

Einführung in die theoretische Informatik  
Sommersemester 2018 – Übungsblatt 9

Selbstständige Vorbereitung

Bereiten Sie sich auf die Tutorgruppen selbstständig vor, indem Sie die Aufgaben 9.1 bis 9.5 ansehen. Lösen Sie insbesondere Aufgabe 9.4a, da alle weiteren Aufgaben darauf aufbauen.

AUFGABE 9.1. (Wichtige Begriffe)

Überprüfen Sie, dass Sie die Folgenden Begriffe korrekt definieren können.

Stufe A

- intuitiv berechenbar
- totale/partielle/echt partielle Funktion
- nicht-deterministische/deterministische Turing-Maschine
- abzählbar
- überabzählbar
- Konfiguration einer Turing-Maschine
- akzeptierte Sprache einer Turing-Maschine
- Turing-berechenbar

AUFGABE 9.2. (Üben mit AutomataTutor)

Auf AutomataTutor gibt es die Möglichkeit sich selber Aufgaben zu 3 verschiedenen Fragestellungen zu generieren:

Stufe C

- CYK algorithm: Durchführen des CYK Algorithmus.
- Grammar  $\rightarrow$  CNF: Üben der Umwandlung einer Grammatik in die Chomsky-Normalform.
- Words in Grammar: Finden von Wörtern, die von einer Grammatik erzeugt / nicht erzeugt werden.

Durch das individuelle Feedback können Sie selbständig üben und überprüfen, ob Sie diese wichtigen Fragestellungen richtig bearbeiten können.

Melden Sie sich dafür auf <http://vmesparza8.informatik.tu-muenchen.de/> an und klicken Sie links im Menü auf "Exercise".

AUFGABE 9.3. (Pumping Lemma für kontextfreie Sprachen üben)

Zeigen Sie durch einen Widerspruchsbeweis unter Verwendung des Pumping Lemmas für kontextfreie Sprachen, dass folgende Sprachen nicht kontextfrei sind:

Stufe C

- (a)  $L_1 = \{a^{2^i} \mid i \geq 0\}$
- (b)  $L_2 = \{a^i b^j a^k \mid j = \max\{i, k\}\}$

Definition (Alternative Akzeptanzbedingungen für Turing-Maschinen)

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

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

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  $M \in \mathcal{M}$  mit  $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$  definiere:

- $L_F(M) = \{w \in \Sigma^* \mid \exists \alpha, \beta \in \Gamma^*, f \in F. (\varepsilon, q_0, w) \rightarrow_M^* (\alpha, f, \beta)\}$ .  
(Menge der Wörter, für die die Maschine einen Endzustand irgendwann besucht.)
- $L_H(M) = \{w \in \Sigma^* \mid \exists \alpha, \beta \in \Gamma^*, q \in Q. (\varepsilon, q_0, w) \rightarrow_M^* (\alpha, q, \beta) \text{ und } \delta(q, \text{first}(\beta)) \text{ ist nicht definiert}\}$ .  
(Menge der Wörter, für die die Maschine hält.)
- $L_{HF}(M) = \{w \in \Sigma^* \mid \exists \alpha, \beta \in \Gamma^*, f \in F. (\varepsilon, q_0, w) \rightarrow_M^* (\alpha, f, \beta) \text{ und } \delta(f, \text{first}(\beta)) \text{ ist nicht definiert}\}$ .  
(Menge der Wörter, für die die Maschine in einem Endzustand hält.)

AUFGABE 9.4. (TM für Sprache)

Geben Sie für die beiden angegebenen Sprachen die jeweils passende TM  $M_i$  an.

Stufe C

$$(a) L_F(M_1) = \{a^n b^n c^n \mid n \in \mathbb{N}_0\} \quad (b) L_F(M_2) = \{a^n b^{n^2} \mid n \in \mathbb{N}_0\}$$

AUFGABE 9.5. (TM Akzeptanzbedingungen)

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

Stufe B

- (a) Für jede Turing-Maschine  $M \in \mathcal{M}_A$  gibt es eine Turing-Maschine  $M' \in \mathcal{M}$  mit  $L_F(M) = L_H(M')$ .
- (b) Für jede Turing-Maschine  $M \in \mathcal{M}$  gibt es eine Turing-Maschine  $M' \in \mathcal{M}_A$  mit  $L_H(M) = L_F(M')$ .
- (c) Für jede Turing-Maschine  $M \in \mathcal{M}_A$  gibt es eine Turing-Maschine  $M' \in \mathcal{M}$  mit  $L_F(M) = L_F(M')$ .
- (d) Für jede Turing-Maschine  $M \in \mathcal{M}$  gibt es eine Turing-Maschine  $M' \in \mathcal{M}_A$  mit  $L_F(M) = L_F(M')$ .
- (e) Für jede Turing-Maschine  $M \in \mathcal{M}$  gibt es eine Turing-Maschine  $M' \in \mathcal{M}$  mit  $L_F(M) = L_H(M')$ .

