Prof. Javier Esparza
Philipp Czerner, Kevin Kappelmann, Valentin Krasotin

Technische Universität München
Lehrstuhl für Theoretische Informatik

Einführung in die Theoretische Informatik

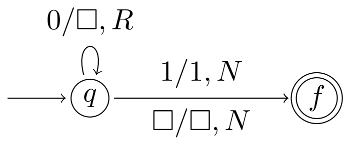
Sommersemester 2024 – Übungsblatt 9

- 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.

Notation von TMs: Bei Turing Maschinen verwendet man eine zu PDAs ähnliche graphische Notation: Sei $M = (\{q, f\}, \{0, 1\}, \{0, 1, \Box\}, \delta, q, \Box, \{f\})$ eine TM mit δ:

$$\begin{aligned} \delta(q, \Box) &= (f, \Box, N) \\ \delta(q, 0) &= (q, \Box, R) \\ \delta(q, 1) &= (f, 1, N) \end{aligned}$$

Diese Maschine entfernt führende Nullen. Nun schreibt man auf die Transitionen $\alpha/\beta, D$ mit $\alpha, \beta \in \Gamma$ und $D \in \{L, N, R\}$. Dies bedeutet, dass der Bandbuchstabe α an der Kopf Position steht und durch β ersetzt wird. Danach bewegt sich der Kopf nach links (L), rechts (R) oder gar nicht (N). Graphisch ist dies:



Vorbereitung (vor der Übung selbständig zu bearbeiten)

Vorbereitungsaufgabe 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

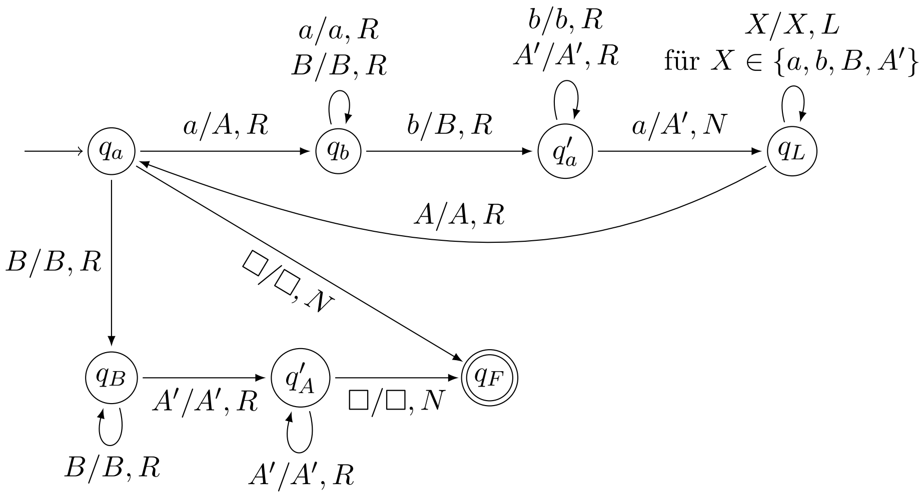
Vorbereitungsaufgabe Ü9.2. (*Meine erste TM*)

Sei $\Sigma = \{a, b\}$. Geben Sie eine TM M an, sodass:

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

Tipp: Es gibt verschiedene Webseiten auf denen Turing Maschinen interaktiv konstruiert und simuliert werden können, z.B. `https://turingmachinesimulator.com/`.

Lösungsskizze. **Idee:** Ersetze für jedes a je ein a, b und a durch einen Marker, wobei nach dem Ersetzen die restlichen Buchstaben der gleichen Art unverändert gelassen werden. Wenn am Ende nur noch Marker auf dem Band stehen, terminiere. Sonst bleibt die Berechnung in einem Nichtendzustand stecken.
Wir schreiben $\square$ für eine leere Bandzelle.
Sei TM $M = (\{q_a, q_b, q'_a, q_L, q_B, q'_A, q_F\}, \{a, b\}, \Sigma \cup \{A, B, A', \square\}, \delta, q_a, \square, \{q_F\})$.



Übung und Nachbereitung

Übungsaufgabe Ü9.3. (*TM Berechnung*)

(a) Konstruieren Sie eine Turingmaschine $T = (Q, \Sigma, \Gamma, q_0, \square, \delta, F)$, mit $\Sigma = \{|\}$, die eine eingegebene Strichzahl verdoppelt.

