrix} b \\ ba \\ b \\ bab \end{bmatrix} \right)$$

LÖSUNGSVORSCHLAG:

K_1 hat die Lösung 2, 1, 4, 3, also

$$\begin{bmatrix} a \\ ab \\ abb \\ a \end{bmatrix}, \begin{bmatrix} bbc \\ bca \\ cab \\ bbca \end{bmatrix}, \begin{bmatrix} abc \\ bcc \\ ccc \\ bcc \end{bmatrix}, \begin{bmatrix} cca \\ ca \\ a \\ ca \end{bmatrix}$$

K_2 hat keine Lösung: Mit den Spielsteinen 1, 2 und 3 kann man nicht beginnen, da alle mindestens ein Wort haben das bereits im ersten Symbol nicht mit den anderen übereinstimmt. Betrachten wir also den Fall, dass wir mit

dem Spielstein $\begin{bmatrix} b \\ ba \\ b \\ bab \end{bmatrix}$ anfangen. Wenn wir mit $\begin{bmatrix} c \\ b \\ bab \\ cc \end{bmatrix}$ oder $\begin{bmatrix} acb \\ baa \\ aaa \\ acba \end{bmatrix}$ weiter machen wollen, passen die ersten beiden Wörter nicht zusammen, da bc kein Präfix von bab bzw. ba kein Präfix von $baaa$ ist, also können diese keine Weiterführungen sein. Mit $\begin{bmatrix} bca \\ caaa \\ b \\ c \end{bmatrix}$ haben wir allerdings in den letzten beiden Wörtern einen Unterschied, da bb kein Präfix von bab ist. Somit gibt es keine mögliche Weiterführung und da $\begin{bmatrix} b \\ ba \\ b \\ bab \end{bmatrix}$ auch alleine noch keine Lösung ist, kann K_2 keine Lösung haben.

- b) Zeigen Sie durch Reduktion von PCP auf PCP4, dass PCP4 unentscheidbar ist.

LÖSUNGSVORSCHLAG:

Wir zeigen $\text{PCP} \leq \text{PCP4}$. Da PCP unentscheidbar ist, folgt daraus, dass auch PCP4 unentscheidbar ist.

Sei $K = ((x_1, y_1), \dots, (x_n, y_n))$ eine Instanz von PCP mit Alphabet Σ . Wir definieren $f(K) = f((x_1, y_1), \dots, f((x_n, y_n)))$ und $f((x_i, y_i)) = (x_i, y_i, x_i, x_i)$. Damit ist $f(K)$ auf jeden Fall eine Instanz von PCP4 und f ist offensichtlich total und berechenbar.

Es gilt somit:

- Wenn $K \in \text{PCP}$ ist, dann hat K eine Lösung $i_1, \dots, i_m$ mit $x_{i_1} \cdots x_{i_m} = y_{i_1} \cdots y_{i_m}$. Da $x_{i_1} \cdots x_{i_m} = y_{i_1} \cdots y_{i_m} = x_{i_1} \cdots x_{i_m} = x_{i_1} \cdots x_{i_m}$, ist $i_1, \dots, i_m$ auch eine Lösung für $f(K)$, somit $f(K) \in \text{PCP4}$.
- Wenn $f(K) \in \text{PCP4}$ ist, dann hat $f(K)$ eine Lösung $i_1, \dots, i_m$ mit $x_{i_1} \cdots x_{i_m} = y_{i_1} \cdots y_{i_m} = x_{i_1} \cdots x_{i_m} = x_{i_1} \cdots x_{i_m}$. Somit gilt auch bereits $x_{i_1} \cdots x_{i_m} = y_{i_1} \cdots y_{i_m}$ und $i_1, \dots, i_m$ ist auch eine Lösung für K und $K \in \text{PCP}$.

- c) Wir betrachten das 456PCP-Problem, eine Variante von PCP, bei der die 'Spielsteine' auf das Alphabet $\Sigma = \{4, 5, 6\}$ beschränkt sind. Eine Instanz von 456PCP ist also eine endliche Folge von Paaren $(x_1, y_1), \dots, (x_n, y_n)$ mit $x_i, y_i \in \{4, 5, 6\}^+$ für $i = 1, \dots, n$. Eine Lösung der Instanz K ist wie bei PCP eine endliche Folge von Indices $i_1, \dots, i_m \in \mathbb{N}$, sodass $x_{i_1} \cdots x_{i_m} = y_{i_1} \cdots y_{i_m}$.

