Abschlussklausur Theoretische Informatik I 2017/2018

08. Februar 2018

1. **Bitte ausfüllen:**

   Vorname:

   Nachname:

   Matrikelnummer:

   Unterschrift:

2. Achten Sie darauf, dass Ihre Klausur vollständig ist und getackert bleibt (15 Blätter)!

3. Benutzen Sie **nur das an dieses Blatt angeheftete Papier**. Bei Bedarf können wir weitere Leerblätter austeilen. Wenn der Platz auf der Vorderseite des jeweiligen Aufgabenblatts nicht ausreicht, machen Sie kenntlich, wo Sie die Bearbeitung der Aufgabe fortsetzen.

4. Als Hilfsmittel sind **ausschließlich** Sprachwörterbücher und ein zweiseitig **handgeschriebenes DIN A4-Blatt** erlaubt. Elektronische Geräte müssen während der Klausur ausgeschaltet bleiben. Täuschungsversuche werden mit 0 Punkten gewertet und dem Prüfungsamt gemeldet.

5. Schreiben Sie leserlich und bearbeiten Sie Ihre Klausur **nicht mit Bleistift** und auch **nicht in roter oder grüner Farbe!**

6. Wir werden das Deckblatt während der Klausur auf korrekte Daten überprüfen. Legen Sie dazu Ihren **Studentenausweis** und einen **amtlichen Lichtbildausweis** bereit.

Die Bearbeitungszeit beträgt 120 Minuten.

Mit 40 Punkten ist die Klausur sicher bestanden.

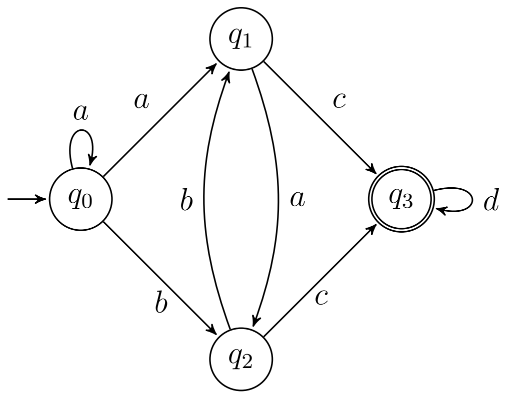
Punkteverteilung: (wird von den Korrektoren ausgefüllt)

Aufgabe	1	2	3	4	5	6	7	8	9	10	Σ
Max.	10	10	10	10	10	10	10	10	10	10	**100**
Punkte											

1. Ardens Lemma

$\boxed{\text{10 Punkte}}$

Gegeben sei der folgende NFA A über dem Alphabet $\Sigma = \{a, b, c, d\}$:



Geben Sie das zu A gehörige Gleichungssystem an und lösen Sie dieses unter Verwendung von Ardens Lemma.

2. CYK-Algorithmus

10 Punkte

Entscheiden Sie mit Hilfe des Cocke–Younger–Kasami-Algorithmus, ob das Wort *baacaa* von folgender Grammatik erzeugt wird:

$$
\begin{aligned}
S &\to AA \mid BC, \\
A &\to AA \mid a, \\
B &\to BA \mid b, \\
C &\to CS \mid c.
\end{aligned}
$$

3. Reaching Definitions

$\boxed{2 + 3 + 5 = 10 \text{ Punkte}}$

Führen Sie für das folgende Programm P eine Reaching-Definitions-Analyse durch. Beachten Sie, dass es sich hierbei um eine Vorwärts-May-Analyse handelt.

$[x := 4]^1$
if $[y > 3]^2$ **then**
  while $[y < 5]^3$ **do**
    $[x := x - 2]^4$
  end while
else
  $[x := 7]^5$
end if
$[y := x + 3]^6$

a) Zeichnen Sie den Kontrollflussgraphen G von P.

b) Betrachten Sie den Verband

$$(D, \leq) = (\mathcal{P}(\{x, y\} \times (\{1, \ldots, 6\} \mathbin{\dot\cup} \{?\})), \subseteq) .$$

Intuitiv bedeutet (x, i), dass der letzte Schreibzugriff auf x in Block i erfolgte (bzw. x noch nicht initialisiert ist, falls $i =?$). Geben Sie für die Blöcke 1 - 6 geeignete monotone Transferfunktionen über diesem Verband an.

