Prof. Javier Esparza
Philipp Czerner, Martin Helfrich
Technische Universität München
Lehrstuhl für Theoretische Informatik

Einführung in die Theoretische Informatik

Sommersemester 2021 – Übungsblatt 9

- Das Übungsblatt ist in zwei Teile gegliedert: den Vorbereitungsteil, den Sie vor der Übung selbstständig bearbeiten sollen, und den Übungs-/Nachbereitungsteil, der Aufgaben enthält, die in der Übung besprochen werden und von Ihnen anschließend zur Nachbereitung verwendet werden können.
- Das ist nicht das Hausaufgabenblatt! Die Hausaufgaben finden Sie auf einem separaten Blatt.

Vorbereitung (→ vor der Übung selbständig zu bearbeiten)

Individualaufgabe Ü9.1. (*Wichtige Begriffe*)

Überprüfen Sie, dass Sie die folgenden Begriffe oder Notationen korrekt definieren können.

- abzählbar / überabzählbar
- berechenbar / unberechenbar
- Church-Turing These
- nichtdeterministische / deterministische Turing-Maschine (TM)
- Konfiguration einer TM
- akzeptierte Sprache einer TM
- Turing-berechenbar
- k-Band Turing-Maschine
- WHILE-Programm
- GOTO-Programm
- Konvertierung: WHILE → TM → GOTO → WHILE

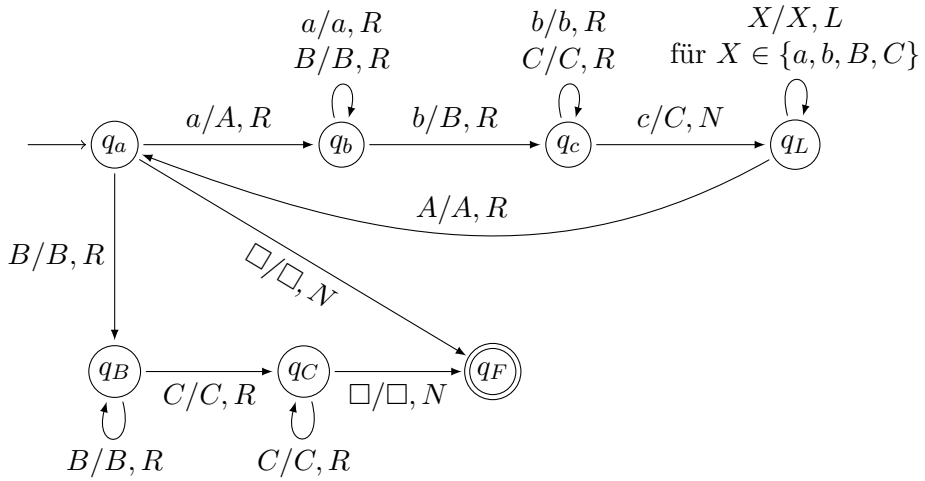
Individualaufgabe Ü9.2. (*Meine erste TM*)

Sei $\Sigma = \{a, b, c\}$. Geben Sie TM M an, so dass:

$$L(M) = \{a^n b^n c^n \mid n \geq 0\}$$

Tipp: Es gibt verschiedene Webseiten auf denen Turing Maschinen interaktiv konstruiert und simuliert werden können, z.B. `https://wimmers.github.io/turing-machine-viz/`. Beachten Sie dass diese Seite Endzustände nicht visualisiert.

Lösungsskizze. Für diese Aufgabe gibt es eine Videolösung: Link
Idee: Ersetze für jedes a je ein a, b und c durch x, wobei nach dem Ersetzen die restlichen Buchstaben der gleichen Art unverändert gelassen werden. Wenn am Ende nur noch x auf dem Band stehen, terminiere. Sonst bleibt die Berechnung in einem Nichtendzustand stecken.
Wir schreiben $\square$ für eine leere Bandzelle.
Sei TM $M = (\{q_a, q_b, q_c, q_L, q_B, q_C, q_F\}, \{a, b, c\}, \Sigma \cup \{A, B, C, \square\}, \delta, q_a, \square, \{q_F\})$.



Übung und Nachbereitung

Fokusaufgabe Ü9.3. (*Zweierpotenz: TM & WHILE*)

(a) Sei $f : \{0,1\}^* \to \{0,1\}$ die Funktion, die berechnet, ob ein Wort die Binärdarstellung einer Zweierpotenz ist[1]:

$$f(w) = \begin{cases} 1 & \text{wenn } (w)_2 = 2^n \text{ für ein } n \\ 0 & \text{sonst} \end{cases}$$

Geben Sie (graphisch) eine TM an, die $f(w)$ berechnet.
Erinnerung: Damit eine TM eine Funktion berechnet, muss das Band nach der Berechnung nur noch die Ausgabe enthalten und der Kopf der TM muss auf das erste Zeichen der Ausgabe zeigen.

(b) Geben Sie ein WHILE-Programm an, das berechnet, ob x_1 (die Eingabe) eine Zweierpotenz ist.
Erinnerung: Es gibt bei WHILE-Programmen (und GOTO-Programmen) kein Band und keinen direkten Zugriff auf die Binärrepräsentation der Eingabe. Nach dem Terminieren des Programms muss die Ausgabe in x_0 stehen.

