Theoretische Informatik 1
Übungsblatt zur Klausurvorbereitung

Sebastian Muskalla
Prof. Dr. Roland Meyer
TU Braunschweig
Wintersemester 2018/19

Ausgabe: 23.01.2019 Abgabe: keine

Wir raten Ihnen, die folgenden Aufgaben als Vorbereitung auf die Abschlussklausur am 7. Februar zu bearbeiten. Dass Blatt soll jedoch weder abgegeben werden, noch wird es bepunktet.

Die Aufgaben werden im Rahmen von zwei Fragestunden in der Woche der Abschlussklausur besprochen:

Montag, 4. Februar, ab 13:15, Raum MS3.1 in der Mendelssohnstr. 2-3 ,

Dienstag, 5. Februar, ab 15:00, Raum PK2.2 .

Dieses Übungsblatt beinhaltet Übungsaufgaben zum Stoff der gesamten Vorlesung.

A) Die Aufgaben 1 bis 3 beziehen sich auf die vorletzte Vorlesungswoche (21. & 22.02).

B) Die Aufgaben 4 bis 6 beziehen sich auf die letzte Vorlesungswoche (28. & 29.02).

C) Die Aufgaben 7 bis 9 beziehen sich auf Teil I. der Vorlesung (Fixpunkte, Verbände, …).

D) Die Aufgaben 10 bis 13 beziehen sich auf Teil II. der Vorlesung (Reguläre Sprachen).

E) Die Aufgaben 14 bis 16 beziehen sich auf Teil III. der Vorlesung (Kontextfreie Sprachen).

Teilweise sind die Aufgaben der Klausur aus dem Sommersemester 2017 entnommen. Die gesamten Altklausuren können Sie über die Seite der Fachgruppe Informatik, `https://fginfo.cs.tu-bs.de/wiki/doku.php?id=infos:studium:pruefungsprotokolle` abrufen. Das Lösen der Altklausuren ist eine weitere sinnvolle Übung zur Klausurvorbereitung.

A) Pushdown-Automaten I

Aufgabe 1: Greibach Normalform

Berechnen Sie zur Grammatik $G = (\{S, A, B, C, D\}, \{a, b\}, P, S)$ mit den Produktionen

$$S \to DA \mid BA \mid b$$
$$A \to AC \mid a$$
$$D \to AB$$
$$B \to b$$
$$C \to a$$

eine äquivalente Grammatik in Greibach-Normalform unter Verwendung des in der Vorlesung vorgestellten Verfahrens.

Aufgabe 2: Normalisierte Pushdowns

In dieser Aufgabe betrachten wir Pushdown-Automaten der Form $M = (Q, \Sigma, \Gamma, q_0, \#, \delta, Q_F)$, die mit Endzuständen akzeptieren, jedoch trotzdem mit einem **initialen Stacksymbol** # ausgestattet sind. Die Initialkonfiguration eines solchen Automaten ist also $(q_0, \#)$, und die akzeptierenden Konfigurationen sind von der Form (q_f, α) mit $q_f \in Q_F$ und $\alpha \in \Gamma^*$ beliebig.

Wir nennen einen solchen Automaten **normalisiert**, wenn alle Transitionen

- immer genau ein Symbol vom Stack nehmen und
- immer höchstens zwei Symbole auf den Stack pushen.

Zeigen Sie, dass die Sprachen, die von normalisierten Pushdowns akzeptiert werden genau die Sprachen sind, die von gewöhnlichen Pushdowns akzeptiert werden.

Hinweis: Konstruieren Sie für jeden PDA einen sprachäquivalenten normalisierten PDA und anders herum.

