Institut für Theoretische Informatik
Lehrstuhl Prof. Dr. D. Wagner

Übungsblatt 7

Vorlesung Theoretische Grundlagen der Informatik im WS 21/22

Ausgabe: 28. Januar 2022
Abgabe: 11. Februar 2022 (digital im ILIAS)

Bitte bearbeiten Sie die Aufgaben **handschriftlich** und laden Sie eine eingescannte PDF-Version im Übungsmodul Ihrer ILIAS-Tutoriumsgruppe hoch! Beschriften Sie Ihren handschriftlichen Aufschrieb gut sichtbar mit Name und Matrikelnummer. Nicht handschriftliche oder unbeschriftete Abgaben werden nicht akzeptiert!

Aufgabe 1

(3 + 1 + 1 = 5 Punkte)

Sei $G = (\Sigma, V, S, R)$ mit $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ und $V = \{S, A, B, C, D, E\}$ die durch folgende Regelmenge gegebene Grammatik:

$$\begin{aligned} S &\to A \mid E\mathsf{a} \\ A &\to B\mathsf{a} \mid \mathsf{c}E\mathsf{c} \\ B &\to AD \mid \mathsf{b}E \mid \varepsilon \\ C &\to C\mathsf{a} \mid AB \\ D &\to \mathsf{c} \\ E &\to E\mathsf{b} \mid ED \end{aligned}$$

(a) Identifizieren Sie alle nutzlosen Variablen in G mit dem Verfahren aus der Vorlesung. Geben Sie die Grammatik G' an, die durch Entfernen der nutzlosen Variablen entsteht.

(b) Welche Sprache erzeugt G? Welchen Chomsky-Typ hat $L(G)$? Ist $L(G)$ endlich?

(c) Ist G' minimal in dem Sinne, dass es keine Grammatik mit weniger Variablen gibt, die $L(G')$ erzeugt? Begründen Sie Ihre Antwort.

Aufgabe 2

(2 + 3 = 5 Punkte)

Sei $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$. Geben Sie zu den folgenden Sprachen jeweils einen deterministischen Kellerautomaten an, der die Sprache erkennt. Erklären Sie jeweils kurz die Arbeitsweise des Kellerautomaten.

(a) $L_1 = \{\mathsf{a}^i\mathsf{b}^j\mathsf{c}^k \mid i, j, k \in \mathbb{N}_0, i + k = j\}$

(b) $L_2 = \{uv \mid u \in \{\mathsf{a},\mathsf{b}\}^*, v \in \{\mathsf{c}\}^* \text{ und } |u|_\mathsf{a} + 2 \cdot |u|_\mathsf{b} = 5 \cdot |v|_\mathsf{c}\}$, wobei $|w|_x$ die Anzahl der Vorkommen von $x \in \Sigma$ in $w \in \Sigma^*$ bezeichnet.

Aufgabe 3 (1 + 1 + 2 + 2 + 1 = 7 Punkte)

Ein *Input-Driven Deterministic Pushdown Automaton (IDPA)* ist ein etwas modifizierter deterministischer Kellerautomat. Dabei ist für jedes Symbol des Alphabets Σ vorher festgelegt, ob bei der Abarbeitung des Symbols etwas auf den Stack gelegt oder vom Stack gelöscht wird. Zusätzlich zum Alphabet Σ betrachten wir also ein *Pushdown-Alphabet* $\widehat{\Sigma} = (\Sigma_{push}, \Sigma_{pop}, \Sigma_{-})$ mit $\Sigma = \Sigma_{push} \dot\cup \Sigma_{pop} \dot\cup \Sigma_{-}$. Das Pushdown-Alphabet $\widehat{\Sigma}$ partitioniert Σ in drei untereinander paarweise disjunkte Mengen.

Formal besteht ein IDPA über dem Pushdown-Alphabet $\widehat{\Sigma} = (\Sigma_{push}, \Sigma_{pop}, \Sigma_{-})$ wie ein deterministischer Kellerautomat aus $(Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)$ mit $\Sigma = \Sigma_{push} \dot\cup \Sigma_{pop} \dot\cup \Sigma_{-}$. Außerdem hat ein Zustandsübergang $\delta(q, a, Z) = (q', \gamma)$ mit $q, q' \in Q$, $a \in \Sigma$ und $\gamma \in \Gamma^*$ die folgenden Einschränkungen:

- Ist $a \in \Sigma_{push}$, gilt $\gamma = Z'Z$ für ein $Z' \in \Gamma \setminus \{Z_0\}$, es wird also genau ein neues Symbol auf den Stack gelegt.
- Ist $a \in \Sigma_{pop}$, gilt $Z \neq Z_0$ und $\gamma = \varepsilon$, es wird also das oberste Symbol auf dem Stack gelöscht, außer es ist das Initialisierungssymbol.
- Ist $a \in \Sigma_{-}$, gilt $\gamma = Z$, der Stack bleibt also unverändert.

Der IDPA hat keine ε-Übergänge und akzeptiert mit akzeptierendem Endzustand.

(a) Beschreiben Sie, wie die Sprache der korrekten Klammerausdrücke über dem Alphabet $\Sigma = \{[,]\}$ von einem IDPA erkannt werden kann.

(b) Geben Sie eine kontextfreie Sprache an, die von keinem IDPA erkannt werden kann. Begründen Sie.