(b) (Optional) In der Vorlesung haben Sie auf `https://turingmachinesimulator.com/`, Beispiel "Duplicate Binary String", gesehen, wie eine Maschine für ein Wort $w \in \Sigma^* = \{0, 1\}^*$ das Wort ww^R erzeugt. Modifizieren Sie die Maschine auf der Website, sodass sie stattdessen das Wort ww erzeugt.

Lösungsskizze. Die Berechnung erfolgt, indem jeweils ein Strich am linken Anfang der Strichfolge durch eine Markierung x ersetzt wird und anschließend am rechten Ende der Strichfolge eine Markierung y angefügt wird. Falls keine Strichzeichen mehr vorhanden sind, werden alle Zeichen in Striche umgewandelt.

Sei $T = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ mit $Q = \{q_0, q_1, q_2, q_3, q_4, q_f\}$, $\Sigma = \{|\}$, $\Gamma = \{|, x, y, \square\}$ und $F = \{q_f\}$.

Die Zustände und Zeichenmengen kann man der folgenden Tabelle der Übergangsfunktion δ entnehmen.

Übergang			Kommentar
$\delta(q_0, \square)$	$\rightarrow$	$(q_3, \square, L)$	Nichts mehr zu verdoppeln
$\delta(q_0, \vert)$	$\rightarrow$	(q_1, x, R)	$\vert$ wird verdoppelt, x ist ein Hilfszeichen
$\delta(q_0, y)$	$\rightarrow$	(q_0, y, R)	Überspringe Hilfszeichen y
$\delta(q_1, \square)$	$\rightarrow$	(q_2, y, N)	Schreibe $\vert$ an rechten Rand, markiert durch Hilfszeichen y
$\delta(q_1, \vert)$	$\rightarrow$	$(q_1, \vert, R)$	An den rechten Rand gehen
$\delta(q_1, y)$	$\rightarrow$	(q_1, y, R)	An den rechten Rand gehen
$\delta(q_2, \vert)$	$\rightarrow$	$(q_2, \vert, L)$	Zurück zum nächsten $\vert$
$\delta(q_2, y)$	$\rightarrow$	(q_2, y, L)	Zurück zum nächsten $\vert$
$\delta(q_2, x)$	$\rightarrow$	(q_0, x, R)	Nächstes $\vert$ verdoppeln oder halten
$\delta(q_3, y)$	$\rightarrow$	$(q_3, \vert, L)$	Hilfszeichen durch $\vert$ ersetzen
$\delta(q_3, x)$	$\rightarrow$	$(q_3, \vert, L)$	Hilfszeichen durch $\vert$ ersetzen
$\delta(q_3, \square)$	$\rightarrow$	$(q_f, \square, R)$	Halten

Hier der Code für `https://turingmachinesimulator.com/`:

```
name: Duplicate unary string
init: q0
accept: qf

q0,_
q3,_,<

q0,1
q1,x,>

q0,y
q0,y,>

q1,_
q2,y,-

q1,1
q1,1,>

q1,y
q1,y,>

q2,1
q2,1,<

q2,y
q2,y,<

q2,x
q0,x,>

q3,y
q3,1,<

q3,x
q3,1,<

q3,_
```

qf ,_,>

Übungsaufgabe Ü9.4. (*Blau und Schlau*)

Entscheiden Sie ob die folgenden Aussagen wahr oder falsch sind. Begründen Sie ihre Behauptung. Ein ausführlicher formaler Beweis ist nicht gefordert, aber eine umfassende Begründung.

(a) Wenn f, g berechenbar sind, dann ist $f \circ g$ berechenbar.

(b) Wenn $f \circ g$ berechenbar ist, dann sind f, g berechenbar.

(c) In einem unzerstörbaren, unknackbaren, uneinsehbaren Safe befindet sich ein Zettel, auf dem eine jedem unbekannte natürliche Zahl z steht. Die Funktion

$$f(n) = \begin{cases} 1 & \text{wenn } n = z \\ 0 & \text{sonst} \end{cases}$$

ist berechenbar.

(d) Die Klasse der "ruhelosen" Turingmaschinen, die ihren Kopf in jedem Schritt bewegen müssen (d.h. für alle Transitionen $\delta(q, a) = (q', b, D)$ muss $D \neq N$ sein), akzeptiert genau die Typ-0-Sprachen.

