Aufgabe 1

//Mehrband-Turing-Maschine, Verdopplungstransformation?

Zweiband-Turing-Maschine $(Q, \Sigma, \Gamma, \delta, q_0, \square, \{q_4\})$
mit Ausgabeband = Eingabeband als Band 1 und Arbeitsspeicherband als Band 2

$Q = \{q_0, q_1, q_2, q_3, q_4\} \quad \wedge \quad \square \notin \Sigma$ gegeben $\quad \wedge \quad \Gamma := \Sigma \cup \{\square\}$

$\delta: (Q \setminus \{q_4\}) \times \Gamma^2 \rightarrow Q \times (\Gamma \times \{L, N, R\})^2$

$\forall x \in \Sigma \quad \forall y \in \Gamma: \; \delta(q_0, (x, y)) = (q_0, (x, R), (x, R))$ //Band1 → Band2 kopieren

für $x = \square \quad \forall y \in \Gamma: \; \delta(q_0, (\square, y)) = (q_1, (\square, L), (\square, L))$ //fertig kopiert

$\forall x \in \Sigma \quad \forall y \in \Gamma: \; \delta(q_1, (x, y)) = (q_1, (x, L), (x, L))$ //Bänder zurückspulen

für $x = \square \quad \forall y \in \Gamma: \; \delta(q_1, (\square, y)) = (q_2, (\square, R), (\square, R))$ //fertig zurückgespult

$\forall x \in \Gamma \quad \forall y \in \Sigma \; \delta(q_2, (x, y)) = (q_3, (y, R), (y, N))$ //schreibe ersten Paar-Teil

$\forall x \in \Gamma$ für $y = \square \; : \; \delta(q_2, (x, \square)) = (q_4, (\square, L), (\square, L))$ //beende Programm

$\forall x \in \Gamma \quad \forall y \in \Gamma: \; \delta(q_3, (x, y)) = (q_2, (y, R), (y, R))$ //schreibe zweiten Paar-Teil

Aufgabe 2

//Radó - Turingmaschine?

$|Q_n \setminus F| =: n \in \{1, 2, 3\} \quad \wedge \quad \Sigma := \{1\} \quad \wedge \quad \Gamma := \{1, \square\} \quad \wedge \quad \delta: Q_n \times \Gamma \rightarrow Q_n \times \Gamma \times \{L, N, R\}$

Turingmaschine $M_n := (Q_n, \Sigma, \Gamma, \delta_n, q_0, \square, F)$

2.1: $Q_1 = \{q_0, q_F\}$

$\forall x \in \Gamma: \; \delta_1(q_0, x) = (q_F, 1, N)$ //schreiben und fertig

2.2: $Q_2 = \{q_0, q_1, q_F\}$ //Die Indizes der 1 dient nur der Lesbarkeit der Ableitungssequenz

Strategie: Immer 1 setzen, ansonsten alles abwechseln und schauen was rauskommt...

$\delta_2(q_0, \square) = (q_1, 1_0, R) \quad \wedge \quad \delta_2(q_0, 1) = (q_1, 1_1, L) \quad \wedge$

$\delta_2(q_1, \square) = (q_0, 1_2, L) \quad \wedge \quad \delta_2(q_1, 1) = (q_F, 1_3, R)$

Ableitungssequenz: // $\boxtimes = \square$, „Zeiger-Highlighting“

$\square\square\square q_0 \boxtimes \square\square \vdash \square\square\square 1_0 q_1 \boxtimes \square \vdash \square\square\square q_0 \underline{1_0} 1_2 \square \vdash \square\square q_1 \boxtimes 1_1 1_2 \square \vdash$
$\square q_0 \boxtimes 1_2 1_1 1_2 \square \vdash \square 1_0 q_1 \underline{1_2} 1_1 1_2 \square \vdash \square 1_0 1_3 q_F \underline{1_1} 1_2 \square \vdash$

2.3: $Q_3 = \{q_0, q_1, q_2, q_F\}$

Strategie: versuchen, bei obigem Ergebnis rechts und links noch eine Eins dranzubekommen.