Aufgabe 3: Von Grammatik zu Pushdown

In der Vorlesung haben Sie ein Verfahren kennen gelernt, um aus einer Grammatik G in Greibach-Normalform einen sprachäquivalenten Pushdown-Automaten zu konstruieren. In dieser Aufgabe betrachten wir eine alternative Konstruktion, die ohne die Greibach-Normalform auskommt.

a) Es sei $G = (N, \Sigma, P, S)$ eine kontextfreie Grammatik. Die **Starke-Linksableitung-Relation** $\Rightarrow_{SL}$ auf Satzformen erlaubt es uns (im Gegensatz zur normalen Ableitungsrelation), nur das linkeste Nichtterminal zu ersetzen. Es gilt also $\alpha \Rightarrow_{SL} \beta$ gdw. $\alpha = wX\alpha'$ für ein $X \in N$, $w \in \Sigma^*$, $\beta = w\eta\beta'$ und es die Regel $X \to \eta$ in P gibt.

Zeigen Sie, dass sich die Worte, die sich mit starken Linksableitungen erzeugen lassen genau die Worte aus der Sprache von G sind, also

$$\mathcal{L}(G) = \left\{ w \in \Sigma^* \mid S \Rightarrow_{SL}{}^* w \right\} .$$

b) Nun wollen wir einen zu G äquivalenten Pushdown-Automaten konstruieren, der seinen Stack nutzt, um starke Linksableitungen zu simulieren. Hierzu verwenden wir einen Automaten, der mit leerem Stack akzeptiert und ein initiales Stacksymbol hat.

Zeigen Sie, wie man zur Grammatik G einen solchen Pushdown-Automaten $M = (\{q\}, \Sigma, N \cup \Sigma \cup \{\#\}, q, \#, \delta)$ konstruieren kann, sodass $\mathcal{L}(G) = \mathcal{L}(M)$ gilt. Beachten Sie, dass M nur einen Kontrollzustand hat und auf seinem Stack Satzformen speichert.

Automat M soll Linksableitungen auf dem Stack simulieren: Wenn $S \Rightarrow_{SL}{}^* w.X.\alpha$ gemäß G, dann soll P eine Transitionsfolge $(q, \#) \xrightarrow{w} (q, \#.\text{reverse}(\alpha).X)$ zulassen, die das Terminalpräfix $w \in \Sigma^*$ ausgibt und den Rest der Satzform (in der richtigen Reihenfolge) auf dem Stack speichert.

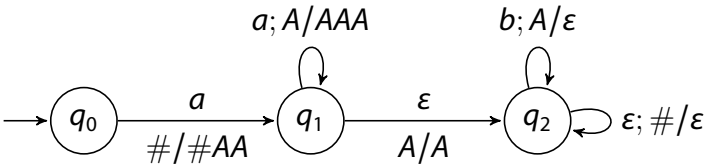
B) Pushdown-Automaten II

Aufgabe 4: Pumping Lemma

Es sei $\Sigma = \{a\}$. Zeigen Sie, dass die Sprache $L = \{a^{2^n} \mid n \in \mathbb{N}\}$ nicht kontextfrei ist.

Aufgabe 5: Von Pushdown zu Grammatik

Gegeben sei der folgenden PDA M, der mit leerem Stack akzeptiert.



Nutzen Sie das Verfahren aus der Vorlesung um eine kontextfreie Grammatik G mit $\mathcal{L}(G) = \mathcal{L}(M)$ zu konstruieren.

Welche Sprache wird von M erzeugt?

Hinweis: Ein Nichtterminal der Form (q, A, q') erzeugt ein Terminalwort w genau dann, wenn M, gestartet auf q mit Input w und initialem Stacksymbol A, nach q' übergehen kann und dort leeren Stack hat.

Aufgabe 6: Zwei-Stack-Pushdowns

Ein **Zwei-Stack-Pushdown-Automat** (2PDA) ist ein Tupel $(Q, \Sigma, \Gamma, q_0, \delta, Q_F)$, mit einer endlichen Menge von Zuständen Q, einem Eingabealphabet Σ, einem Stackalphabet Γ, einem Startzustand $q_0 \in Q$, einer Menge von Endzuständen $Q_F \subseteq Q$ und einer Transitionsrelation, die es erlaubt, zwei Stacks zu manipulieren:

$$\delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times \underbrace{\left((\Gamma \cup \{\varepsilon\}) \times \Gamma^*\right)}_{\text{Stack 1}} \times \underbrace{\left((\Gamma \cup \{\varepsilon\}) \times \Gamma^*\right)}_{\text{Stack 2}} \times Q$$

Konfigurationen sind von der Form $(q, \alpha, \beta) \in Q \times \Gamma^* \times \Gamma^*$. Die Transitionsrelation auf Konfigurationen und Akzeptanz sind analog zu Pushdown-Automaten definiert.

