Übungsblatt 12

Aufgabe 1. Wahr oder falsch?

(a) Jeder linear beschränkte Automat ist eine Turingmaschine.

(b) Eine Turingmaschine darf nie das Blanksymbol $\Box$ auf das Band schreiben.

(c) Es gibt überabzählbar unendlich viele Wörter über einem endlichen Alphabet.

(d) Es gibt abzählbar unendlich viele berechenbare Funktionen $f : \mathbb{N}^k \to \mathbb{N}$.

(e) Es gibt überabzählbar unendlich viele Funktionen $f : \mathbb{N}^k \to \mathbb{N}$.

Lösung zu Aufgabe 1.

(a) wahr

Ein LBA ist eine Turingmaschine mit Einschränkungen.

(b) falsch

Nur LBAs dürfen $\Box$ nicht schreiben.

(c) falsch

Es gibt abzählbar unendlich viele Wörter über einem endlichen Alphabet. Eine mögliche Abzählung aller Wörter ergäbe sich z.B. aus der längen-lexikographischen Ordnung, bei der zuerst nach Länge und bei gleicher Länge alphabetisch sortiert wird.

(d) wahr

Siehe Folien 371 / 372 der Vorlesung.

Es gibt höchstens abzählbar viele Maschinen / Programme, die eine Funktion der Form $f : \mathbb{N} \to \mathbb{N}$ berechnen.

Dies lässt sich auf Funktionen der Form $f : \mathbb{N}^k \to \mathbb{N}$ verallgemeinern.

(e) wahr

Siehe Folie 372 der Vorlesung: Schon die Menge $\{f \mid f : \mathbb{N} \to \mathbb{N}\}$ ist überabzählbar.

Für Funktionen der Form $f : \mathbb{N}^k \to \mathbb{N}$ gilt dies also erst recht.

Mit der gleichen Technik wie in der Vorlesung können wir auch einen formalen Beweis dafür führen:

Angenommen $\mathcal{F} = \{f \mid f : \mathbb{N}^k \to \mathbb{N}\}$, die Menge alle Funktionen der Form $f : \mathbb{N}^k \to \mathbb{N}$, ist abzählbar.

Dann gibt es eine bijektive Abbildung $F : \mathbb{N} \to \mathcal{F}$.

Wir konstruieren eine Funktion $g : \mathbb{N}^k \to \mathbb{N}$ mit

$$g(x_1, \dots, x_k) = f_{x_1}(x_1, \dots, x_k) + 1$$

wobei $f_{x_1} = F(x_1)$.

Da F bijektiv (und somit surjektiv) ist, muss es ein $i \in \mathbb{N}$ geben mit $F(i) = g$.

Für dieses i gilt also $g(i, x_2, \dots, x_k) = f_i(i, x_2, \dots, x_k)$.

Dies ist aber ein Widerspruch zur Definition von g mit

$$g(i, x_2, \dots, x_k) = f_i(i, x_2, \dots, x_k) + 1.$$

Aufgabe 2. Gegeben sei die Turingmaschine $(Q, \Sigma, \Sigma \cup \{\square\}, \delta, z_0, \square, \{z_2\})$ mit $Q = \{z_0, z_1, z_2\}$, $\Sigma = \{0, 1\}$ und folgenden Transitionen:

$$\begin{aligned}
\delta(z_0, 0) &= (z_0, 0, R)\\
\delta(z_0, 1) &= (z_0, 1, R)\\
\delta(z_0, \square) &= (z_1, 0, L)\\
\delta(z_1, 0) &= (z_1, 0, L)\\
\delta(z_1, 1) &= (z_1, 1, L)\\
\delta(z_1, \square) &= (z_2, \square, R)
\end{aligned}$$

(a) Untersuchen Sie, wie sich die Turingmaschine auf den Eingaben 10, 11 und 110 verhält. Was tut sie allgemein bei Eingaben $w \in 1\{0,1\}^* \cup \{0\}$?

(b) Wie verändert sich das Verhalten, wenn man die Transition $\delta(z_0, \square) = (z_1, 0, L)$ durch $\delta(z_0, \square) = (z_1, 1, L)$ ersetzt?