$\delta_2(q_0, \square) = (q_1, 1_0, R) \quad \wedge \quad \delta_2(q_0, 1) = (q_2, 1_1, L) \quad \wedge$

$\delta_2(q_1, \square) = (q_0, 1_2, L) \quad \wedge \quad \delta_2(q_1, 1) = (q_1, 1_3, R)$

$\delta_2(q_2, \square) = (q_1, 1_4, L) \quad \wedge \quad \delta_2(q_2, 1) = (q_F, 1_5, N)$

Ableitungssequenz:

$\square\square\square\square q_0 \boxtimes \square\square\square \vdash \square\square\square\square 1_0 q_1 \boxtimes \square\square \vdash \square\square\square\square q_0 \underline{1_0} 1_2 \square\square \vdash \square\square\square q_2 \boxtimes 1_1 1_2 \square\square \vdash$
$\vdash \square\square q_1 \boxtimes 1_4 1_1 1_2 \square\square \vdash \square q_0 \boxtimes 1_2 1_4 1_1 1_2 \square\square \vdash \square 1_0 q_1 1_2 1_4 1_1 1_2 \square\square \vdash$ //ähnlich wie oben
$\vdash \ldots \vdash \square 1_0 1_3 1_3 1_3 1_3 q_1 \boxtimes \square \vdash \square 1_0 1_3 1_3 1_3 q_0 1_3 1_2 \square \vdash$ // q_1 geht komplett nach rechts durch
$\vdash \square 1_0 1_3 1_3 q_2 1_3 1_1 1_2 \square \vdash \square 1_0 1_3 1_3 q_F 1_5 1_1 1_2 \square \vdash$

Aufgabe 3

//verschiedene Turingmaschinen bauen?

3.1: //Turingmaschine für reguläre Sprache „gerade Anzahl 1en“

Interpretation: Das leere Wort soll nicht enthalten sein, weil ja mindestens das Paritätsbit enthalten sein soll. Das Paritätsbit des leeren Wortes ist 0, daher bekommt das Wort 0 grünes Licht.

Definiere $a\colon \{0, 1, 2\} \rightarrow \{1, 2\}\colon \quad a(i) = 2 - ((i+1)\,MOD\,2)$, also $a(0) = a(2) = 1 \;\wedge\; a(1) = 2$

//a steht für der **a**ndere Index, a² ist dann der gleiche Index, aber in {1, 2}

Um das leere Wort auszuschließen, benötigen wir einen separaten Startzustand.

$\forall i \in \{0, 1, 2\}\colon \quad \delta(q_i, 0) = (q_{a^2(i)}, 0, R) \quad \wedge \quad \delta(q_i, 1) = (q_{a(i)}, 1, R)$

$\delta(q_0, \square) = \delta(q_1, \square) = \perp \quad \wedge \quad \delta(q_2, \square) = (q_{JA}, \square, N)$

3.2: //Inkrement mit Turingmaschine

Definiere $+_2$ als die binäre Addition $1 +_2 1 = 0$

$\forall x \in \Sigma\colon \quad \delta(q_0, x) = (q_0, x, R) \quad \wedge \quad \delta(q_0, \square) = (q_1, \square, L)$ //q₀: nach rechts fahren

$\delta(q_1, 0) = (q_2, 1, L) \quad \wedge \quad \delta(q_1, 1) = (q_1, 0, L) \quad \wedge \quad \delta(q_1, \square) = (q_2, 1, L)$ //q1: Übertrag

$\delta(q_2, 0) = (q_2, 0, L) \quad \wedge \quad \delta(q_2, 1) = (q_2, 1, L) \quad \wedge \quad \delta(q_2, \square) = (q_{JA}, \square, L)$ //q2: kein Übertrag

3.3: //Binärzahlkopierer

$\forall x \in \Sigma\colon \quad \delta(q_0, x) = (q_1, x, R) \quad \wedge \quad \delta(q_0, \square) = \perp$ //erstes Zeichen ignorieren

$\forall x \in \Sigma\colon \quad \delta(q_1, x) = (q_{x_{MOD2}+2}, \square, R) \quad \wedge \quad \delta(q_1, \square) = (q_{10}, \square, L)$ //Schleifenbeginn

//q_2 kopiert 0 nach rechts, q_3 kopiert 1 n.r.

//Wenn wir fertig sind, ist das nächste Zeichen das ignorierte erste Zeichen.

$\forall i \in \{2, 3\} \;\; \forall x \in \Sigma\colon \quad \delta(q_i, x) = (q_i, x, R) \quad \wedge \quad \delta(q_i, \square) = (q_{i+2}, \square, R)$ //erstes blank ignorieren

$\forall i \in \{4, 5\} \;\; \forall x \in \Sigma\colon \quad \delta(q_i, x) = (q_i, x, R) \quad \wedge \quad \delta(q_i, \square) = (q_{i+2}, i_{MOD2}, L)$ //schreiben und zurück

$\forall i \in \{6, 7\} \;\; \forall x \in \Sigma\colon \quad \delta(q_i, x) = (q_i, x, L) \quad \wedge \quad \delta(q_i, \square) = (q_{i+2}, i_{MOD2}, L)$ //erstes blank ignore

