

## Theoretische Informatik 2 Übungsblatt 4

Thomas Haas  
Prof. Dr. Roland Meyer

TU Braunschweig  
Sommersemester 2020

Ausgabe: 16.06.2020

Abgabe: 26.06.2020, 17:00

Geben Sie Ihre Lösungen bis Freitag, 26.06.2020 17:00 Uhr, per E-Mail an ihren Tutor ab.  
Fertigen Sie dazu ihre Hausaufgaben direkt in .pdf Form an oder scannen ihre handschriftlichen Hausaufgaben ein. Geben Sie in Gruppen von **4 Personen** ab.

### Aufgabe 1: PCP [9 Punkte]

Im folgenden betrachten Sie Abwandlungen des PCP-Problems.

- (a) [3 Punkte] Zeigen Sie, dass das 1-PCP, also das PCP für Instanzen, bei denen die  $x_i$  und  $y_i$  Wörter über dem unären Alphabet  $\{1\}$  sind, entscheidbar ist.
- (b) [3 Punkte] Es sei  $|\Sigma| \geq 2$ . Das  $\text{PCP}_{\geq k}$  ist folgendes Entscheidungsproblem.

$\text{PCP}_{\geq k}$

**Gegeben:** Eine endliche Menge von Tupeln aus Wörter  $(x_1, y_1), \dots, (x_m, y_m)$  über  $\Sigma^*$

**Entscheide:** Gibt es eine endliche, nicht-leere Sequenz von Indizes  $i_1, \dots, i_n$  mit  $n \geq k$  und  $x_{i_1}x_{i_2} \dots x_{i_n} = y_{i_1}y_{i_2} \dots y_{i_n}$ ?

Zeigen Sie, dass das  $\text{PCP}_{\geq k}$  unentscheidbar ist.

- (c) [3 Punkte] Es sei  $|\Sigma| \geq 2$ . Das Last-PCP ist folgendes Entscheidungsproblem.

Last-PCP

**Gegeben:** Eine endliche Menge von Tupeln aus Wörter  $(x_1, y_1), \dots, (x_m, y_m)$  über  $\Sigma^*$

**Entscheide:** Gibt es eine endliche, nicht-leere Sequenz von Indizes  $i_1, \dots, i_n$  mit  $i_n = m$  und  $x_{i_1}x_{i_2} \dots x_{i_n} = y_{i_1}y_{i_2} \dots y_{i_n}$ ?

Zeigen Sie, dass das Last-PCP unentscheidbar ist.

### Aufgabe 2: Der Satz von Rice [8 Punkte]

Wenden Sie den Satz von Rice, falls möglich, auf die folgenden Sprachen an. Begründen Sie jeweils, warum oder warum nicht der Satz angewendet werden kann.

- (a) [2 Punkte]  $\mathcal{L}_{Dec} = \{w \in \{0, 1\}^* \mid \mathcal{L}(M_w) \text{ ist entscheidbar.}\}$
- (b) [2 Punkte]  $\mathcal{L}_{Even} = \{w \in \{0, 1\}^* \mid M_w \text{ akzeptiert ein Wort gerader Länge.}\}$
- (c) [2 Punkte]  $\mathcal{L}_{N-S-Dec} = \{w \in \{0, 1\}^* \mid \mathcal{L}(M_w) \text{ ist nicht semi-entscheidbar.}\}$
- (d) [2 Punkte]  $\mathcal{L}_{Step} = \{w \in \{0, 1\}^* \mid M_w \text{ akzeptiert kein Wort in unter 100 Schritten.}\}$

### Aufgabe 3: Totalität [6 Punkte]

Betrachten Sie die Sprache  $\mathcal{L}_{total} = \{w \in \{0, 1\}^* \mid M_w \text{ ist ein Entscheider.}\}$ . Warum kann der Satz von Rice hier nicht angewendet werden? Was ist der Unterschied zu Aufgabenteil (a) in der vorherigen Aufgabe 2?

Zeigen Sie per Reduktion, dass  $\mathcal{L}_{total}$  weder semi-entscheidbar noch co-semi-entscheidbar ist.

**Hinweis.** Sie haben in der letzten Hausaufgabe bereits von einer Sprache gezeigt, dass diese weder semi-entscheidbar noch co-semi-entscheidbar ist.

### Aufgabe 4: Komplexitätsanalyse [8 Punkte]

Betrachten Sie die folgende Sprache

$$L_{subword} = \{u\#w \mid w = x.u.y \text{ mit } x, y, u, w \in \{a, b\}^*\}.$$

Hierbei ist  $u\#w$  ein Wort der Sprache, wenn  $u$  ein zusammenhängendes Teilwort von  $w$  ist.

Ordnen Sie die Sprache  $L_{subword}$  möglichst genau in die folgenden Klassen ein:  $DTIME(O(f))$ ,  $NTIME(O(g))$ ,  $DSPACE(O(h))$  und  $NSPACE(O(j))$ . Findet Sie dazu möglichst kleine Funktionen  $f, g, h$  und  $j$ , sodass  $L_{subword}$  in den jeweiligen Klassen enthalten ist.

Begründen Sie ihre Wahl, indem Sie jeweils die Arbeitsweise einer passenden Turingmaschine erklären (genaue Konstruktionen sind unnötig).