Zeigen Sie durch Reduktion von PCP auf 456PCP, dass 456PCP unentscheidbar ist.

LÖSUNGSVORSCHLAG:

Wir zeigen $PCP \leq 456PCP$. Da PCP unentscheidbar ist, ist damit auch 456PCP unentscheidbar.

Sei $K = ((x_1, y_1), \dots, (x_n, y_n))$ eine Instanz von PCP mit Alphabet $\Sigma = \{a_1, \dots, a_j\}$. Wir definieren $f(a_i) = 45^i$ für die Elemente aus Σ und $f(\varepsilon) = \varepsilon$, $f(a_i w) = f(a_i) f(w)$ für Wörter aus Σ^* , sowie schließlich $f(K) = (f(x_1), f(y_1)), \dots, (f(x_n), f(y_n))$ für Instanzen. Da 45^i ein Wort aus $\{4, 5, 6\}^+$ darstellt, überführt f Instanzen von PCP in Instanzen von 456PCP. Damit ist f auch offensichtlich total und berechenbar.

Weiter gilt:

- Wenn $K \in PCP$ ist, dann hat K eine Lösung $i_1, \dots, i_m$ mit $x_{i_1} \cdots x_{i_m} = y_{i_1} \cdots y_{i_m}$. Dann ist

$$f(x_{i_1}) \cdots f(x_{i_m}) = f(x_{i_1} \cdots x_{i_m}) = f(y_{i_1} \cdots y_{i_m}) = f(y_{i_1}) \cdots f(y_{i_m})$$

und es ist also $i_1, \dots, i_m$ eine Lösung für $f(K)$, somit $f(K) \in 456PCP$.

- Wenn $f(K) \in 456PCP$ ist, dann hat $f(K)$ eine Lösung $i_1, \dots, i_m$ mit $f(x_{i_1}) \cdots f(x_{i_m}) = f(y_{i_1}) \cdots f(y_{i_m})$. Die Zuordnung 45^i zu Buchstaben a_i ist eindeutig für ein gegebenes Alphabet Σ , damit können wir eine Funktion g angeben die sich invers zu f verhält. Es gilt also

$$\begin{aligned} x_{i_1} \cdots x_{i_m} &= g(f(x_{i_1} \cdots x_{i_m})) = g(f(x_{i_1}) \cdots f(x_{i_m})) = \\ &= g(f(y_{i_1}) \cdots f(y_{i_m})) = g(f(y_{i_1} \cdots y_{i_m})) = y_{i_1} \cdots y_{i_m} \end{aligned}$$

Somit ist $i_1, \dots, i_m$ auch eine Lösung für $K \in PCP$.

- d) Für Mengen Σ und Δ nennen wir eine Funktion $f : \Sigma^* \rightarrow \Delta^*$ einen *Homomorphismus* (siehe auch Aufgabe FSK6-4), wenn gilt:

$$\begin{aligned} f(\varepsilon) &= \varepsilon \\ f(u \circ v) &= f(u) \circ f(v) \quad \forall u, v \in \Sigma^* \end{aligned}$$

Wir definieren das Problem HOMPCP. Eine Instanz dieses Problem ist ein 4-Tupel (Σ, Δ, f, g) , wobei Σ und Δ endliche Mengen sind und $f, g : \Sigma^* \rightarrow \Delta^*$ Homomorphismen. Eine Lösung der Instanz ist ein Wort $w \in \Sigma^+$ sodass gilt: $f(w) = g(w)$.

Wir betrachten außerdem das LPCP-Problem, eine Variante von PCP, bei der die ‚Spielsteine‘ auch das leere Wort enthalten können. Eine Instanz von LPCP mit Alphabet Σ ist also eine endliche Folge von Paaren $(x_1, y_1), \dots, (x_n, y_n)$ mit $x_i, y_i \in \Sigma^*$.

Σ^* für $i = 1, \dots, n$ (wohingegen bei PCP gilt: $x_i, y_i \in \Sigma^+$). Eine Lösung der Instanz K ist wie bei PCP eine endliche Folge von Indices $i_1, \dots, i_m \in \mathbb{N}$ sodass $x_{i_1} \cdots x_{i_m} = y_{i_1} \cdots y_{i_m}$. Das LPCP-Problem ist unentscheidbar.