Lösung zu Aufgabe 2. (a) Bei beliebigen Eingaben $w \in \Sigma^*$ läuft die Turingmaschine bis zum Ende der Eingabe, ersetzt das erste Blanksymbol durch eine 0 und läuft dann wieder zum linken Ende der Eingabe zurück.

Aus der Eingabe 10 wird so 100, aus 11 wird 110 und aus 110 wird 1100.

Eingaben der Form $w \in 1\{0,1\}^* \cup \{0\}$ können als binär kodierte Zahlen verstanden werden.

Kodiert die Eingabe die Zahl n, so steht nach dem Durchlauf der Turingmaschine die Binärkodierung von $2n$ auf dem Band, da das Anfügen einer 0 an eine Zahl im Binärsystem der Multiplikation der Zahl mit 2 entspricht.

(b) Mit der Änderung wird am Bandende eine 1 statt einer 0 angefügt.

Interpretieren wir die Eingabe wieder als eine binär kodierte Zahl n, so steht nach dem Durchlauf die Binärkodierung von $2n + 1$ auf dem Band.

Aufgabe 3. Geben Sie eine Turingmaschine an, die bei Eingabe $w \in \{a, b\}^*$ das Wort ww auf das Band schreibt, den Lesekopf auf das erste Zeichen von ww bewegt und in einen Endzustand übergeht. Skizzieren Sie anschließend, wie eine äquivalente 2-Band-Turingmaschine aussehen würde.

Lösung zu Aufgabe 3.

- $M = (Z, \Sigma, \Gamma, \delta, z_0, \Box, \{z_e\})$
- $Z = \{z_0, z_1, z_2, z_3, z_4, z_5, z_e\}$
- $\Sigma = \{a, b\}$
- $\Gamma = \Sigma \cup \{\Box, \#_a, \#_b, \$_a, \$_b\}$

Dabei ist δ wie folgt definiert:

z_0: Ersetzen von a und b

- $\delta(z_0, \Box) = (z_e, \Box, N)$
- $\delta(z_0, a) = (z_1, \#_a, R)$
- $\delta(z_0, b) = (z_2, \#_b, R)$
- $\delta(z_0, x) = (z_4, x, R)$ für alle $x \in \{\$_a, \$_b\}$

Das leere Wort wird direkt akzeptiert. Die Buchstaben a oder b ersetzen wir jeweils durch $\#_a$ bzw. $\#_b$ (wobei bereits ersetzte Buchstaben übergangen werden) und gehen in z_1 bzw. z_2 über, um ein $\$_a$ oder $\$_b$ rechts an das Band anzuhängen. Der letzte Übergang wird erst später relevant. Steht der Lesekopf im Zustand z_0 auf einem $\$_a$ oder $\$_b$ haben wir alle a's und b's verarbeitet und können anfangen die $\#_a$, $\#_b$, $\$_a$, $\$_b$ in a's und b's umzuschreiben.

<u>z_1: $\$_a$ rechts ans Band anhängen</u>

- $\delta(z_1, x) = (z_1, x, R)$ für alle $x \in \{a, b, \$_a, \$_b\}$
- $\delta(z_1, \square) = (z_3, \$_a, L)$

<u>z_2: $\$_b$ rechts ans Band anhängen</u>

- $\delta(z_2, x) = (z_2, x, R)$ für alle $x \in \{a, b, \$_a, \$_b\}$
- $\delta(z_2, \square) = (z_3, \$_b, L)$

<u>z_3: Nach links zum letzten $\#_a$ oder $\#_b$</u>

- $\delta(z_3, x) = (z_3, x, L)$ für alle $x \in \{a, b, \$_a, \$_b\}$
- $\delta(z_3, x) = (z_0, x, R)$ für alle $x \in \{\#_a, \#_b\}$

<u>z_4: Laufe zum rechten Bandende</u>

- $\delta(z_4, x) = (z_4, x, R)$ für alle $x \in \{\$_a, \$_b\}$
- $\delta(z_4, \square) = (z_5, \square, L)$

<u>z_5: Symbole ersetzen</u>

- $\delta(z_5, x) = (z_5, a, L)$ für alle $x \in \{\#_a, \$_a\}$
- $\delta(z_5, x) = (z_5, b, L)$ für alle $x \in \{\#_b, \$_b\}$
- $\delta(z_5, \square) = (z_e, \square, R)$