$\forall i \in \{8, 9\} \;\; \forall x \in \Sigma\colon \quad \delta(q_i, x) = (q_i, x, L) \quad \wedge \quad \delta(q_i, \square) = (q_1, i_{MOD2}, R)$

$\forall x \in \Sigma\colon \quad \delta(q_{10}, x) = (q_{10}, x, L) \quad \wedge \quad \delta(q_{10}, \square) = (q_{11}, \square, R)$ //jetzt wieder beim ersten Zeichen

$\forall x \in \Sigma\colon \quad \delta(q_{11}, x) = (q_{12+x_{MOD2}}, x, R) \quad \wedge \quad \delta(q_{11}, \square) = \perp$ //blank hier nicht möglich!

$\forall i \in \{12, 13\} \;\; \forall x \in \Sigma\colon \quad \delta(q_i, x) = (q_i, x, L) \quad \wedge \quad \delta(q_i, \square) = (q_{JA}, i_{MOD2}, R)$

Theoretische Informatik, Übungsblatt 13 Florian Ott

Aufgabe 4

//rekursive mirror-Funktion?

4.1 //Beispielwort

Für das Wort $\vec{w} := aabbb$ ist $mir(\vec{w})=bbbaa$ das Spiegelbild des Wortes.

4.2 //mir turing-berechenbar

Definiere bijektive Funktion $f:\ \Sigma=\{a, b\} \rightarrow \{0, 1\}$ durch $f(a)=0 \wedge f(b)=1$

//Beispiel: 12345 (Die Ziffern symbolisieren Indizes a_1, a_2, a_3, a_4, a_5 mit $a_i \in \Sigma$)

$\forall x \in \Sigma:\quad \delta(q_0, x)=(q_{1+f(x)}, \square, R) \quad \wedge \quad \delta(q_0, \square)=(q_6, \square, R)$ // erstes Zeichen mitnehmen

//Beispiel: □2345, q ist mit 1 nach rechts unterwegs

$\forall i \in \{1, 2\}\ \forall x \in \Sigma:\quad \delta(q_i, x)=(q_i, x, R) \quad \wedge \quad \delta(q_i, \square)=(q_3, f^{-1}(i-1), L)$ // nach rechts

// das mitgenommene erste Zeichen wird hinter dem String abgelegt (erste freie Feld)

//Beispiel: □234□1, q ist mit 5 nach links unterwegs

für i = 3 $\forall x \in \Sigma:\quad \delta(q_3, x)=(q_{4+f(x)}, \square, L) \quad \wedge \quad \delta(q_i, \square)=(q_6, \square, R)$

$\forall i \in \{4, 5\}\ \forall x \in \Sigma:\quad \delta(q_i, x)=(q_i, x, L) \quad \wedge \quad \delta(q_i, \square)=(q_0, f^{-1}(i-4), R)$

// das mitgenommene letzte Zeichen landet auf dem ersten Platz. Dann Schleife.

//Beispiel: 5 q_0 234□1

//Beispiel nach zweitem Durchlauf: 54 q0 3□21 und im drittem Durchlauf: 54 q0 □321,

//jetzt soll das □ nach links mitgenommen werden, also Abbruch: q_6

//Schleifenabbruchkriterium (q_6): Ein blank-Symbol soll mitgenommen werden.

//Wir sind fast fertig, aber müssen das □ noch entfernen:

für i = 6 $\forall x \in \Sigma:\quad \delta(q_6, x)=(q_{7+f(x)}, \square, L) \quad \wedge \quad \delta(q_i, \square)=(q_{JA}, \square, N)$

$\forall i \in \{7, 8\}\ \forall x \in \Gamma:\quad \delta(q_i, x)=(q_6, f^{-1}(i-7), R)$

//Beispiel: 54□321 $\vdash$ 543□21 $\vdash$ 5432□1 $\vdash$ 54321□

4.3 //Band- und Zeitkomplexität

An Band benötige ich genau 1 Zeichen mehr als das Wort selbst hat, sowohl für Lese- als auch für Schreibzugriff.

Zur Zeitkomplexität: Zu Beginn gehen die Zeichen an allen anderen Zeichen vorbei bis auf die bereits abgearbeiteten, also Laufzeit t(n) = n + (n-1) + (n-2) +…+ 1 = n(n+1)/2 ist $O(n^2)$.

Danach wird das □ aus der Mitte entfernt, also Laufzeit t(n) = n ist O(n).

Gesamtlaufzeitkomplexität ist $O(n^2)$.