Das Ziel dieser Aufgabe ist es, zu beweisen, dass 2PDAs eine ausdrucksstärkere Sprachklasse definieren als PDAs.

a) Konstruieren Sie einen 2PDA für die Sprache $\mathcal{L} = \{a^n.b^k.c^n.d^k \mid n, k \in \mathbb{N}\}$.

b) Beweisen Sie, dass es keinen PDA geben kann, der die Sprache $\mathcal{L}$ akzeptiert.

C) Verbände & Datenflussanalyse

Aufgabe 7: Partielle Ordnungen und Verbände

Wir definieren $\mathbb{N}_\omega = \mathbb{N} \cup \{\omega\} = \{0, 1, 2, \ldots\} \cup \{\omega\}$. Wir betrachten die partielle geordnete Menge $(\mathbb{N}_\omega^2, \sqsubseteq)$ mit $(x, y) \sqsubseteq (x', y')$ wenn (1) $x = x'$ oder $x' = \omega$, und (2) $y = y'$ oder $y' = \omega$.

a) Zeichnen Sie das Hasse-Diagramm zu $(\mathbb{N}_\omega^2, \sqsubseteq)$, eingeschränkt auf die Werte aus $\{0, 1, 2, \omega\}^2$.

b) Bestimmen Sie den Join $(0, 0) \sqcup (0, 2)$, den Join $(1, \omega) \sqcup (\omega, 2)$ und den Meet $(1, \omega) \sqcap (\omega, 2)$.

c) Ist $(\mathbb{N}_\omega^2, \sqsubseteq)$ ein Verband?

Aufgabe 8: Distributivität

Seien $(D, \leqslant)$ ein Verband und $x, y \in D$.

a) Zeigen Sie: Ist $f : D \to D$ monoton, so gilt $f(x \sqcup y) \geqslant f(x) \sqcup f(y)$.

b) $f : D \to D$ heißt **distributiv**, falls $f(x \sqcup y) = f(x) \sqcup f(y)$ für alle $x, y \in D$.
Zeigen Sie: Falls f distributiv ist, so ist f auch monoton.

Aufgabe 9: Reachable Values-Analyse

Betrachten Sie das links stehende Bool'sche Programm. Untersuchen Sie für jeden Block, welche Belegungen die Variablen am Eingang annehmen können. Benutzen Sie dazu das Datenflusssystem $S = (G, (D, \subseteq), i, TF)$, mit $D = \mathcal{P}(\{false^x, true^x, false^y, true^y\})$ und $i = \{false^x, false^y\} \in D$, und gehen Sie wie folgt vor:

```
1:  [x := true]^1
2:  [y := true]^2
3:  while [x]^3 do
4:      [y := ¬x]^4
5:      if [¬y]^5 then
6:          [x := ¬x]^6
7:      else
8:          [x := ¬y]^7
9:      end if
10: end while
11: [skip]^8
```

a) Zeichnen Sie den zugehörigen Kontrollflussgraphen G.

b) Geben Sie die Familie der monotonen Transferfunktionen

$$TF = \{f_i : D \to D \mid i \in \{1, \ldots, 8\}\}$$

an, wobei f_i den Effekt des Blocks mit Label i im Abstrakten imitiert. Hierbei überapproximieren wir und berücksichtigen die Bedingungen (z.B. von Schleifen) nicht.