(e) Die Klasse der nichtdeterministischen Turingmaschinen, die jede Zelle ihres Bandes höchstens zweimal ändern, akzeptiert genau die Typ-0-Sprachen. Eine Transition $\delta(q, a) = (q', b, D)$ ändert eine Zelle, falls $a \neq b$.

Lösungsskizze.

(a) Wahr. Führe zunächst den Algorithmus für g aus, dann führe den Algorithmus für f auf das Ergebnis aus.

(b) Falsch. Sei g eine beliebige unberechenbare Funktion und f die konstante Nullfunktion. Dann ist $f \circ g$ die konstante Nullfunktion und berechenbar.

(c) Wahr. Diese Funktion ist für jedes fixe z berechenbar. Hierfür ist es irrelevant welches z dies ist. f ist berechenbar, da die Eingabe nur mit einer Konstante(z) verglichen werden muss und je nach Ergebnis 0 oder 1 zurückgegeben wird.

(d) Wahr. Typ-0-Sprachen sind genau die Sprachen, die von Turingmaschinen akzeptiert werden. Jede ruhelose Turingmaschine ist offensichtlich eine Turingmaschine. Eine beliebige TM kann von einer ruhelosen TM simuliert werden, indem man jede Transition, die den Kopf nicht bewegt, durch zwei Transitionen simuliert (einen Schritt nach rechts, dann wieder einen nach links).

(e) Wahr. Jede TM, die jede Zelle nur zweimal ändert, ist eine TM. Noch zu zeigen: Jede TM M kann von einer TM M', die jede Zelle nur zweimal ändert, simuliert werden:

Zu jedem Zeitpunkt steht nur endlich viel Information auf dem Band einer Turingmaschine. Die Idee ist es nun, für jeden Schritt der originalen Turingmaschine den aktuellen Bandinhalt auf einen noch freien Platz (z.B. rechts) zu kopieren und dabei einen Schritt der TM auszuführen. Dabei wird außerdem die Kopfposition der originalen TM auf dem Band mit einem neuen Markierungszeichen vermerkt. Die Sequenz an Konfigurationen kann dabei durch Trennzeichen abgegrenzt werden.

Initial fügt die Maschine das Kopfmarkierungszeichen ein. Die Maschine kopiert dann stets die aktuelle Konfiguration Zeichen für Zeichen nach rechts. Sobald ein Zeichen kopiert werden soll, wird es mit einer Markierung versehen. Falls das Zeichen nicht an der Kopfmarkierung anliegt, wird es einfach kopiert. Anderenfalls wird es entsprechend des Überganges der originalen Turingmaschine angepasst. Die Prozedur modifiziert jede Zelle somit höchstens zweimal: einmal, um ein Zeichen zum erstem mal hinzuschreiben und ein zweites mal, um das Zeichen im nächsten Schritt als kopiert zu markieren.

In der Vorlesung wurde die Annahme gemacht, dass die Übergangsfunktion δ einer Turingmaschine folgende Eigenschaft erfüllt:

$$\delta(q, \mathsf{a}) \textit{ ist nicht definiert für alle } q \in \mathsf{F},\ \mathsf{a} \in \Gamma.$$

Sei $\mathcal{M}_A$ die Menge der Turingmaschinen, die diese Annahme erfüllen, und sei $\mathcal{M}$ die Menge aller Turingmaschinen. Es gilt somit $\mathcal{M}_A \subsetneq \mathcal{M}$.

Für $\mathsf{M} \in \mathcal{M}$ mit $\mathsf{M} = (\mathsf{Q}, \Sigma, \Gamma, \delta, \mathsf{q}_0, \Box, \mathsf{F})$ definiere:

- $\mathsf{L}_F(\mathsf{M}) := \{w \in \Sigma^* \mid \exists \alpha, \beta \in \Gamma^*, f \in \mathsf{F}.\ (\varepsilon, \mathsf{q}_0, w) \to^*{}_{\mathsf{M}} (\alpha, f, \beta)\}$.
  (Menge der Wörter, für die die Maschine einen Endzustand irgendwann besucht.)
- $\mathsf{L}_H(\mathsf{M}) := \{w \in \Sigma^* \mid \exists \alpha, \beta \in \Gamma^*, q \in \mathsf{Q}.\ (\varepsilon, \mathsf{q}_0, w) \to_M^* (\alpha, q, \beta)$ und $\delta(q, \mathit{first}(\beta))$ ist nicht definiert$\}$.
  (Menge der Wörter, für die die Maschine hält.)
- $\mathsf{L}_{HF}(\mathsf{M}) := \{w \in \Sigma^* \mid \exists \alpha, \beta \in \Gamma^*, f \in \mathsf{F}.\ (\varepsilon, \mathsf{q}_0, w) \to_M^* (\alpha, f, \beta)$ und $\delta(f, \mathit{first}(\beta))$ ist nicht definiert$\}$.
  (Menge der Wörter, für die die Maschine in einem Endzustand hält.)

