Theory of Computer Science

G. Röger
S. Eriksson
Spring Term 2023

University of Basel
Computer Science

Exercise Sheet 8

Due: Wednesday, May 3, 2023

Exercise 8.1 (Rice's Theorem; 0.5+0.5+0.5+0.5 points)

For which of the following languages does Rice's theorem show that the language is undecidable? For each language where Rice's theorem can be used, specify the subset of Turing-computable functions $\mathcal{S}$ for which you use the theorem.
Hint: You do not have to write down any proofs. If Rice's theorem is applicable, specify the set $\mathcal{S}$, otherwise give a short reason (1 sentence) why Rice's theorem is not applicable.

(a) $L = \{w \in \{0, 1, \}^* \mid M_w$ computes a unary function over the natural numbers that is undefined on input $0\}$

(b) $L = \{w \in \{0, 1, \}^* \mid M_w$ halts on all inputs$\}$

(c) $L = \{w \in \{0, 1, \}^* \mid M_w$ always halts after an even number of steps$\}$

(d) $L = \{w \in \{0, 1, \}^* \mid M_w$ computes the binary multiplication function $mul : \mathbb{N}_0^2 \to \mathbb{N}_0$ with $mul(x, y) = x \cdot y\}$

Exercise 8.2 (Undecidability of the emptiness problem, 3 points)

The *emptiness problem* Emptiness for general (type-0) grammars is defined as:

$$\text{Given a general grammar } G\text{, is } \mathcal{L}(G) = \emptyset?$$

Prove that Emptiness is undecidable.

Hints: you can use without proof that there is a computable function that transforms a given type-0 grammar G to a DTM M_G with $\mathcal{L}(M_G) = \mathcal{L}(G)$. Likewise, there is a computable function that transforms a given DTM M to a type-0 grammar G_M with $\mathcal{L}(M) = \mathcal{L}(G_M)$.
Use Rice's theorem in an appropriate way to show the undecidability.

Exercise 8.3 (Non-deterministic algorithms; 2+2 points)

The problem HittingSet is defined as follows:

Given: finite non-empty set U,
finite set of sets $S = \{S_1, \dots, S_n\}$ with $S_i \subseteq U$ for all $S_i \in S$,
natural number $k \in \mathbb{N}_0$

Question: Does a set $H \subseteq U$ with $|H| \leq k$ exist such that $S_i \cap H \neq \emptyset$ for all $S_i \in S$?

(a) Specify a non-deterministic algorithm for HittingSet, whose runtime is polynomial in the size of the input. Briefly explain why your algorithm is correct and has polynomial runtime.

(b) Specify a deterministic algorithm for HittingSet. Does your algorithm have worst-case runtime polynomial in the size of the input? Justify your answer.

You can use any common programming concepts in your answer (IF, WHILE, FOR, recursion, ...). High-level pseudo code is sufficient as long as it can be easily seen that each step runs in polynomial time. Use the GUESS statements from the lecture for non-deterministic statements.

Exercise 8.4 (Time Complexity; 0.5+0.5 points)

You have an algorithm that solves problem X with worst-case runtime t. What can you conclude regarding the membership of X in P?

(a) $t(n) = n^3 + 2^3 n + \log_2(n)$

(b) $t(n) = n^{\log_2(n)}$