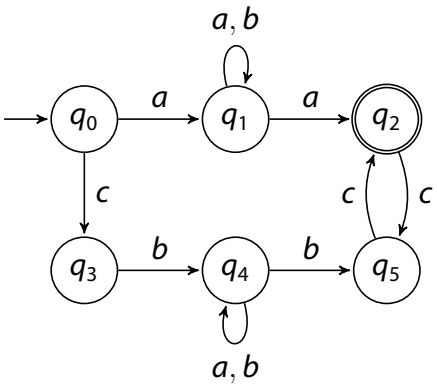
c) Geben Sie das durch S induzierte Gleichungssystem an.

d) Lösen Sie das Gleichungssystem mit Hilfe des Satzes von Kleene.

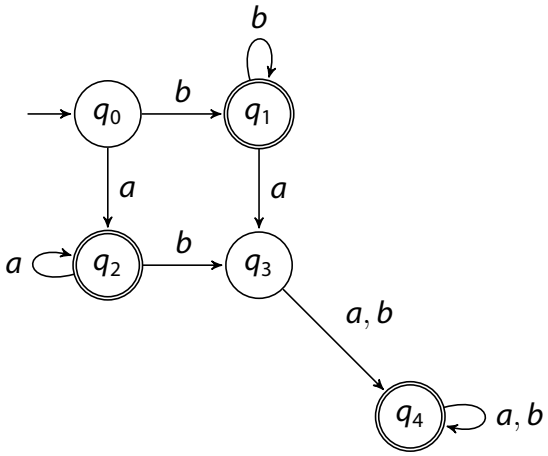
D) Reguläre Sprachen

Aufgabe 10: Algorithmik regulärer Sprachen

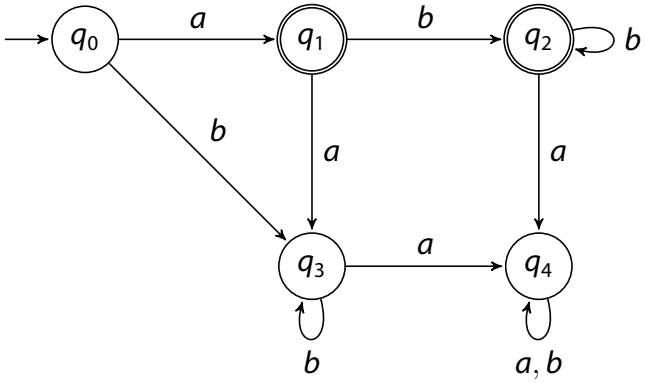
a) Berechnen Sie zum folgenden NFA über dem Alphabet $\Sigma = \{a, b, c\}$ einen sprachäquivalenten DFA unter Verwendung der Rabin-Scott-Potenzmengen-Konstruktion aus der Vorlesung.



b) Bestimmten Sie zum folgenden endlichen Automaten über dem Alphabet $\Sigma = \{a, b\}$ einen sprachäquivalenten regulären Ausdruck unter Verwendung von Ardens Lemma und des Verfahrens aus der Vorlesung.



c) Verwenden Sie den Table-Filling-Algorithmus aus der Vorlesung, um den minimalen DFA zur Sprache des folgenden DFAs zu konstruieren. Geben Sie an, in welcher Reihenfolge Sie die Tabelle ausfüllen.



Aufgabe 11: Automatenkonstruktion

Es sei $k \in \mathbb{N}$ eine feste, aber beliebige gerade Zahl und $\Sigma = \{a, b\}$. Konstruieren Sie einen NFA A_k über Σ, der die folgende Sprache akzeptiert:

$$L = \left(\left(a^k\right)^* \cup \left((ab)^{k/2}\right)^*\right)^k .$$

Hinweis: A_K darf interne Transitionen enthalten.

Aufgabe 12: Kongruenzen

a) Zu einem NFA $A = (Q, q_0, \to, Q_F)$ definieren wir eine Relation $\equiv \subseteq \Sigma^* \times \Sigma^*$ durch $w \equiv v$ gdw. für alle Zustände $q \in Q$ gilt: $q_0 \xrightarrow{w} q$ gdw. $q_0 \xrightarrow{v} q$.