Übungsaufgabe Ü9.5. (*TM Akzeptanzbedingungen*)

Begründen Sie folgende Aussagen, indem Sie eine passende Konstruktion angeben.

(a) Für jede Turing-Maschine $\mathsf{M} \in \mathcal{M}_A$ gibt es eine Turing-Maschine $\mathsf{M}' \in \mathcal{M}$ mit $\mathsf{L}_F(\mathsf{M}) = \mathsf{L}_H(\mathsf{M}')$.

(b) Für jede Turing-Maschine $\mathsf{M} \in \mathcal{M}$ gibt es eine Turing-Maschine $\mathsf{M}' \in \mathcal{M}_A$ mit $\mathsf{L}_H(\mathsf{M}) = \mathsf{L}_F(\mathsf{M}')$.

(c) Für jede Turing-Maschine $\mathsf{M} \in \mathcal{M}_A$ gibt es eine Turing-Maschine $\mathsf{M}' \in \mathcal{M}$ mit $\mathsf{L}_F(\mathsf{M}) = \mathsf{L}_F(\mathsf{M}')$.

(d) Für jede Turing-Maschine $\mathsf{M} \in \mathcal{M}$ gibt es eine Turing-Maschine $\mathsf{M}' \in \mathcal{M}_A$ mit $\mathsf{L}_F(\mathsf{M}) = \mathsf{L}_F(\mathsf{M}')$.

(e) Für jede Turing-Maschine $\mathsf{M} \in \mathcal{M}$ gibt es eine Turing-Maschine $\mathsf{M}' \in \mathcal{M}$ mit $\mathsf{L}_F(\mathsf{M}) = \mathsf{L}_H(\mathsf{M}')$.

Lösungsskizze.

(a) Füge einen neuen Fangzustand q_t ein und mache die Transitionsfunktion total für Nichtendzustände. Formal:

$$\mathsf{Q}' = \mathsf{Q} \cup \{q_t\} \qquad \begin{aligned}\delta' = &\delta \cup \{(q_t, a, q_t, a, N) \mid a \in \Gamma\} \cup \\ &\{(q, a, q_t, a, N) \mid a \in \Gamma \wedge q \in \mathsf{Q} \setminus \mathsf{F} \wedge \delta(q, a) \text{ ist undefiniert}\}\end{aligned}$$

(b) Füge einen neuen Endzustand q_f ein und mache die Transitionsfunktion total (außer für q_f). Formal:

$$\mathsf{Q}' = \mathsf{Q} \cup \{q_f\} \qquad \mathsf{F}' = \{q_f\} \qquad \delta' = \delta \cup \{(q, a, q_f, a, N) \mid a \in \Gamma \wedge q \in \mathsf{Q} \wedge \delta(q, a) \text{ ist undefiniert}\}$$

(c) Folgt sofort, da $\mathcal{M}_A \subsetneq \mathcal{M}$.

(d) Lösche alle ausgehenden Kanten von Endzuständen. Formal:

$$\delta' = \delta \setminus \mathsf{F} \times \Gamma \times \mathsf{Q} \times \Gamma \times \{L, R, N\}$$

(e) Lösche alle ausgehenden Kanten von Endzuständen und mache die Transitionsfunktion total für Nichtendzustände. Formal:

$$\begin{aligned}
\mathsf{Q}' &= \mathsf{Q} \cup \{q_t\} \\
\delta' &= (\delta \cup \{(q_t, a, q_t, a, N) \mid a \in \Gamma\} \\
&\qquad \cup \{(q, a, q_t, a, N) \mid a \in \Gamma \wedge q \in \mathsf{Q} \setminus \mathsf{F} \wedge \delta(q, a) \text{ ist undefiniert}\}) \\
&\qquad \setminus \mathsf{F} \times \Gamma \times \mathsf{Q} \times \Gamma \times \{L, R, N\}
\end{aligned}$$