Nachdem das Wort mit Hilfe der Hilfssymbole kopiert wurde, wird mit z_4 zum rechten Bandende gelaufen. Danach durchlaufen wir mit z_5 das Band von rechts nach links und ersetzen alle $\#_a$'s und $\$_a$'s durch a und alle $\#_b$'s und $\$_b$'s durch b. Am Ende (wenn wir das linke $\square$ erreichen), gehen wir nach rechts und akzeptieren mit dem Lesekopf auf dem ersten Symbol.

Im Gegensatz zu dieser 1-Band Konstruktion, ist eine 2-Band TM deutlich weniger aufwendig. Die Idee ist, dass wir w von Band 1 auf Band 2 kopieren, mit dem Lesekopf auf Band 1 stehen bleiben, während wir auf Band 2 wieder an den Anfang fahren und zum Schluss die Kopie von w von Band 2 auf Band 1 hinter die Eingabe anhängen. Wir können ab diesem Zeitpunkt das zweite Band ignorieren und auf Band 1 an den Start zurücklaufen.

Formal sähe die eben beschriebene 2-Band TM wie folgt aus:

- $M' = (Z', \Sigma, \Gamma', \delta', z_0, \square, \{z_e\})$
- $Z' = \{z_0, z_1, z_2, z_3, z_e\}$
- $\Sigma = \{a, b\}$
- $\Gamma' = \Sigma \cup \{\square\}$

Hierbei ist δ' gegeben durch

- $\delta(z_0, (x, \square)) = (z_0, (x, x), (R, R))$ für alle $x \in \{a, b\}$
- $\delta(z_0, (\square, \square) = (z_1, (\square, \square), (N, L))$
- $\delta(z_1, (\square, x)) = (z_1, (\square, x), (N, L))$ für alle $x \in \{a, b\}$
- $\delta(z_1, (\square, \square) = (z_2, (\square, \square), (N, R))$
- $\delta(z_2, (\square, x)) = (z_2, (x, x), (R, R))$ für alle $x \in \{a, b\}$
- $\delta(z_2, (\square, \square) = (z_3, (\square, \square), (L, N))$
- $\delta(z_3, (x, \square)) = (z_3, (x, \square), (L, N))$ für alle $x \in \{a, b\}$
- $\delta(z_3, (\square, \square) = (z_e, (\square, \square), (R, N))$

Wie bereits auf Blatt 11 sei hier angemerkt, dass für nicht erreichbare Paare $(z, \gamma) \in Z \times \Gamma$ eigentlich auch Übergänge definiert werden müssten, wenn wir eine deterministische TM erhalten wollen. Gleiches für die 2-Band TM.

Aufgabe 4. Geben Sie (formal) Turingmaschinen M_1 bzw. M_2 an, die die Funktionen $f_i : \mathbb{N}^2 \to \mathbb{N}$ mit

$$f_i(n_1, n_2) = n_i \quad (i = 1, 2)$$

berechnen.

Lösung zu Aufgabe 4.
Zu Beginn steht $\mathrm{bin}(n_1)\#\mathrm{bin}(n_2)\#$ auf dem Band. Nach einem Durchlauf der TM für $f_i, i \in \{1, 2\}$ soll nur noch $\mathrm{bin}(n_i)$ auf dem Band stehen.

<u>Beachte</u>: Der Lesekopf der TM muss nach dem Durchlauf auf dem ersten Symbol von $\mathrm{bin}(n_1)$ bzw. $\mathrm{bin}(n_2)$ stehen.