Übungsaufgabe Ü9.4. (*GOTO → WHILE*)

(a) Welche Ausgabe produziert das folgende GOTO-Programm für die Eingaben $16, 3$ und 42? Was macht das Programm im Allgemeinen für die Eingabe $x_1 > 0$? Terminiert es immer?

[1] Führende Nullen sind erlaubt: $(0100)_2 = (100)_2 = 4$

M_1: $x_0 := x_0 + 1$
M_2: **if** $x_1 = 1$ **goto** M_{16}
M_3: $x_2 := x_1$
M_4: $x_3 := 0$
M_5: $x_4 := 4$
M_6: **if** $x_2 = 1$ **goto** M_{12}
M_7: **if** $x_2 = 0$ **goto** M_{14}
M_8: $x_2 := x_2 - 2$
M_9: $x_3 := x_3 + 1$
M_{10}: $x_4 := x_4 + 6$
M_{11}: **goto** M_6
M_{12}: $x_1 := x_4$
M_{13}: **goto** M_1
M_{14}: $x_1 := x_3$
M_{15}: **goto** M_1
M_{16}: `HALT`

(b) Übersetzen Sie das GOTO-Programm mit Hilfe der Konstruktion aus der Vorlesung (Satz 5.24, Folie 253f) in ein WHILE-Programm. Nutzen Sie diese Vorlage:

$pc :=$
while **do**
if $pc = 1$ **then** **end**
if $pc = 2$ **then** **end**
if $pc = 3$ **then** **end**
if $pc = 4$ **then** **end**
if $pc = 5$ **then** **end**
if $pc = 6$ **then** **end**
if $pc = 7$ **then** **end**
if $pc = 8$ **then** **end**
if $pc = 9$ **then** **end**
if $pc = 10$ **then** **end**
if $pc = 11$ **then** **end**
if $pc = 12$ **then** **end**
if $pc = 13$ **then** **end**
if $pc = 14$ **then** **end**
if $pc = 15$ **then** **end**
if $pc = 16$ **then** **end**
end

Lösungsskizze.

(a)
- Eingabe $x_1 = 16$ produziert Ausgabe $x_0 = 5$
  $(16 \to 8 \to 4 \to 2 \to 1)$
- Eingabe $x_1 = 3$ produziert Ausgabe $x_0 = 8$
  $(3 \to 10 \to 5 \to 16 \to \cdots \to 1)$
- Eingabe $x_1 = 42$ produziert Ausgabe $x_0 = 9$
  $(42 \to 21 \to 64 \to 32 \to 16 \to \cdots \to 1)$

Das Programm berechnet die Länge der Collatz-Sequenz für die positive Zahl x_1. Die Collatz-Sequenze entsteht durch wiederholtes Anwenden der Funktion

$$f(x) = \begin{cases} \frac{x}{2} & \text{wenn } x \equiv 0 \pmod 2 \\ 3x + 1 & \text{sonst} \end{cases}$$

bis der Wert 1 erreicht wird.[2] Es ist unbekannt, ob die Sequenz / das Programm immer terminiert.

(b) Lösungsvorschlag:

[2] Es gibt auch Definitionen der Sequenz, bei denen die Sequenz bei 1 weitergeht und somit die Sequenz $1 \to 4 \to 2 \to 1$ unendlich oft wiederholt wird.

$pc := 1$
while $pc \neq 0$ **do**
  if $pc = 1$ **then** $x_0 := x_0 + 1;\ pc := pc + 1$ **end**
  if $pc = 2$ **then if** $x_1 = 1$ **then** $pc := 16$ **else** $pc := pc + 1$ **end end**
  if $pc = 3$ **then** $x_2 := x_1;\ pc := pc + 1$ **end**
  if $pc = 4$ **then** $x_3 := 0;\ pc := pc + 1$ **end**
  if $pc = 5$ **then** $x_4 := 4;\ pc := pc + 1$ **end**
  if $pc = 6$ **then if** $x_2 = 1$ **then** $pc := 12$ **else** $pc := pc + 1$ **end end**
  if $pc = 7$ **then if** $x_2 = 0$ **then** $pc := 14$ **else** $pc := pc + 1$ **end end**
  if $pc = 8$ **then** $x_2 := x_2 - 2;\ pc := pc + 1$ **end**
  if $pc = 9$ **then** $x_3 := x_3 + 1;\ pc := pc + 1$ **end**
  if $pc = 10$ **then** $x_4 := x_4 + 6;\ pc := pc + 1$ **end**
  if $pc = 11$ **then** $pc := 6$ **end**
  if $pc = 12$ **then** $x_1 := x_4;\ pc := pc + 1$ **end**
  if $pc = 13$ **then** $pc := 1$ **end**
  if $pc = 14$ **then** $x_1 := x_3;\ pc := pc + 1$ **end**
  if $pc = 15$ **then** $pc := 1$ **end**
  if $pc = 16$ **then** $pc := 0$ **end**
end