

Theoretische Informatik 2

Übungsblatt 1

René Maseli
Prof. Dr. Roland Meyer

TU Braunschweig
Sommersemester 2023

Ausgabe: 18.04.2023

Abgabe: 26.04.2023,20:00

Hand in your solutions to the Vips directory of the Stud.IP course until wednesday, 26.04.2023 06:30 pm. You should provide your solutions 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: TM Construction [10 points]

Construct an NTM which accepts the language $L := \{a^n b^m c^n d^m \mid n, m \in \mathbb{N}\} \subseteq \{a, b, c, d\}^*$. Note that n and m may also be 0. Proceed as follows.

- [3 points] Describe the idea of the workings of your TM.
- [5 points] Draw the state graph of your TM.
- [2 points] Write an accepting computation of your TM on the input $abbcdd$.

The following exercises are optional.

Exercise 2:

Konstruieren Sie eine NTM, welche eine Eingabe $w \in \{0, 1\}^+$ akzeptiert, falls die Binärzahl w (most significant bit first) gerade ist.

Exercise 3:

Konstruieren Sie eine kontext-sensitvie Grammatik für die Sprache

$$L_{<} := \{v\#w \mid v \in 0^* \cdot \text{bin}(n) \text{ UND } w \in 0^* \cdot \text{bin}(m) \text{ UND } |v| = |w| \text{ UND } n < m\} \subseteq \{0, 1, \#\}^*$$

Dabei ist $\text{bin}: \mathbb{N} \rightarrow \{0, 1\}^*$ die Binärdarstellung natürlicher Zahlen, startend mit dem most-significant bit, mit z.B. $\text{bin}(0) = \varepsilon$ und $\text{bin}(4) = 100$.

Geben Sie Ihre Grammatik formal als Tupel an. Erläutern Sie kurz, wie Ihre Grammatik die Sprache beschreibt. Geben Sie eine Ableitung Ihrer Grammatik für das Wort $001\#010 \in L_{<}$ an.

Exercise 4:

Zeigen Sie, dass alle regulären Sprachen auch per TM erkannt werden können.

Geben Sie einen NFA A für die Sprache $L = a^* \cdot (b+bb) \cdot cc$ an. Konstruieren Sie nun formal eine TM M , die wie A die obige Sprache akzeptiert. Geben Sie die Konfigurationsabfolge der Berechnung Ihrer TM für die Eingabe $aabbcc$ an.

Sei $A = (Q, \Sigma, q_0, \rightarrow, Q_F)$ nun ein beliebiger NFA. Konstruieren Sie nun formal eine TM M , die die gleiche Sprache wie A akzeptiert. Erklären Sie die Arbeitsweise der TM. Wie viel Speicherplatz benötigt ihre TM. Handelt es sich um einen LBA?

Exercise 5:

Bisher haben wir Turing-Maschinen, wie vorherige Automatenmodelle, nur als Entscheider betrachtet, d.h. als eine Maschine, die nur Ja/Nein Antworten liefert. Allgemeiner können wir Turing-Maschinen aber auch benutzen um partielle Funktionen zu berechnen. Dazu betrachten wir den Bandinhalt beim Akzeptieren als die Ausgabe der Turing-Maschine. Falls die Maschine gar nicht akzeptiert, dann gibt es keine Ausgabe (daher partielle Funktion).

Konstruieren Sie eine Turing-Maschine M^- , die als Eingabe eine Binärzahl (most significant bit first) $w \in \{0, 1\}^+$ bekommt und diese um 1 verringert. D.h. die Maschine soll die Binärzahl um 1 verringern und dann akzeptieren. Sie können davon ausgehen, dass die eingegebene Zahl immer positiv ist und um 1 verringert werden kann. Geben Sie die TM formal als Tupel an und beschreiben Sie ihre Arbeitsweise.

Konstruieren Sie nun eine Turing-Maschine M^{copy} , die die Funktion $w \mapsto w\#w$ ($w \in \{0, 1\}^*$) berechnet, d.h. als Eingabe ein Wort w bekommt und als Ausgabe $w\#w$ produzieren soll. Geben Sie die TM formal als Tupel an und beschreiben Sie ihre Arbeitsweise.

Exercise 6:

Turing-Maschinen können als Subroutinen für kompliziertere Turing-Maschinen benutzt werden. Dies soll in der folgenden Aufgabe demonstriert werden.

Wir erweitern die Transitionsfunktion von Turing-Maschinen zu $\delta : Q \times \Gamma \rightarrow (Q \times \Gamma \times \{L, R, N\}) \cup (Q \times \{M_1, M_2, \dots, M_k\})$, wobei M_1 bis M_k Turing-Maschinen sind, die als Subroutine verwendet werden dürfen. Bei einer Transition $\delta(q, a) = (q', M_i)$ passiert folgendes:

Der Inhalt des Bandes unter und rechts vom Lese/Schreibkopf wird als Eingabe in die Turing-Maschine M_i gesteckt, und dann durch die Ausgabe von M_i ersetzt. Dabei bewegt sich der Lese/Schreibkopf nicht, aber die Turing-Maschine ändert ihren Zustand zu q' . Dies alles passiert in einem Berechnungsschritt. Sie können davon ausgehen, dass M_i eine totale Funktion darstellt, also immer eine Ausgabe hat.

Konstruieren Sie nun formal eine Turing-Maschine, die folgende Funktion berechnet. Zu einer Zahl n (binär-kodiert) soll die absteigende Sequenz $bin(n)\#bin(n-1)\#bin(n-2)\#bin(n-3)\#\dots\#bin(1)\#bin(0)$ berechnet werden ($bin(n)$ ist die binäre Kodierung der Zahl n , wie in Aufgabe 1). Sie dürfen dabei die Turing-Maschinen aus Aufgabe 3 als Subroutinen benutzen, aber auch eigene Subroutinen definieren (diese müssen Sie aber formal als TM definieren!). Erklären Sie die Arbeitsweise ihrer TM und geben Sie eine formale Beschreibung als Tupel an.