TM für f_1

- $M_1 = (Z_1, \Sigma_1, \Gamma_1, \Box, \delta_1, z_0, \{z_e\})$
- $Z_1 = \{z_0, z_1, z_2, z_3, z_e\}$
- $\Sigma_1 = \{0, 1, \#\}$
- $\Gamma_1 = \Sigma \cup \{\Box\}$

Dabei ist δ_1 wie folgt definiert:

z_0: Überspringen von $\text{bin}(n_1)$

- $\delta_1(z_0, x) = (z_0, x, R)$ für alle $x \in \{0, 1\}$
- $\delta_1(z_0, \#) = (z_1, \#, R)$

Wir durchlaufen das Band von links nach rechts, bis wir auf ein # treffen. Dann gehen wir in z_1 über, um den restlichen Bandinhalt zu löschen.

z_1: Löschen des restlichen Bandinhaltes

- $\delta_1(z_1, x) = (z_1, \Box, R)$ für alle $x \in \{0, 1, \#\}$
- $\delta_1(z_1, \Box) = (z_2, \Box, L)$

Alle Symbole von $\text{bin}(n_2)\#$ werden mit $\Box$ überschrieben. Wenn das rechte Bandende erreicht ist, gehen wir in z_2 über, um zurück zum linken Bandende zu laufen.

z_2, z_3: Zurück zum linken Bandende

- $\delta_1(z_2, \Box) = (z_2, \Box, L)$
- $\delta_1(z_2, \#) = (z_3, \Box, L)$
- $\delta_1(z_3, x) = (z_3, x, L)$ für alle $x \in \{0, 1\}$
- $\delta_1(z_3, \Box) = (z_e, \Box, R)$

Hier brauchen wir zwei Zustände, da die $\Box$'s rechts und links von $\text{bin}(n_1)$ unterschiedlich behandelt werden.

Das mittlere # haben wir zu Beginn mit z_0 stehen gelassen um diesen Übergang zu markieren.

TM für f_2

- $M_2 = (Z_2, \Sigma_2, \Gamma_2, \Box, \delta_2, z_0, \{z_e\})$
- $Z_2 = \{z_0, z_1, z_2, z_e\}$
- $\Sigma_2 = \{0, 1, \#\}$
- $\Gamma_2 = \Sigma_2 \cup \{\Box\}$

Dabei ist δ_2 wie folgt definiert:

z_0: Löschen von bin(n_1)

- $\delta_2(z_0, x) = (z_0, \Box, R)$ für alle $x \in \{0, 1\}$
- $\delta_2(z_0, \#) = (z_1, \Box, R)$

z_1: Löschen des letzten #

- $\delta_2(z_1, x) = (z_1, x, R)$ für alle $x \in \{0, 1\}$
- $\delta_2(z_1, \#) = (z_2, \Box, L)$

z_2: Zurück zum Anfang von bin(n_2)

- $\delta_2(z_2, x) = (z_2, x, L)$ für alle $x \in \{0, 1\}$
- $\delta_2(z_2, \Box) = (z_e, \Box, R)$

Aufgabe 5. Wahr oder falsch?

(a) Das folgende LOOP-Programm terminiert nicht.

$$x_1 := 5;\ \textsc{Loop}\ x_1\ \textsc{Do}\ x_1 := x_1 + 1;\ \textsc{End}$$

(b) Das folgende WHILE-Programm berechnet die Funktion $f(x) = 0$.

$$\textsc{While}\ x \neq 0\ \textsc{Do}\ x := x - 2; x := x + 1;\ \textsc{End}$$

Lösung zu Aufgabe 5.

(a) falsch

Der LOOP wird $x_1 = 5$ mal ausgeführt, das veränderte x_1 wird nicht erneut gelesen. Grundsätzlich gilt: LOOP-Programme terminieren immer.

(b) falsch

Das Programm berechnet $f(x) = \begin{cases} 0 & \text{falls } x = 0 \\ \text{undef} & \text{falls } x \geq 1 \end{cases}$

Fall 1, $x = 0$

Das Programm terminiert mit einem Rückgabewert von 0.

Fall 2, $x \geq 1$

Das Programm terminiert nicht. Falls $x \geq 2$, dekrementiert ein Durchlauf der Schleife den Wert um 1, bis $x = 1$ erreicht ist.

Für $x = 1$ ist der Wert nach einem Durchlauf der Schleife aber wieder 1, da wir beim Subtrahieren nicht ins Negative, sondern nur bis zur 0 gehen (also $a - b = 0$ für $b \geq a$). Somit ergibt $x := x - 2$ zuerst $x = 0$ und anschließend erhalten wir durch $x := x + 1$ wiederum $x = 1$, was zu einer Endlosschleife führt.