Theorie der Informatik

G. Röger
Frühjahrssemester 2019

Universität Basel
Fachbereich Informatik

Präsenzaufgaben 6 — Lösungen

Aufgabe 6.1

Gegeben sei eine Turing Maschine mit $E = \{q_E\}$, welche die Funktion $f(\mathtt{a}^n) = (\mathtt{ab})^n$ für $n \in \mathbb{N}_0$ berechnet. Muss die Turing Maschine unter dem folgenden Input anhalten? Falls ja, wie sieht die Endkonfiguration aus?

(i) `a`

Lösung:

Die TM hält mit Endkonfiguration $\langle \square \dots \square, q_E, \mathtt{ab}\square \dots \square \rangle$

(ii) `aa`

Lösung:

Die TM hält mit Endkonfiguration $\langle \square \dots \square, q_E, \mathtt{abab}\square \dots \square \rangle$

(iii) ε

Lösung:

Die TM hält mit Endkonfiguration $\langle \square \dots \square, q_E, \square\square \dots \square \rangle$

(iv) `ab`

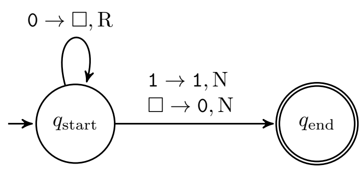
Lösung:

Da der Input ungültig ist für die zu berechnende Funktion ist, gibt es 2 Möglichkeiten: (1) Die TM bleibt in einer Endlosschleife oder (2) Die TM endet in einer ungültigen Konfiguration (Kopf nicht am Anfang der Ausgabe, $\square$ zwischenendrin oder Symbole aus $\Gamma \setminus \Sigma$ auf dem Band).

Aufgabe 6.2

(a) Geben Sie eine Turing Maschine an, welche einen Präfix aus Nullen einer Eingabe über $\Sigma = \{0, 1\}$ entfernt. Besteht die Eingabe nur aus Nullen oder ist ε, so soll das Ergebnis 0 sein.

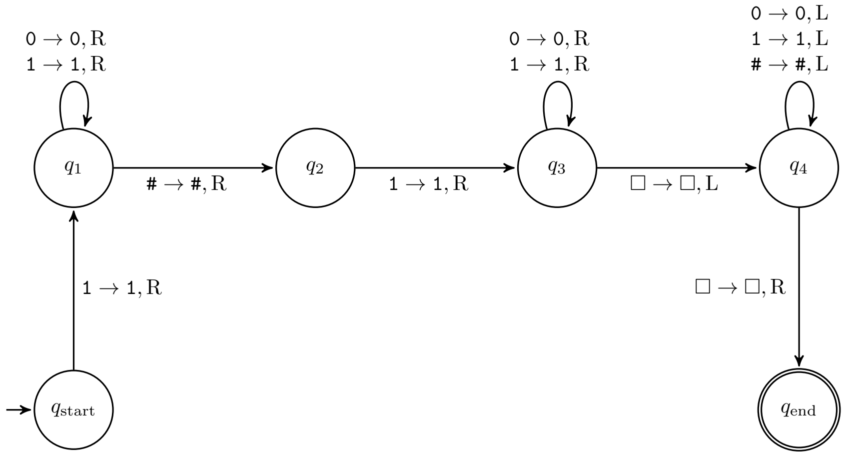
Lösung:



(b) Geben Sie eine Turing Maschine an, welche eine Eingabe über $\Sigma = \{0, 1, \mathtt{\#}\}$ genau dann akzeptiert, wenn diese zwei positive, durch ein `#`-Zeichen getrennte Binärzahlen kodiert.

Lösung:

Alle fehlenden Transitionen gehen in einen Fehlerzustand.