c) Betrachten Sie das Datenflusssystem

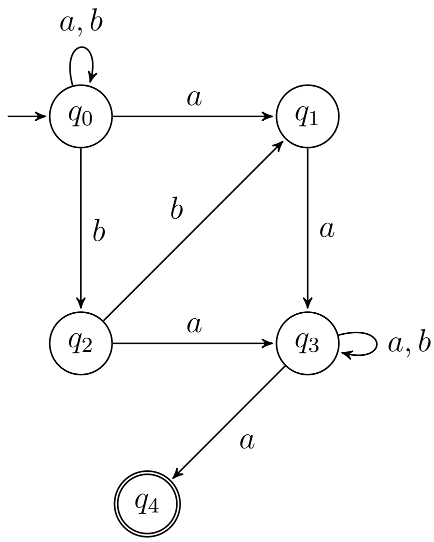
$$(G, (D, \leq), \{(x, ?), (y, ?)\}, TF) ,$$

wobei TF die Transferfunktionen aus Teil b) sind. Geben Sie das induzierte Gleichungssystem an und bestimmen Sie seine kleinste Lösung mit dem Satz von Kleene.

4. Determinisierung

10 Punkte

Determinisieren Sie folgenden NFA über $\Sigma = \{a, b\}$ unter Verwendung der Potenzmengen-Konstruktion:



5. Pumping-Lemma

10 Punkte

Zeigen Sie, dass die Sprache

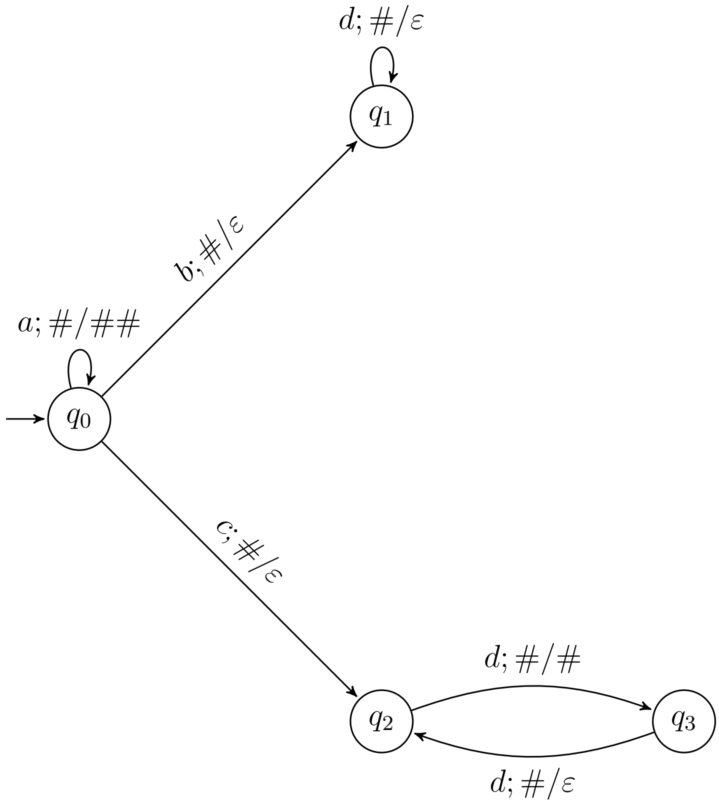
$$L = \left\{ a^{\sum_{i=1}^{n} i} \;\middle|\; n \in \mathbb{N} \right\}$$

nicht regulär ist.

6. Tripelkonstruktion

10 Punkte

Gegeben sei der folgende PDA P über dem Alphabet $\Sigma = \{a, b, c, d\}$ mit initialem Stack-Symbol $\#$. Ferner akzeptiert P mit leeren Stack.



Benutzen Sie das Verfahren aus der Vorlesung um eine kontextfreie Grammatik G mit $L(G) = L(P)$ zu konstruieren.

7. Pushdown-Automaten

$\boxed{4 + 6 = 10 \text{ Punkte}}$

Es sei P ein PDA mit Zuständen Q, finalen Zuständen $Q_F \subseteq Q$ und einem Stack-Symbol $\# \in \Gamma$. Wir kombinieren die Akzeptanzbedingungen von den beiden PDA-Modellen aus der Vorlesung: P akzeptiert, falls der aktuelle Stack-Inhalt leer ist **und** der aktuelle Zustand q final ist ($q \in Q_F$). Der PDA startet mit initialem Stack-Symbol $\#$. Wir nennen diese Akzeptanzbedingung **kombinierte Akzeptanz**.

a) Zeigen Sie, dass es zu jedem PDA mit Endzustands-Akzeptanz einen PDA mit kombinierter Akzeptanz gibt, der die gleiche Sprache akzeptiert.

b) Zeigen Sie, dass es zu jedem PDA mit kombinierter Akzeptanz einen PDA mit Endzustands-Akzeptanz gibt, der die gleiche Sprache akzeptiert.