Zeigen Sie durch Reduktion von LPCP auf HOMPCP, dass HOMPCP für $|\Delta| \geq 2$ unentscheidbar ist.

LÖSUNGSVORSCHLAG:

Wir zeigen $\text{LPCP} \leq \text{HOMPCP}$. Da LPCP unentscheidbar ist, folgt daraus, dass auch HOMPCP unentscheidbar ist.

Sei $K = ((x_1, y_1), \dots, (x_n, y_n))$ eine Instanz von LPCP mit Alphabet Γ . Wir definieren eine Instanz $F(K) = (\Sigma, \Delta, f, g)$ von HOMPCP wie folgt:

$$\begin{aligned}\Sigma &= \{1, \dots, n\} \\ \Delta &= \Gamma \\ f(i) &= \begin{cases} \varepsilon & \text{für } i = \varepsilon \\ x_i & \text{für } i \in \Sigma \\ f(j) \circ f(k) & \text{für } i = j \circ k; j, k \in \Sigma^+ \end{cases} \\ g(i) &= \begin{cases} \varepsilon & \text{für } i = \varepsilon \\ y_i & \text{für } i \in \Sigma \\ g(j) \circ g(k) & \text{für } i = j \circ k; j, k \in \Sigma^+ \end{cases}\end{aligned}$$

Wörter aus Σ^+ sind dabei endliche, nicht leere Folgen $i_1 \cdots i_m$ von Indizes aus der Indexmenge Σ . Offensichtlich sind f und g per Definition Homomorphismen.

Nun gilt:

- Eine Lösung für K ist eine endliche, nicht leere Folge von Indizes $i_1, \dots, i_m \in \Sigma$. Sei $w = i_1 \cdots i_m$. Dann gilt:

$$f(w) = f(i_1) \cdots f(i_m) = x_{i_1} \cdots x_{i_m} = y_{i_1} \cdots y_{i_m} = g(i_1) \cdots g(i_m) = g(w)$$

Somit ist w eine Lösung von $F(K)$.

- Umgekehrt ist eine Lösung von $F(K)$ ein nicht leeres Wort $w = i_1 \cdots i_m$ sodass

$$x_{i_1} \cdots x_{i_m} = f(i_1) \cdots f(i_m) = f(w) = g(w) = g(i_1) \cdots g(i_m) = y_{i_1} \cdots y_{i_m}$$

Somit ist $i_1, \dots, i_m$ eine Lösung von K .

F ist außerdem total und berechenbar und somit eine valide Reduktionsfunktion.

FSK10-2 Entscheidbarkeit

Prüfen Sie, ob die folgenden Behauptungen wahr oder falsch sind. Begründen Sie Ihre Antworten wie folgt: Wenn eine Sprache L (semi-)entscheidbar ist, beschreiben Sie die Funktionsweise einer deterministischen Turingmaschine, die die charakteristische Funktion χ_L bzw. χ'_L berechnet. Wenn L nicht (semi-)entscheidbar ist, leiten Sie einen Widerspruch ab.

- a) Wenn A und B entscheidbare Sprachen sind, dann ist $A \cap B$ entscheidbar.

LÖSUNGSVORSCHLAG:

Wahr. Sei T_A eine DTM, die A entscheidet, und T_B eine DTM, die B entscheidet. Konstruiere eine DTM für $A \cap B$, die sich wie folgt verhält: Gegeben x , berechne $T_A(x)$. Falls $T_A(x) = 0$, lehne x ab, ansonsten berechne $T_B(x)$. Gilt $T_B(x) = 0$, lehne x ab, sonst akzeptiere x . Da T_A, T_B die jeweiligen Mengen entscheiden, terminieren beide DTM immer, womit auch die DTM zu $A \cap B$ stets mit dem korrekten Ergebnis terminiert. Somit ist $A \cap B$ entscheidbar.

- b) Wenn A und $A \cup B$ entscheidbar sind, dann ist B entscheidbar.

LÖSUNGSVORSCHLAG:

Falsch. Sei $A = \{0,1\}^*$ und $B = H_0 \subseteq \{0,1\}^*$. Dann sind A und $A \cup B = A$ entscheidbar, aber B nicht.

- c) Das Problem, ob $L(M) \neq \emptyset$ für eine gegebene Turingmaschine M gilt, ist semi-entscheidbar.

