Prof. Javier Esparza
Kevin Kappelmann, Simon Roßkopf, Lukas Stevens

Technische Universität München
Lehrstuhl für Theoretische Informatik

Einführung in die Theoretische Informatik

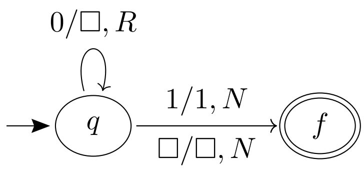
Sommersemester 2023 – Übungsblatt 9

- 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.

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:

$$\delta(q, \Box) = (f, \Box, N)$$
$$\delta(q, 0) = (q, \Box, R)$$
$$\delta(q, 1) = (f, 1, N)$$

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)

Individualaufgabe Ü9.1. (*Wichtige Begriffe*)

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

- Konversion CFG ↔ PDA
- deterministischer Kellerautomat (DPDA)
- deterministische kontextfreie Sprache (DCFL)
- Abschlusseigenschaften von DCFL
- Abschlusseigenschaften von CFL
- abzählbar / überabzählbar
- berechenbar / unberechenbar
- Church-Turing These
- nichtdeterministische / deterministische Turing-Maschine (TM)
- Konfiguration einer TM
- akzeptierte Sprache einer TM
- Turing-berechenbar

Individualaufgabe Ü9.2. (*Automata Tutor: "DPDAs"*)

Lösen Sie die Aufgaben Ü9.2 (a–b) auf Automata Tutor.[1] **Achtung:** Ihr konstruierter Automat darf nicht zu viele Zustände haben (siehe Aufgabenstellung). Wenn Sie einen ε-Übergang angeben wollen, geben Sie statt ε bitte `E` ein (siehe Hinweisbox über Canvas) und beim Entfernen eines Kellersymbols, lassen Sie den Teil hinter "/" bitte leer (Beispiel: "$\texttt{a}, X/$"). Die Simulation ist nicht aktiviert.

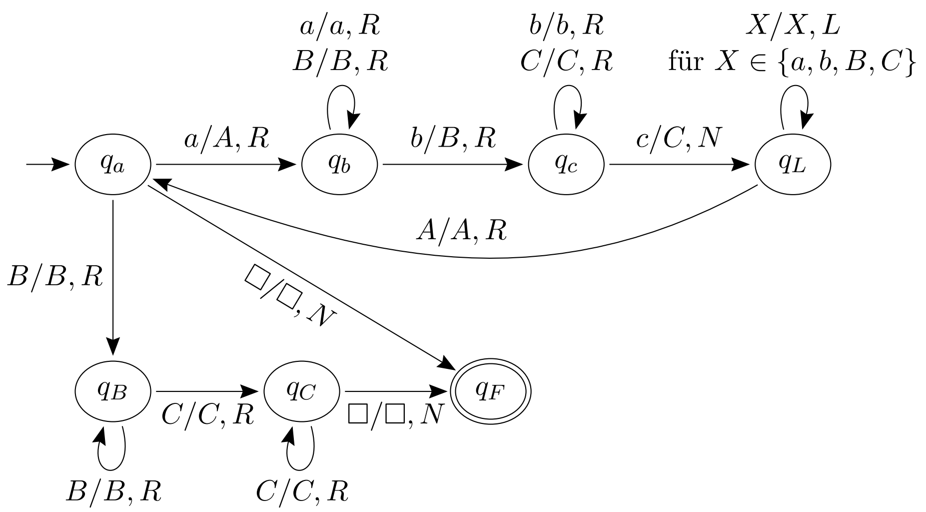
Individualaufgabe Ü9.3. (*Meine erste TM*)

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/` oder `https://turingmachinesimulator.com/`.

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 $\Box$ für eine leere Bandzelle.
Sei TM $M = (\{q_a, q_b, q_c, q_L, q_B, q_C, q_F\}, \{a, b, c\}, \Sigma \cup \{A, B, C, \Box\}, \delta, q_a, \Box, \{q_F\})$.



Übung und Nachbereitung

Übungsaufgabe Ü9.4. (*CFG $\leftrightarrow$ PDA*)

Wie Sie in der Vorlesung gesehen haben (ab Folie 203), können kontextfreie Grammatiken und Pushdown-Automaten sich gegenseitig simulieren. Wir üben nun diese Übersetzungen zwischen CFG und PDA.

[1] Wenn Sie Automata Tutor noch nicht verwendet haben, folgen Sie erst den Schritten in Ü1.2, um sich richtig zu registrieren.

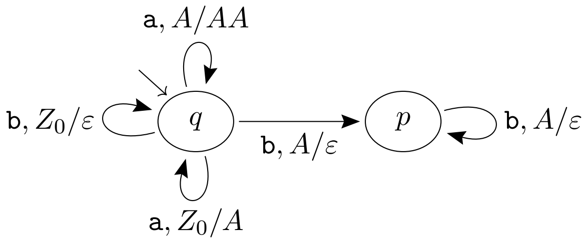
(a) Überführen Sie die folgende CFG $G = (\{S\}, \{a, b, c\}, P, S)$ mit Hilfe des Satzes der Vorlesung in einen PDA M mit $L_\varepsilon(M) = L(G)$:

$$S \to SS \mid aSb \mid bSa \mid c$$

(b) Übersetzen Sie den folgenden PDA $M = (\{q, p\}, \{a, b\}, \{Z_0, A, B\}, q, Z_0, \delta)$ in eine CFG G mit $L_\varepsilon(M) = L(G)$, wobei δ definiert ist durch:

$$\begin{array}{lll} \delta(q, a, A) = \{(q, AA)\} & \delta(q, b, A) = \{(p, \epsilon)\} & \delta(p, b, A) = \{(p, \epsilon)\} \\ \delta(q, a, Z_0) = \{(q, A)\} & \delta(q, b, Z_0) = \{(q, \epsilon)\} & \end{array}$$

Geben Sie außerdem eine Ableitung in der konstruierten Grammatik für die Wörter b und $aabb$ an.



Übungsaufgabe Ü9.5. (*TM Berechnung*)

(a) Konstruieren Sie eine Turingmaschine $T = (Q, \Sigma, \Gamma, q_0, \Box, \delta, F)$, 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.

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.)

Übungsaufgabe Ü9.6. (*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 $\mathsf{L}_F(\mathsf{M}) = \mathsf{L}_H(\mathsf{M}')$.