

## Berechenbarkeit und Komplexität – Übung 4

Abgabe: bis Freitag, der 29. Mai 2026, um 11:00 Uhr am Fachgebiet, vor der Übung oder handschriftlich auf moodle.

**Geben Sie bitte Ihre Matrikelnummer an.  
Heften Sie zudem alle Ihre Lösungsblätter geeignet zusammen.**

### Bonusaufgaben

Bonusaufgaben können schriftlich bearbeitet werden. Ihre Lösungen geben Sie bitte bis zum oben angegebenen Termin ab. Die Abgaben werden von uns korrigiert und die erreichten Punkte werden bei der Bewertung der Prüfung als Bonuspunkte berücksichtigt.

#### Aufgabe 1\*

1+4+1 Punkte

Bearbeiten Sie in Anlehnung an das Lemma von Folie 6.20 die folgenden Teilaufgaben:

- (a) Bestimmen Sie  $(\text{bin}(1)\#\text{bin}(4))_5$ .

*Hinweis.* Es gilt  $(\square)_5 = 0$ ,  $(\text{bin}(1))_5 = 1$ ,  $(\text{bin}(0))_5 = 2$  und  $(\#)_5 = 3$ .

- (b) Geben Sie ein GOTO-Programm  $P$  an, welches die Funktion  $f : \mathbb{N}^2 \rightarrow \mathbb{N}$  mit

$$f(n_1, n_2) = (\text{bin}(n_1)\#\text{bin}(n_2))_5$$

berechnet (vgl. Folie 6.22).

*Hinweis.* Sie dürfen die Programmkonstrukte von Folien 1.26 und 1.27 verwenden. Außerdem können Sie in diesen Konstrukten auch natürliche Zahlen anstatt Variablen verwenden. (Zum Beispiel  $x_i := x_j + 2$ )

- (c) Berechnen Sie die eindeutige Zahl  $n \in \mathbb{N}$  mit  $(\text{bin}(n))_5 = 286$ .

#### Aufgabe 2\*

1+1+1+2 Punkte

Sei  $\Sigma = \{0, 1\}$ . Zeigen Sie die folgenden Aussagen:

- (a) Die Reduzierbarkeitsbeziehung  $\leq$  ist reflexiv und transitiv.  
(b) Seien  $A, B \subseteq \Sigma^*$  zwei entscheidbare Probleme mit  $\emptyset \neq B \neq \Sigma^*$ . Dann gilt  $A \leq B$ .  
(c) Seien  $A, B \subseteq \Sigma^*$  mit  $A \leq B$ . Dann gilt auch  $\Sigma^* \setminus A \leq \Sigma^* \setminus B$ .  
(d) Seien  $A, B \subseteq \Sigma^*$  und  $C = \{0\}A \cup \{1\}B$ . Dann gilt:

- (i)  $A, B \leq C$  und  
(ii)  $C \leq D$  für alle  $D \subseteq \Sigma^+$  mit  $A, B \leq D$ .

#### Aufgabe 3\*

2+2 Punkte

Sei  $\varphi : \Sigma_1^* \rightarrow \Sigma_2^*$  ein Homomorphismus bezüglich der Konkatenation. Das heißt es gilt:  $\varphi(w_1 \cdot w_2) = \varphi(w_1) \cdot \varphi(w_2)$  für alle  $w_1, w_2 \in \Sigma_1^*$ . Für eine Sprache  $L \subseteq \Sigma_1^*$  definieren wir die  $\varphi$ -Sprache von  $L$  als  $\varphi(L) = \{\varphi(w) \mid w \in L\} \subseteq \Sigma_2^*$ . Zeigen Sie:

- (a) Wenn  $\varphi(L)$  entscheidbar und  $\varphi$  injektiv ist, dann ist auch  $L$  entscheidbar.  
(b) Geben Sie einen Homomorphismus  $\varphi$  an, sodass  $\varphi(K)$  entscheidbar ist. (Dabei ist  $K$  das spezielle Halteproblem)

## Präsenzaufgaben

### Aufgabe 4

Geben Sie eine Turingmaschine an (formal als Tupel, vgl. Übung 3 Aufgabe 4), welche bei Eingabe von  $w \in \{0, 1\}^*$  das Wort  $w#w$  berechnet. Bei Eingaben andere Form, also wenn # als Teil der Eingabe auftritt, soll die Maschine (ohne Forderung an die Ausgabe) terminieren.

### Aufgabe 5

Ein 2-Keller-Automat ist ein Tupel  $M = (Z, \Sigma, \Gamma, \delta, z_0, \#, E)$ , wobei die einzelnen Komponenten die gleiche Bedeutung haben wie bei einem Kellerautomaten mit Endzuständen. Jedoch besitzt ein 2-Keller-Automat zwei Kellerspeicher, die in jedem Berechnungsschritt gelesen und aktualisiert werden. Somit hat die Transitionsfunktion die Gestalt

$$\delta: Z \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \times \Gamma \rightarrow \mathcal{P}_e(Z \times \Gamma^* \times \Gamma^*)$$

und eine Konfiguration von  $A$  ist beschrieben durch ein Tupel  $(z, w, \alpha_1, \alpha_2) \in Z \times \Sigma^* \times \Gamma^* \times \Gamma^*$ . Ein Konfigurationsübergang hat die Form

$$(z, aw, b_1\alpha_1, b_2\alpha_2) \vdash_M (z', w, \gamma_1\alpha_1, \gamma_2\alpha_2)$$

mit  $z, z' \in Z, a \in \Sigma \cup \{\varepsilon\}, w \in \Sigma^*, b_1, b_2 \in \Gamma, \alpha_1, \alpha_2, \gamma_1, \gamma_2 \in \Gamma^*$  und ist genau dann möglich, wenn  $(z', \gamma_1, \gamma_2) \in \delta(z, a, b_1, b_2)$  gilt. Eine Haltekonfiguration ist eine Konfiguration ohne Nachfolger und eine Konfiguration  $(z, w, \alpha_1, \alpha_2)$  ist akzeptierend, wenn  $z \in E$  und  $w = \varepsilon$  gilt.

Zeigen Sie, dass es zu jeder Turing-berechenbaren Funktion  $f: \Sigma^* \dashrightarrow \Sigma^*$  einen 2-Keller-Automaten gibt, der für jedes Wort  $w \in \Sigma^*$  von der initialen Konfiguration  $(z_0, w, \#, \#)$  genau dann eine akzeptierende Haltekonfiguration erreicht, wenn  $f(w)$  definiert ist und in diesem Fall der Inhalt des zweiten Kellers genau  $f(w)\#$  ist. Es genügt, wenn Sie das Verhalten des 2-Keller-Automaten für  $f$  beschreiben, ohne diesen explizit anzugeben.