LÖSUNGSVORSCHLAG:

Wahr. Algorithmus: Für $i = 0, 1, \dots$ schreibe nacheinander alle Wörter $w \in \Sigma^*$ mit $|w| \leq i$ auf das Band und simuliere M auf jedem w für i Schritte. Falls M in i Schritten akzeptiert, gib 1 aus. Ist $L(M) \neq \emptyset$, so testen wir M irgendwann für ausreichend viele Schritte auf Wörtern ausreichender Länge, um ein Wort aus $L(M)$ zu finden.

FSK10-3 Halteprobleme

- a) Betrachten Sie die Sprache

$$L_1 = \{w_M \in \{0,1\}^* \mid \text{TM } M \text{ hält für Eingabe } 01\}$$

wobei w_M das Encoding für M ist.

Welche der folgenden Aussagen ist korrekt? Beweisen Sie Ihre Antwort.

- L_1 ist entscheidbar.
- L_1 ist semi-entscheidbar, aber nicht entscheidbar.
- L_1 ist weder entscheidbar noch semi-entscheidbar.

Um zu zeigen, dass L_1 entscheidbar (bzw. semi-entscheidbar) ist, beschreiben Sie kurz die Funktionsweise einer deterministischen Turingmaschine, die L_1 (semi-)entscheidet. Um zu zeigen, dass L_1 nicht (semi-)entscheidbar ist, reduzieren Sie ein geeignetes Problem auf L_1 .

LÖSUNGSVORSCHLAG:

L_1 ist semi-entscheidbar: Für eine Eingabe z simulieren wir die Maschine M_z mit Eingabe 01. Wenn die Simulation hält, geben wir 1 aus.

L_1 ist nicht entscheidbar. Wir beweisen das durch Reduktion von H_0 auf L_1 .

Um $H_0 \leq L_1$ zu zeigen, müssen wir eine totale, berechenbare Funktion $f: \{0,1\}^* \rightarrow \{0,1\}^*$ angeben, sodass für alle z gilt: $z \in H_0 \iff f(z) \in L_1$. Wir wählen

$$f(z) = \begin{cases} w_{M'} & \text{falls } z \text{ ein valides Encoding für eine Turingmaschine } M \text{ ist.} \\ z & \text{sonst} \end{cases}$$

Dabei ist

- M' eine Turingmaschine, die zunächst ihre Eingabe löscht und dann M ausführt.

Die Funktion f ist offensichtlich total. Sie ist auch berechenbar, denn:

- Wir können Turingmaschinen binär kodieren und dekodieren (und erkennen, ob ein binäres Wort eine Turingmaschine kodiert).
- Wir können M' aus M konstruieren.

Für M' gilt:

- Wenn M mit Eingabe ε hält, dann hält M' mit jeder beliebigen Eingabe und also insbesondere mit Eingabe 01.
- M' hält mit einer beliebigen Eingabe, und also insbesondere mit Eingabe 01, nur dann, wenn M mit Eingabe ε hält.

Um die Äquivalenz $z \in H_0 \iff f(z) \in L_1$ zu zeigen, beweisen wir die beiden Implikationen $z \in H_0 \Rightarrow f(z) \in L_1$ und $z \notin H_0 \Rightarrow f(z) \notin L_1$.

$$z \in H_0 \Rightarrow f(z) \in L_1:$$

$z \in H_0$
 $\Rightarrow z$ ist ein valides Encoding für M und M hält mit Eingabe ε
 $\Rightarrow f(z) = w_{M'}$ und M' hält mit jeder Eingabe (insbesondere 01)
 $\Rightarrow f(z) \in L_1$

$z \notin H_0 \Rightarrow f(z) \notin L_1$:

Fall 1: $z \notin H_0$
 z ein valides Encoding für M
 $\Rightarrow M$ hält nicht mit Eingabe ε
 $\Rightarrow f(z) = w_{M'}$ und M' hält auf keiner Eingabe (insbesondere nicht 01)
 $\Rightarrow f(z) \notin L_1$
 Fall 2: z kein valides Encoding
 $\Rightarrow f(z) = z$ ist kein valides Encoding
 $\Rightarrow f(z) \notin L_1$

Somit ist $H_0 \leq L_1$ und da H_0 unentscheidbar ist, ist auch L_1 unentscheidbar.

- b) Zeigen Sie, dass das folgende Problem für jede deterministische Turingmaschine M und natürliche Zahl n entscheidbar ist.