**AUFGABE 9.6.** (Aussagen über TMs)

Stufe B

Entscheiden Sie, ob die folgenden Aussagen korrekt sind und begründen Sie Ihre Antwort kurz.

- (a) Es gibt eine Turingmaschine, die den Kopf nie weiter als vier Schritte von der Startposition weg bewegt und eine unendliche Sprache akzeptiert.
- (b) Sei  $M \in \mathcal{M}_A$  eine Turingmaschine. Dann existiert eine TM  $M' \in \mathcal{M}$ , so dass  $L_F(M) = L_F(M')$  und  $M'$  hat nur einen Zustand.
- (c) Sei  $M \in \mathcal{M}_A$  eine Turingmaschine, die ihren Kopf immer nur nach links bewegt. Dann gilt:

$$L_F(M) \in \{A\Sigma^* \mid A \subseteq \Sigma\}$$

**AUFGABE 9.7.** (La-Ola-TM)

Stufe C

Wir konstruieren eine TM, die La-Ola-Wellen simuliert. Geben Sie hierzu eine deterministische TM an, welche als Eingabe ein Wort  $w \in \{u, m, o\}^*$  mit  $|w| \geq 3$  erwartet, wobei  $w = a_0 \dots a_{l-1}$  den aktuellen Zustand einer La-Ola-Welle beschreibt, wobei sich die Welle bei  $a_{l-1}$  wieder bei  $a_0$  fortsetzen sollen (mod  $l$ ). Die Buchstaben von  $w$  beschreiben die aktuelle Armhaltung (**u**nten, **m**ittig, **o**ben) des Zuschauers auf Platz  $i$ .

Die DTM soll zuerst prüfen, dass  $w$  eine zulässige La-Ola-Welle ist, d.h.

$$\begin{aligned} \forall i \in \mathbb{Z}_l. \quad & \wedge (a_i = o \rightarrow a_{i-1 \bmod l} a_i a_{i+1 \bmod l} \in L((o|m)o(o|m))) \\ & \wedge (a_i = u \rightarrow a_{i-1 \bmod l} a_i a_{i+1 \bmod l} \in L((u|m)u(u|m))) \\ & \wedge (a_i = m \rightarrow a_{i-1 \bmod l} a_i a_{i+1 \bmod l} \in L(umo|omu)) \end{aligned}$$

und anschließend die Welle um eine Position nach links verschieben, d.h. falls  $w$  zulässig ist, soll die DTM mit der Ausgabe  $\dots \square a_1 \dots a_{l-1} a_0 \square \dots$  terminieren. Falls die Eingabe jedoch nicht zulässig ist, soll die DTM mit leerem Band terminieren.

**AUFGABE 9.8.** ( $k$ -PDA und TMs)

Stufe D

Im Folgenden sagen wir, dass ein  $k$ -PDA ein PDA ist, der  $k$  Stacks zur Verfügung hat. In jedem Schritt kann der PDA in Abhängigkeit vom aktuellen Zustand, dem gelesenen Eingabezeichen und den Symbolen, die zuoberst auf jedem der  $k$  Stacks liegen, in einen neuen Zustand wechseln und jeden der  $k$  Stacks wie im Fall eines gewöhnlichen PDAs modifizieren.

- (a) Geben Sie eine formale Definition für  $k$ -PDAs an.
- (b) Geben Sie eine Sprache an, die von einem 2-PDA, aber von keinem 1-PDA akzeptiert wird.
- (c) Geben Sie eine allgemeine Übersetzung von einem PDA  $A$  in eine Turingmaschine  $M$  an, so dass  $L_\varepsilon(A) = L_F(M)$ . Verwenden Sie bei Bedarf eine TM mit mehreren Bändern.
- (d) Beschreiben Sie wie Sie das Verfahren aus (c) erweitern können, so dass ein  $k$ -PDA  $A$  in eine Turingmaschine  $M$  übersetzt werden kann.
- (e) Geben Sie eine allgemeine Übersetzung von einer Turingmaschine  $M$  in einen 2-PDA  $A$  an, so dass  $L_F(M) = L_F(A)$ .
- (f) Zeigen Sie unter Verwendung der vorherigen Ergebnisse, dass jeder  $k$ -PDA  $A$  ( $k \geq 3$ ) von einem 2-PDA  $A'$  simuliert werden kann, d.h.  $L_\varepsilon(A) = L_\varepsilon(A')$ . Somit können beliebig viele Stacks immer durch genau zwei Stacks simuliert werden.
- (g) Skizzieren Sie eine direkte Übersetzung von einem  $k$ -PDA zu einem 2-PDA an.