8. Fragen zu Sprachen

$\boxed{2 + 3 + 3 + 2 = 10 \text{ Punkte}}$

Beantworten Sie die folgenden Fragen. Begründen Sie ihre Antwort mit einem kurzen Beweis oder einem Gegenbeispiel.

a) Ist die unendliche Vereinigung von regulären Sprachen immer nicht-regulär?

b) Es seien L_1 eine kontextfreie Sprache und L_2 eine nicht-kontextfreie Sprache. Ist dann $L_1.L_2$ auch immer nicht-kontextfrei?

c) Es sei L eine reguläre Sprache über $\{a, b, c\}$. Ist dann die Sprache

$$L' = \{a^n \mid \exists w \in L \text{ mit } |w| = n\}$$

auch regulär?

d) Ist jede endliche Sprache kontextfrei?

9. Spiegel-Sprache

10 Punkte

Es sei Σ ein endliches Alphabet. Für ein Wort $w = a_1.a_2.\ldots.a_n \in \Sigma^*$, wobei $a_1, \ldots, a_n$ Symbole aus Σ sind, ist das invertierte Wort w^R definiert durch $w^R = a_n.a_{n-1}.\ldots.a_1$.

Zeigen Sie dass die Sprache

$$\mathsf{spiegel}(L) = \{w_1^R.w_2 \mid w_1, w_2 \in \Sigma^* \text{ und } w_1.w_2 = w \in L\}$$

kontextfrei ist, für jede reguläre Sprache L.

10. Ersetzen-Sprache

$\boxed{1 + 6 + 3 = 10 \text{ Punkte}}$

Sei Σ ein endliches Alphabet. Für ein $w \in \Sigma^*$ und eine Konstante $k \in \mathbb{N}$ sei

$$\begin{aligned}\mathsf{Ersetzen}(w,k) = \{w_1.b_1.w_2.b_2.\dots w_n.b_n.w_{n+1} \mid & w = w_1.a_1.w_2.a_2.\dots w_n.a_n.w_{n+1} \in L,\\ & w_1,\dots,w_n \in \Sigma^*,\\ & a_1,\dots,a_n,b_1,\dots,b_n \in \Sigma \text{ und } n \leq k\}\end{aligned}$$

die Funktion die die Menge der Wörter zurückgibt, welche durch Ersetzung von höchstens k Positionen in w durch jeweils ein beliebiges Symbol aus Σ erzeugt werden können. Beispielsweise ist für $\Sigma = \{a, b\}$:

$$\begin{aligned}\mathsf{Ersetzen}(aaa,2) = \{&aaa && \text{// 0 Ersetzungen}\\ &baa, aba, aab && \text{// 1 Ersetzung}\\ &bba, bab, abb\} && \text{// 2 Ersetzungen}\\ \mathsf{Ersetzen}(\varepsilon,k) = \{&\varepsilon\}\end{aligned}$$

Das Ziel dieser Aufgabe ist zu beweisen dass für jede reguläre Sprache L über Σ

$$\mathsf{Ersetzen}(L,k) = \bigcup_{w \in L} \mathsf{Ersetzen}(w,k)$$

auch regulär ist.

a) Geben Sie einen NFA für $\mathsf{Ersetzen}(L,0)$ an.

b) Konstruieren Sie (schematisch) einen NFA für $\mathsf{Ersetzen}(L,1)$.
Hinweis: Erweitern Sie den NFA aus Aufgabenteil a) indem Sie sich an folgender rekursiven Definition von $\mathsf{Ersetzen}(w,k)$ für $w \neq \varepsilon$ orientieren.

$$\mathsf{Ersetzen}(w,k) = \bigcup_{\substack{\text{Zerlegung von w}\\ w=w_1.a.w_2}} \bigcup_{b \in \Sigma} w_1.b.\mathsf{Ersetzen}(w_2,k-1)$$

Hierbei sind $w_1, w_2 \in \Sigma^*$, $a \in \Sigma$ und $w.L' = \{w.w' \mid w' \in L'\}$ für eine Sprache $L' \subseteq \Sigma^*$.

c) Konstruieren Sie (schematisch) einen NFA für $\mathsf{Ersetzen}(L,k)$ für eine gegebene Konstante $k \in \mathbb{N}$.