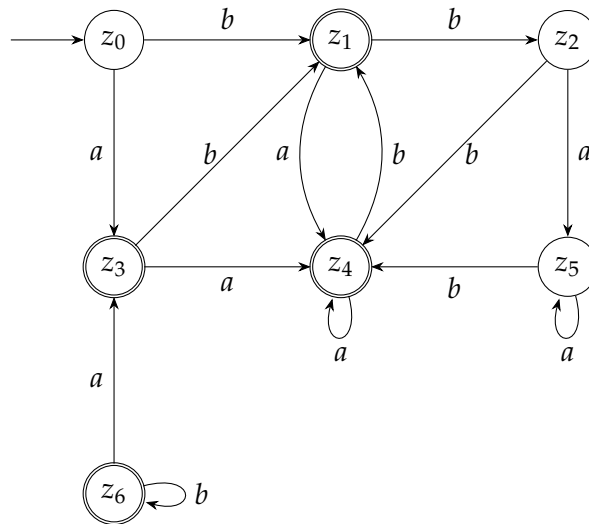
„ M hält auf jeder Eingabe nach höchstens n Schritten.“

LÖSUNGSVORSCHLAG:

Für jede Eingabe w können wir prüfen, ob M auf w in n Schritten hält, indem wir M für n Schritte simulieren. Weiterhin kann M in n Schritten nur höchstens n Zeichen lesen. Deshalb gilt für Wörter w mit $|w| > n$: Wenn M nicht in n Schritten auf $w[0] \dots w[n]$ hält, dann hält M auch nicht in n Schritten auf w . Somit genügt es, nur Wörter mit Länge höchstens n zu testen. Da das endlich viele Wörter sind, ist das Problem entscheidbar.

FSK10-4 Algorithmen Wiederholung

- a) Minimieren Sie den folgenden DFA. Verwenden Sie die tabellarische Variante des Algorithmus zur Minimierung von DFAs aus der Vorlesung (nicht die graphische Variante und nicht den Algorithmus von letztem Jahr). Geben Sie die Partitionstabelle und den minimalen DFA an.



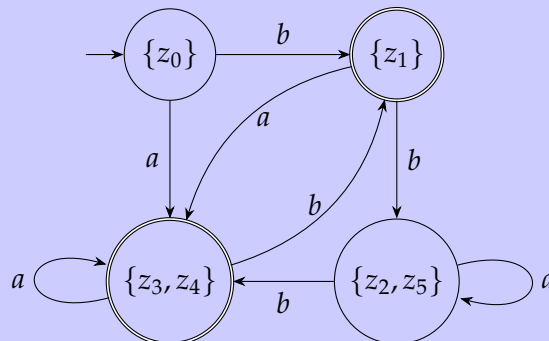
LÖSUNGSVORSCHLAG:

Wir entfernen den unerreichbaren Zustand z_6 .

Partitionstabelle:

z_0	z_2	z_5	z_1	z_3	z_4	
z_0	z_2	z_5	z_1	z_3	z_4	mit a
z_0	z_2	z_5	z_1	z_3	z_4	mit b

Aus der Partitionierung ergibt sich der Minimalautomat:



- b) Sei G die Grammatik $(\{A_1, A_2, A_3, A_4, A_5\}, \{\$, \#\}, P, A_1)$ mit

$$P = \{A_1 \rightarrow A_2A_5 \mid A_3A_5, \\ A_2 \rightarrow A_4A_3 \mid \$, \\ A_3 \rightarrow A_2A_5, \\ A_4 \rightarrow \$, \\ A_5 \rightarrow A_5A_1 \mid \#\}$$

Prüfen Sie mit dem CYK-Algorithmus, ob das Wort $\$ \# \# \$ \# \#$ in $L(G)$ ist. Erstellen

Sie dazu die entsprechende Tabelle des Algorithmus und erklären Sie anhand der Tabelle, ob das Wort in $L(G)$ ist.

LÖSUNGSVORSCHLAG:

Wort:	\$	#	#	\$	#	#
j \ i	1	2	3	4	5	6
1	A_2, A_4	A_5	A_5	A_2, A_4	A_5	A_5
2	A_1, A_3			A_1, A_3		
3	A_1		A_5	A_1		
4			A_5			
5	A_1					
6	A_1					

Da das Startsymbol A_1 in Zeile 6, Spalte 1 enthalten ist, ist $##### \in L(G)$.