Theoretical Computer Science 2
Exercise Sheet 4

Release: 28/05/2024 Due: 06/06/2024, 23:59

Hand in your answers to the Vips directory of the Stud.IP course until thursday, 06.06.2024 23:59 pm. You should provide your answers either directly as PDF file or as a readable scan or photo of your handwritten notes. Submit your results as a group of four. On the front page, state the **degree programme, name, surname and student id** of each member of your group.

Homework Exercise 1: Reductions [8 points]

Consider the following languages over $\Sigma = \{0, 1, \#\}$. Show **without** using Rice's Theorem, that none of them are semi-decidable.

a) [2 points] $L_1 := \{\, x\#y\#z \mid x \notin \mathcal{L}(M_y) \text{ or } y \notin \mathcal{L}(M_z) \text{ or } z \notin \mathcal{L}(M_x) \,\}$

b) [2 points] $L_2 := \{\, w \mid \{\varepsilon\} \subseteq \mathcal{L}(M_w) \subseteq \{0\}^* \,\}$

c) [2 points] $L_3 := \overline{L_2} = \{\, w \mid \varepsilon \notin \mathcal{L}(M_w) \text{ or } \mathcal{L}(M_w) \nsubseteq \{0\}^* \,\}$

d) [2 points] $L_4 := \{\, w \mid w \text{ encodes a contextfree Grammar } G_w \text{ with } (\Sigma\Sigma)^* \subseteq \mathcal{L}(G_w) \,\}$

Homework Exercise 2: Rice's Theorem [4 points]

Prove Theorem 5.10 of the lecture notes: Every non-monotonic property of recusively-enumerable languages (RE) is not semi-decidable. A property $P : \text{RE} \to \{0, 1\}$ is called monotone, if for all languages $L_1 \subseteq L_2$, $P(L_1) \leq P(L_2)$ holds. (If $P(L_1) = 1$, then $P(L_2) = 1$.)

a) [4 points] Show that no non-monotonic property of RE languages is semi-decidable.

Homework Exercise 3: Computability [6 points]

Consider the following partial function `longestWord` : DTM $\not\to \mathbb{N}$
and `countEquiv` : DTM $\times \{0, 1\}^* \not\to \mathbb{N}$.

$$\texttt{longestWord}(w) = \begin{cases} \max\{\, |x| \mid x \in \mathcal{L}(M_w) \,\} & \text{if } \mathcal{L}(M_w) \text{ finite} \\ \text{undefined} & \text{otherwise} \end{cases}$$

a) [3 points] Show with a reduction, that `longestWord` is uncomputable.

$$\texttt{countEquiv}(w, x) = \begin{cases} |\mathcal{L}(M_w)| & \text{if } x \in \mathcal{L}(M_w) \text{ and } \mathcal{L}(M_w) \text{ finite} \\ |\Sigma^* \setminus \mathcal{L}(M_w)| & \text{if } x \notin \mathcal{L}(M_w) \text{ and } \Sigma^* \setminus \mathcal{L}(M_w) \text{ finite} \\ 0 & \text{sonst} \end{cases}$$

b) [3 points] Show by using a reduction, that `countMissing` is uncomputable.

Exercise 4:

Show by using a reduction, that the following problems are undecidable.

Triple-PCP	
Given:	A finite sequence of triples $\langle x_1, y_1, z_1\rangle, \dots, \langle x_k, y_k, z_k\rangle$ of words over $\{0, 1\}$.
Question:	Does a non-empty sequence of indexes $i_1, \dots, i_n$ exist, such that $x_{i_1}, \dots, x_{i_n} = y_{i_1}, \dots, y_{i_n} = z_{i_1}, \dots, z_{i_n}$?

Show that the Triple-PCP is not co-semi-decidable.

Exercise 5:

Consider the following partial function choose : TM × TM × $\{0, 1\}^* \to \{0, 1\}$:

$$\text{choose}(\langle M_0\rangle, \langle M_1\rangle, x) = \begin{cases} 0 & \text{if } x \in \mathcal{L}(M_0) \\ 1 & \text{if } x \in \mathcal{L}(M_1) \setminus \mathcal{L}(M_0) \\ \text{undefined} & \text{if } x \notin \mathcal{L}(M_0) \cup \mathcal{L}(M_1) \end{cases}$$

Show that choose is uncomputable.

Exercise 6:

If possible, apply Rice's theorem on the following languages. Reason why or why not the theorem is applicable.

$L_5 = \{ w \in \{0, 1\}^* \mid \mathcal{L}(M_w) \text{ is not decidable.} \}$

$L_6 = \{ w \mid \mathcal{L}(M_w) \leq \text{HP}\}$

$L_7 = \{ w \mid \text{ exists } n \leq |\delta_{M_w}| \text{ with } 0^n \in \mathcal{L}(M_w) \}$

$L_8 = \{ w \mid \{0011\}.\mathcal{L}(M_w) = \Sigma^* \}$

Exercise 7:

Consider the following language $L_{\text{Copy}} = \{w\#w \mid w \in \{a, b\}^*\} \subseteq \{a, b, \#\}^*$.

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

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