Matthias Krause
Alexander Moch

Mannheim, den 17.02.2020

Vorlesung **Theoretische Informatik**, Frühjahrssemester 2020

Übungsblatt 1

Vorab: Wenn auf diesem Übungsblatt von Turingmaschinen gesprochen wird, sind damit stets **1-Band-Turingmaschinen** gemeint.

AUFGABE 1.1:

Die Turingmaschine M sei wie folgt beschrieben:
$Q = \{q_0, q_1, q_2, q_3, q_4\}$, Startzustand: q_0, $F = \{q_4\}$, $\Sigma = \{0, 1, \#\}$,

δ	0	1	#
q_0	$(q_1, \#, R)$	—	$(q_4, \#, L)$
q_1	$(q_1, 0, R)$	$(q_1, 1, R)$	$(q_2, \#, L)$
q_2	—	$(q_3, \#, L)$	—
q_3	$(q_3, 0, L)$	$(q_3, 1, L)$	$(q_0, \#, R)$
q_4	—	—	—

In der Startkonfiguration befinde sich der Kopf hier auf dem ersten (d. h. dem am weitesten links auf dem Band stehenden) Zeichen des Eingabewortes. (Falls es sich bei dem Eingabewort um das leere Wort ϵ handelt, befindet sich der Kopf in der Startkonfiguration auf einem Leerzeichen # des Bandes.)

Wie lautet die von M akzeptierte Sprache über dem Eingabealphabet $\{0, 1\}$? Stoppt M immer? Begründen Sie Ihre Behauptung detailliert.

Hinweis: F bezeichnet die Menge der akzeptierenden Zustände, sprich: Falls die Turingmaschine anhält (in der Tabelle oben durch „—" gekennzeichnet) und sich dabei in einem akzeptierenden Zustand befindet (hier: q_4), lautet die Ausgabe „akzeptieren". Hält die Turingmaschine in einem Zustand aus $Q \setminus F$ an, lautet die Ausgabe „verwerfen".

AUFGABE 1.2:

Beschreiben Sie für jedes der im Folgenden aufgelisteten Probleme, wie dieses mittels einer Turingmaschine gelöst werden kann. Formalisieren Sie jeweils ihren Algorithmus bzw. Lösungsansatz, indem sie eine vollständige Spezifikation der entsprechenden Turingmaschine unter Verwendung der gängigen Notation (vgl. Aufgabe 1) angeben.

Hinweise:

- Da es sich in den Teilaufgaben (a) und (b) nicht um Entscheidungsprobleme („Eingabe akzeptieren oder verwerfen?") handelt, muss dort keine Menge F von akzeptierenden Zuständen angegeben werden.
- In der Startkonfiguration befinde sich der Kopf hier auf dem am weitesten links auf dem Band stehenden Zeichen des Eingabewortes. (Falls es sich bei dem Eingabewort um das leere Wort ϵ handelt, befindet sich der Kopf in der Startkonfiguration auf einem Leerzeichen # des Bandes.)

a) Als Eingabe stehe eine beliebig lange Folge von Einsen auf dem Band, als Lösung stehe (nach dem Halten der Turingmaschine) diese Folge zweimal hintereinander, getrennt durch ein $\#$, auf dem Band. Für nichtleere Eingaben soll sich der Kopf in der Endkonfiguration auf dem ersten Leerzeichen rechts der Lösung (hier: rechts der am weitesten rechts stehenden Eins) befinden.

b) Als Eingabe stehen zwei strikt positive natürliche Zahlen, die addiert werden sollen, in unärer Darstellung und getrennt duch ein $\#$ auf dem Band. (Bsp.: Die zu berechnende Summe $3+4$ entspräche also der Eingabe $111\#1111$.) Gesucht ist die Summe der beiden Zahlen, die als Antwort nach dem Halten der Turingmaschine in unärer Darstellung auf dem Band stehen soll (im obigen Beispiel also 1111111). In der Endkonfiguration soll sich der Kopf auf dem ersten Leerzeichen rechts der Antwort (hier: rechts der am weitesten rechts stehenden Eins) befinden.

c) Gesucht ist abschließend eine Turingmaschine, die die Sprache $L := \{0^i 1^i \mid i \in \mathbb{N}, i > 1\}$ über dem Eingabealphabet $\{0, 1\}$ akzeptiert. Das Arbeitsalphabet der Turingmaschine sei $\Sigma = \{0, 1, \#\}$.

AUFGABE 1.3:

Ein n-Busy-Beaver ist eine Turingmaschine mit n Zuständen $\{q_0, \ldots, q_{n-1}\}$ (Startzustand: q_0) und Arbeitsalphabet $\Sigma = \{1, \#\}$. Er soll, angefangen mit dem Band, auf dem nur Leerzeichen $\#$ stehen, irgendwann stoppen und evtl. Einsen auf dem Band hinterlassen. Dabei vereinbart man, dass die Maschine (endgültig) stoppt, wenn der Kopf bei einem Schritt keine Bewegung durchführt (alternativ könnte man auch einen Endzustand einführen, der nicht mitgezählt wird).

a) Gib einen 2-Busy-Beaver an, der vier Einsen schreibt.

b) Die Funktion $bb : \mathbb{N} \to \mathbb{N}$ gibt für jedes n die maximale Anzahl von Einsen an, die ein n-Busy-Beaver schreiben kann. Zeige, dass bb streng monoton wächst.

AUFGABE 1.4:

Die folgenden drei Turingmaschinen M_1, M_2, und M_3 arbeiten über dem Arbeitsalphabet $\{0, 1, \#\}$ und bearbeiten als Eingabe einen $0,1$-String $x \in \{0, 1\}^n$, dem die Eingabelänge n zugeordnet wird. Die Turingmaschinen benutzen als Unterprogramm die in Beispiel 1 im Skript beschriebene Operation `Inkrementiere`, welche x als Binärzahl interpretiert und um 1 erhöht.
Bestimme die asymptotischen Laufzeiten der folgenden drei Turingmaschinen in O-Notation.

a) M_1 ist eine 1-Band Turingmaschine und x steht auf Band 1. M_1 wendet `Inkrementiere` solange auf die Inschrift auf Band 1 an, bis dort ein String aus n Einsen steht.

b) M_2 ist eine 2-Band Turingmaschine und x steht auf Band 1. M_2 kopiert x auf Band 2 und inkrementiert die Inschrift auf Band 1 genau so oft, wie x Binärstellen hat. Also wird die Inschrift auf Band 1 n-mal inkrementiert.

c) M_3 ist eine 3-Band Turingmaschine, x steht auf Band 1 und auf Band 2 und 3 steht jeweils 0.

M_3 geht zeichenweise über x von links nach rechts und inkrementiert den Zähler auf Band 2, wenn auf Band 1 eine 0 gelesen wird und inkrementiert den Zähler auf Band 3, wenn auf Band 1 eine 1 gelesen wird.