Prof. Dr. Titus Dose
Fakultät für Informatik und Mathematik
OTH Regensburg

Theoretische Informatik
Wintersemester 2024/2025
06.11.2024

Übungsblatt 5

(Besprechung am 14.11.2024)

1. **General halting problem**
   Prove that the general halting problem K is undecidable.

2. **Decidability**
   Let $f : \mathbb{N} \to \mathbb{N}$ be an arbitrary function. Prove the decidability of the following set.

   $$A = \{m \in \mathbb{N} \mid \text{there exists an } n \in \mathbb{N} \text{ with } f(n) \text{ defined and } f(n) \geq m\}$$

3. **Decidability**
   Let $M_0, M_1, \ldots$ the Gödel numbering of all RAMs discussed in the lecture. Which of the following sets is decidable and which is not. Prove your statements.

   (a) $A = \{n \in \mathbb{N} \mid \text{there are 2 distinct primes } p, q \geq 2 \text{ and } i, j \geq 1 \text{ with } n = p^i q^j\}$.

   (b) $C = \{i \in \mathbb{N} \mid M_i \text{ does not halt on even inputs } n \in \mathbb{N}\}$.

   (c) $D = \{i \in \mathbb{N} \mid M_i \text{ computes the function } c_{K_0}\}$.

   (d) $E = \{i \in \mathbb{N} \mid M_i \text{ halts on at least 2024 inputs of } \mathbb{N}\}$.

4. **Differences of square numbers**
   Prove that the set $A = \{d \in \mathbb{N} \mid \exists x, y \in \mathbb{N}, d = x^2 - y^2\}$ is in REC.

Hints

Exercise 1:
You can show the undecidability with the help of the special halting problem K_0 and Property 2.75.

Exercise 2:
The function f is not necessarily total nor necessarily computable. This task can be solved without using algorithms.

Exercise 3:
The undecidability of sets can always be shown in this task using Rice's theorem. If you apply Rice's theorem, check whether the requirements of the theorem are met.
Exactly 2 of the sets are decidable.

Exercise 4:
Consider the distance of a square number $n_1 = x^2 \in \mathbb{N}$ to the next larger square number $n_2 = (x+1)^2$. Show that every square number x^2 for $x \geq 2$ is at least $x+1$ away from every other square number.

Extra tasks

1. **RAM-Simulator**
   Write a Python program that can simulate general RAMs (cf. construction from Theorem 2.35). RAM programs are encoded as lists corresponding to the items 1 and 2 on page 217.

   Example:

Program:	Code as Python list:
`0 R3 <- 1`	$[[3, 3, 1, 0],$
`1 IF R1 = 0 GOTO 5`	$[7, 1, 5, 0],$
`2 R2 <- R2 + R0`	$[4, 2, 2, 0],$
`3 R1 <- R1 - R3`	$[5, 1, 1, 3],$
`4 GOTO 1`	$[6, 1, 0, 0],$
`5 R0 <- R2`	$[0, 0, 2, 0],$
`6 STOP`	$[9, 0, 0, 0]]$

   The Python program receives two lists as input. The first list contains the code of a RAM program, the second contains the arguments for its execution. Your Python program should return the function value computed by the RAM.

   NOTE: When having written the program, your program can —via the constructions from the proofs of Theorem 2.33 and Corollary 2.39— be converted into a RAM, which proves that there are universal RAMs.