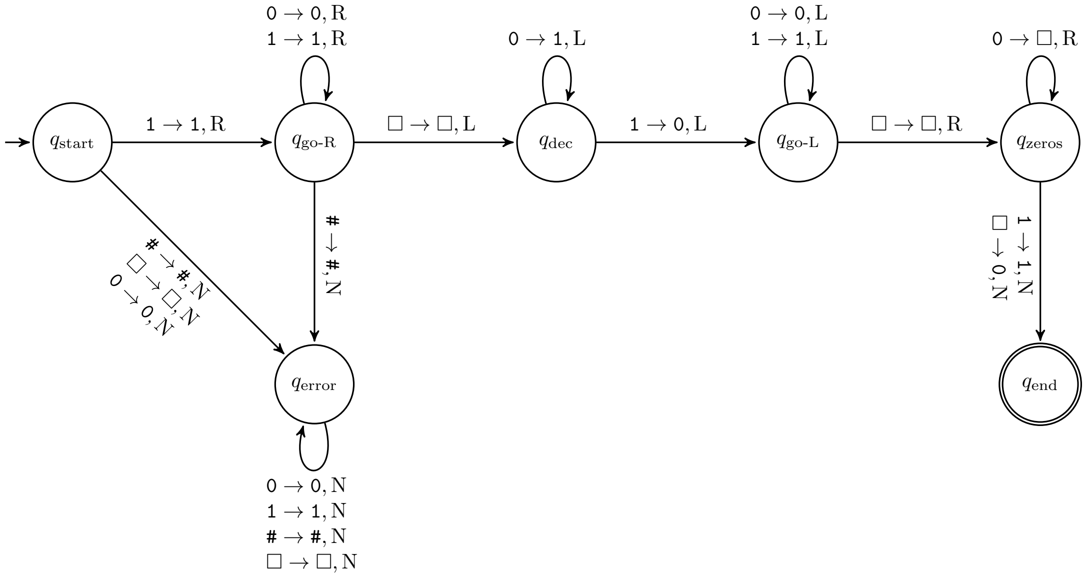
Aufgabe 6.3

Geben Sie das Zustandsdiagramm einer Turingmaschine an, die die *Vorgängerfunktion* $pred_2 : \mathbb{N}_0 \to \mathbb{N}_0$ über den natürlichen Zahlen berechnet:

$$pred_2(n) = \begin{cases} n-1 & \text{falls } n \geq 1 \\ \text{undefiniert} & \text{falls } n = 0 \end{cases}$$

Lösung:

Nach Definition der Berechnung einer numerischen Funktion durch eine DTM müssen wir eine DTM M angeben, die die *Kodierung* $pred_2^{\text{code}} : \{\mathtt{0}, \mathtt{1}, \mathtt{\#}\}^* \to \{\mathtt{0}, \mathtt{1}, \mathtt{\#}\}^*$ von $pred_2$ berechnet, d.h. die Funktion, die auf die Eingabe $bin(n)$ die Ausgabe $bin(pred_2(n))$ berechnet.
Anmerkung: nach Definition der Berechenbarkeit von Funktionen in $\Sigma^* \to \Sigma^*$ bedeutet dies streng genommen auch, dass auf *ungültigen* Eingaben, also z.B. Eingabewörtern wie `#011##`, `11#10`, ε oder `011`, M nicht in einer gültigen Konfiguration anhalten darf. Wir berücksichtigen dies dadurch, dass M bei ungültigen Eingaben in einen Fehlerzustand q_{error} geht und in diesem für immer verbleibt.
Wir verwenden (wie in der Definition der Turing-Berechenbarkeit für numerische Funktionen gefordert) das Eingabealphabet $\Sigma = \{\mathtt{0}, \mathtt{1}, \mathtt{\#}\}$. Als Bandalphabet verwenden wir $\Gamma = \Sigma \cup \{\square\}$, also keine zusätzlichen Symbole ausser dem Blank-Zeichen. Der Anfangszustand ist q_{start} und der einzige Endzustand ist q_{end}. Wir definieren also die DTM $M = \langle Q, \Sigma, \Gamma, \delta, q_{\text{start}}, \square, \{q_{\text{end}}\}\rangle$ mit $Q = \{q_{\text{start}}, q_{\text{error}}, q_{\text{go-R}}, q_{\text{dec}}, q_{\text{go-L}}, q_{\text{zeros}}, q_{\text{end}}\}$ und der Übergangsfunktion δ, die im folgenden Zustandsdiagramm definiert wird. Wir lassen dabei Transitionen weg, die nie auftreten können (z.B. Lesen von `#` im Zustand $q_{\text{go-L}}$). Diese können beliebig gewählt werden.



Die DTM funktioniert wie folgt:

- Im Anfangszustand q_{start} wird das erste Eingabezeichen getestet. Es gibt zwei Möglichkeiten:
  - Wenn das Eingabewort mit `0`, `#`, oder $\Box$ beginnt, ist es ungültig. In diesen Fällen gehen wir in den Fehlerzustand.
  - Wenn das Eingabewort mit `1` beginnt, ist weitere Verarbeitung nötig, die in Zustand $q_{\text{go-R}}$ beginnt.
- Zustand $q_{\text{go-R}}$ hat die Aufgabe, bei Eingaben der Form $\mathtt{1}\ldots$ zum rechten Eingabeende zu gehen und dabei zu testen, ob die Eingabe gültig ist. Falls nein (genau dann, wenn ein `#` auftritt), gehen wir in den Fehlerzustand. Ansonsten gehen wir schliesslich in den Zustand q_{dec} über.
- Beim Betreten von q_{dec} schaut die DTM auf die letzte Ziffer des Eingabewortes. Die Aufgabe des Zustandes ist es, die Zahl zu dekrementieren (um 1 zu reduzieren). Hierzu ersetzen wir `0` durch `1` und gehen nach links, bis wir schliesslich eine `1` erreichen. Diese wird durch `0` ersetzt, womit die Dekrementierung vollendet ist.
- Danach gehen wir in Zustand $q_{\text{go-L}}$ ganz nach links zum Bandanfang.
- Abschliessend müssen wir noch allfällige führende Nullen entfernen und anhalten. Dies übernehmen die Zustände q_{zeros} und q_{end}.