   Beweisen Sie: $\equiv$ ist eine Äquivalenzrelation und Rechtskongruenz.

b) Betrachten Sie die reguläre Sprache

$$\mathcal{L} = \left\{w \in \{a, b, c\}^* \;\middle|\; w \text{ enthält entweder } aa \text{ oder } c\right\} .$$

   Bestimmen Sie die Äquivalenzklassen der Nerode-Rechtskongruenz $\equiv_{\mathcal{L}}$ und geben Sie den Äquivalenzklassenautomaten an.

Aufgabe 13: Quiz zu regulären Sprachen

Geben Sie zu jeder der folgenden Aussagen einen kurzen Beweis oder ein Gegenbeispiel an.

a) Angenommen $\mathcal{L}_1$ und $\mathcal{L}_2$ sind beide nicht-reguläre Sprachen über dem selben Alphabet. Dann ist $\mathcal{L}_1 \cup \mathcal{L}_2$ nie regulär.

b) Es sei $f\colon \Sigma^* \to \Sigma^*$ eine Funktion auf Wörtern und $\mathcal{L}$ eine reguläre Sprache.
   Die Sprache $f(\mathcal{L}) = \{f(w) \mid w \in \mathcal{L}\}$ ist immer regulär.

c) Es sei $\mathcal{L}$ eine reguläre Sprache und k der Index der Nerode-Rechtskongruenz $\equiv_{\mathcal{L}}$. Es kann keinen NFA A mit $\mathcal{L}(A) = \mathcal{L}$ mit echt weniger als $\lfloor \log_2 k \rfloor$ Zuständen geben.

E) Kontextfreie Sprachen

Aufgabe 14: CYK

Gegeben ist die kontextfreie Grammatik $G = (\{S, A, B, C\}, \{a, b\}, P, S)$, wobei P durch die folgenden Regeln gegeben ist.

$$S \to AB \mid AC,$$
$$A \to AA \mid BB \mid a,$$
$$B \to AA \mid b,$$
$$C \to AA \mid CC.$$

Entscheiden Sie mit Hilfe des Cocke–Younger–Kasami-Algorithmus aus der Vorlesung, ob das Wort $w = aabba$ von G erzeugt wird.

Aufgabe 15: CYK 2

a) Modifizieren Sie den Cocke–Younger–Kasami-Algorithmus, so dass er nicht bloß entscheidet, ob $w \in \mathcal{L}(G)$ gilt, sondern in diesem Fall auch eine Linksableitung $S \Rightarrow_{SL} \ldots \Rightarrow_{SL} w$ ausgibt.

   Beschreiben Sie das Verfahren und begründen Sie kurz seine Korrektheit.

b) Illustrieren Sie das Verfahren an dem Wort $w = aaa$ und folgender Grammatik

$$S \to AA \mid SA$$
$$A \to AA \mid a.$$

Aufgabe 16: Quiz zu kontextfreien Sprachen

Geben Sie zu jeder der folgenden Aussagen einen kurzen Beweis oder ein Gegenbeispiel an.

a) Es sei $\mathcal{L}$ eine kontextfreie Sprache. Dann ist der Index der Nerode-Rechtskongruenz $\equiv_{\mathcal{L}}$ immer unendlich.

b) Es sei $\mathcal{L} \subseteq \Sigma^*$ eine kontextfreie Sprache. Wir definieren $\mathcal{L}' = \{a^n \in \{a\}^* \mid \exists w \in \mathcal{L}\colon |w| = n\}$. Die Sprache $\mathcal{L}'$ ist ebenfalls kontextfrei.

c) Wenn $\mathcal{L}_1, \mathcal{L}_2 \subseteq \Sigma^*$ kontextfrei, aber nicht regulär sind, dann ist auch $\mathcal{L}_1.\mathcal{L}_2$ kontextfrei, aber nicht regulär.