(c) Seien L_1 und L_2 zwei Sprachen über einem Alphabet Σ, die von den IDPAs M_1 bzw. M_2 über demselben Pushdown-Alphabet $\widehat{\Sigma} = (\Sigma_{push}, \Sigma_{pop}, \Sigma_{-})$ erkannt werden. Zeigen Sie, dass es einen IDPA $M_\cap$ gibt, der $L_1 \cap L_2$ erkennt.

(d) Die Menge der Sprachen, die von einem deterministischen Kellerautomaten erkannt werden können, sind nicht unter Durchschnittbildung abgeschlossen. Erklären Sie kurz, warum die Konstruktion aus (c) für IDPAs funktioniert, aber nicht für deterministische Kellerautomaten.

(e) Begründen Sie, dass die Menge der Sprachen, die von IDPAs erkannt werden, auch unter Komplement und Vereinigung abgeschlossen sind.

Aufgabe 4 (2 + 4 (+2) = 6 (8) Punkte)

(a) Sei $k \in \mathbb{N}$ eine beliebige, aber feste Konstante. Ein k-beschränkter Kellerautomat ist ein deterministischer Kellerautomat, dessen Stack zu jedem Zeitpunkt höchstens k Symbole enthalten darf. Zeigen Sie, dass k-beschränkte Kellerautomaten genau die Menge der regulären Sprachen erkennen können.

(b) Ein Queue-Automat ist wie ein Kellerautomat definiert, besitzt aber statt eines Stacks eine Queue. Das heißt, bei einem Abarbeitungsschritt kann das vorderste Symbol in der Queue entfernt und neue Symbole hinten in der Queue eingefügt werden. Zusätzlich besitzt der Queue-Automat einen Queue-Lesekopf, der in jedem Schritt ein Element der Queue lesen und sich nach links/rechts bewegen oder stehen bleiben kann.

Welche Klasse von Sprachen können Queue-Automaten erkennen? Beweisen Sie Ihre Behauptung.

(c) *Bonusaufgabe:* Zeigen Sie, dass ein Queue-Automat, der keinen Queue-Lesekopf hat (also nur das vorderste Zeichen lesen kann), dieselbe Klasse von Sprachen wie der Automat in (b) erkennt.

Aufgabe 5

(2 + 3 + 2 = 7 Punkte)

Eine Instanz des Post'schen Korrespondenzproblems (PKP) ist eine endliche Menge von Wortpaaren $K = \{(x_1, y_1), \dots (x_n, y_n)\}$ über einem endlichen Alphabet Σ. Es gilt $x_i \neq \varepsilon$ und $y_i \neq \varepsilon$ für alle $i = 1, \dots, n$. Eine Lösung ist eine Folge von Indizes $i_1, \dots, i_k \in \{1, \dots, n\}$, für die

$$x_{i_1} \dots x_{i_k} = y_{i_1} \dots y_{i_k}$$

gilt. Dieses Problem ist nicht entscheidbar.

Im Folgenden sollen Sie zeigen, dass es nicht entscheidbar ist, ob der Schnitt von zwei gegebenen kontextfreien Sprachen wieder kontextfrei ist. Betrachten Sie dazu die folgende Sprache L_K über dem Alphabet $\Gamma = \Sigma \cup \{\#, 1, \dots, n\}$ zu einer gegebenen PKP-Instanz K:

$$L_K = \left\{ i_k \dots i_1 x_{i_1} \dots x_{i_k} \# y^R_{i_k} \dots y^R_{i_1} i_1 \dots i_k \;\middle|\; x_{i_1} \dots x_{i_k} = y_{i_1} \dots y_{i_k},\ i_1, \dots, i_k \in \{1, \dots, n\} \right\}$$

(a) Geben Sie zwei kontextfreie Sprachen L_1 und L_2 an, sodass $L_K = L_1 \cap L_2$ gilt. Zeigen Sie, dass L_1 und L_2 kontextfrei sind.

(b) Zeigen Sie, dass L_K genau dann kontextfrei ist, wenn die PKP-Instanz K keine Lösung hat. Sie können dabei verwenden, dass ein Wort $i_k \dots i_1 x_{i_1} \dots x_{i_k} \# y^R_{i_k} \dots y^R_{i_1} i_k \dots i_1$ genau dann in L_k ist, wenn $i_1 \dots i_k$ eine Lösung von K ist.

Hinweis 1: Verwenden Sie für eine Beweisrichtung das Pumping-Lemma für kontextfreie Sprachen.

Hinweis 2: Hat die Instanz K eine Lösung $i_1 \dots i_k$, dann sind auch $(i_1 \dots i_k)^m$ für alle $m \in \mathbb{N}_+$ Lösungen von K, da wir Lösungen konkatenieren können, um neue Lösungen zu erhalten.

(c) Folgern Sie, dass es unentscheidbar ist, ob der Schnitt von zwei gegebenen kontextfreien Sprachen kontextfrei ist.

Aufgabe 6

(1 + 1 = 2 Punkte)

Die folgende Tabelle gibt eine Häufigkeitsverteilung für die Zeichen des Alphabets $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{e}, \mathsf{f}, \mathsf{g}, \mathsf{h}\}$ an.

a	b	c	d	e	f	g
12%	22%	5%	15%	14%	28%	4%

(a) Geben Sie einen Huffman-Code für die gegebene Verteilung an. Verwenden Sie die `0`, wenn Sie zu einem häufigeren Buchstaben absteigen und die `1`, wenn Sie zu einem weniger häufigen Buchstaben absteigen.

(b) Was ist die erwartete Länge der Kodierung eines Wortes in Abhängigkeit von dessen Länge n, wenn sie die obige Verteilung und Ihre Huffman-Kodierung zugrunde legen?