Algorithmen und
Berechnungskomplexität II
Sommersemester 2020

Abgabe: 18.05.2020, 12.00 Uhr
Besprechung: KW 21

Prof. Dr. Heiko Röglin
Anna Großwendt, David Kübel
Institut für Informatik

Übungsblatt 4

Aufgabe 4.1: Entscheidbarkeit

Zeigen sie, dass die folgenden Sprachen entscheidbar sind. Begründen Sie jeweils Ihre Antwort.

(a) $H_{\leq 42} = \{\langle M \rangle :\ M \text{ hält auf jeder Eingabe nach höchstens 42 Schritten}\}$,

(b) $\text{TAPE}_{\text{linear}} = \{\langle M \rangle w :\ M \text{ benutzt bei Eingabe } w \text{ nur Bandzellen mit Index } i \in \{-|w|, \dots, |w|\}\}$,

(c) $L_{q_0} = \{\langle M \rangle :\ M \text{ verlässt Zustand } q_0 \text{ bei leerer Eingabe}\}$,

Die Bandzellen seien von links nach rechts aufsteigend durchnummeriert, wobei die Bandzelle, auf der der Lese-Schreib-Kopf zu Beginn steht, die Nummer 1 besitzt.

Aufgabe 4.2: WHILE-Programme

Betrachten Sie die (eingeschränkte) Programmiersprache WHILE. Programme in dieser Programmiersprache gehen aus LOOP-Programmen durch die folgenden beiden Erweiterungen hervor:

1. Zusätzliches Schlüsselwort: WHILE
2. **Bedingte Wiederholung**, d.h. falls P ein WHILE-Programm ist und x_i eine beliebige Variable, so ist auch WHILE $x_i \neq 0$ DO P END ein WHILE-Programm.

Beschreiben Sie, was die folgenden WHILE-Programme berechnen.

a) $x_1 := x_2;$
WHILE $x_3 \neq 0$ DO
  $x_1 := x_1 + 1;$
  $x_3 := x_3 - 1;$
END;

b) $x_1 := 1;$
WHILE $x_1 \neq 0$ DO
  $x_3 := x_2 + 2;$
END;

Entwerfen Sie ein WHILE-Programm, das die Funktion $\log : \mathbb{N}_0^2 \to \mathbb{N}_0$ mit $\log(x, y) = \lceil \log_x y \rceil$ berechnet. Beschreiben Sie Ihr Vorgehen.

Aufgabe 4.3: Diagonalisierung

Beweisen Sie mittels Diagonalisierung, dass es eine intuitiv berechenbare Funktion gibt, die nicht LOOP-berechenbar ist.

Kann die Argumentation auch auf WHILE-Programme angewendet werden? Das heißt, kann durch Diagonalisierung gezeigt werden, dass es eine intuitiv berechenbare Funktion gibt, die nicht WHILE-berechenbar ist? Falls ja, was impliziert dies für die These von Church und Turing? Falls nein, warum ist dies nicht möglich?

Aufgabe 4.4: Turing-Reduktion

Zeigen Sie mithilfe einer Turing-Reduktion, dass die Sprache $A_{\mathbb{P}} = \{\langle M \rangle :\ M \text{ entscheidet } \mathbb{P}\}$ nicht rekursiv ist, wobei $\mathbb{P} = \{x \in \{0,1\}^* :\ \text{val}(x) \text{ ist eine Primzahl}\}$ die Menge aller Primzahlen (in Binärkodierung) ist.