Abschlussklausur
Theoretische Informatik 1
18. Februar 2021

Prof. Dr. Roland Meyer
Thomas Haas
TU Braunschweig
Wintersemester 2020/21

1. Lesen Sie bitte die **Selbstverständniserklärung** durch, bevor Sie die Klausur zu bearbeiten beginnen.

2. Schreiben Sie **leserlich** und **nummerieren** Sie die Seiten Ihrer Abgabe.

3. **Laden** Sie bis zum Ende der Klausur Ihre **Abgabe** in den dafür vorgesehenen **Abgabeordner hoch**. Sie können Ihre **Abgabe** dazu **abfotografieren**, **einscannen** oder **direkt als PDF** per Tablet oder ähnlichem erstellen. Abgaben als **.pdf, .jpg** oder sonstigen **Standardformaten** sind möglich.

4. Sie dürfen das **Skript** und ihre **eigenen Aufzeichnungen** verwenden. Das Heranziehen **fremder Hilfe** (z.B. andere Studenten oder Internetforen) ist **untersagt**.

5. Wenn Sie im Laufe der Klausur **Fragen** haben, steht Ihnen folgender BBB-Raum zur Verfügung:
`https://webconf.tu-bs.de/tho-6n6-t7e`
Stellen Sie Ihre Frage über den öffentlichen Chat. **Nach Beantwortung** Ihrer Frage, **verlassen** Sie bitte den BBB-Raum wieder.

6. Im Falle von Auftreten **technischer Probleme**, machen Sie bitte **Beweisfotos** und melden Sie anschließend die Probleme telefonisch unter +49 531-391-9522.

7. Wir werden den Termin für die Klausureinsicht auf unserer Website bekanntgeben:
`tcs.cs.tu-bs.de/teaching/TheoInf1_WS_20202021.html`.

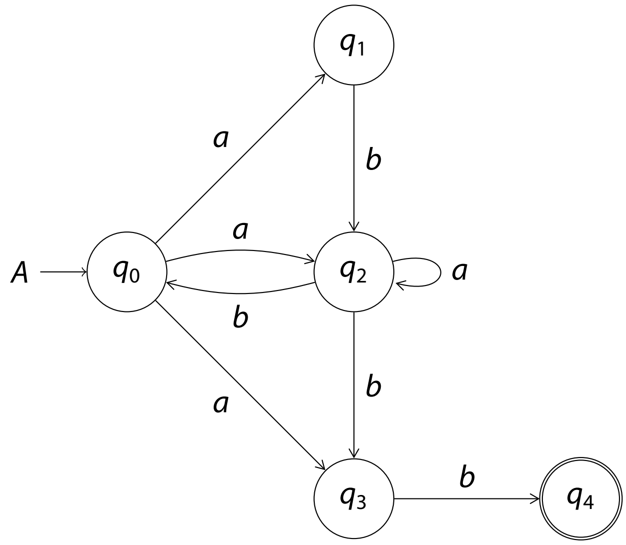
8. Die **Bearbeitungszeit** beträgt **180 Minuten**. Laden Sie die Klausur **rechtzeitig** hoch!

9. Mit **40 Punkten** ist die Klausur **sicher bestanden.**

1. Determinisierung und Komplementierung

10 Punkte

Berechnen Sie einen DFA zur Komplementsprache $\overline{\mathcal{L}(A)}$ der Sprache $\mathcal{L}(A)$ des folgenden NFA A über $\Sigma = \{a, b\}$. Verwenden Sie hierzu die Rabin-Scott-Potenzmengenkonstruktion aus der Vorlesung. Konstruieren Sie nur die vom Startzustand aus erreichbaren Zustände.



2. CYK

10 Punkte

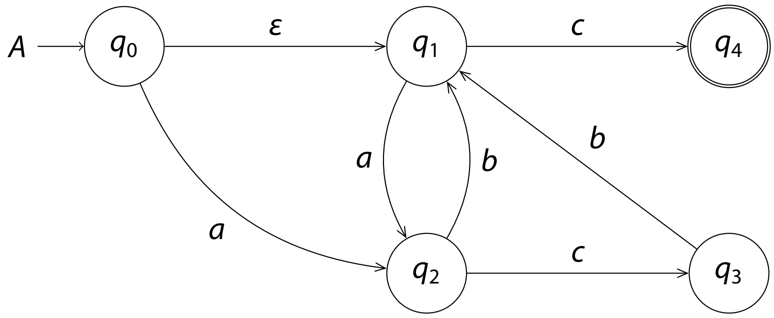
Nutzen Sie den Cocke-Younger-Kasami-Algorithmus aus der Vorlesung, um zu bestimmen, ob das Wort $w = abaabb$ von der kontextfreien Grammatik $G = (\{X, A, B, C\}, \{a, b, c\}, P, S)$ mit den folgenden Produktionsregeln erzeugt wird. Füllen Sie die Tabelle vollständig aus.

$$
\begin{aligned}
S &\to BB \mid CA \mid b \,, \\
A &\to AA \mid a \,, \\
B &\to AC \mid BC \,, \\
C &\to CA \mid b \,.
\end{aligned}
$$

3. NFA zu REG mit Ardens Lemma

10 Punkte

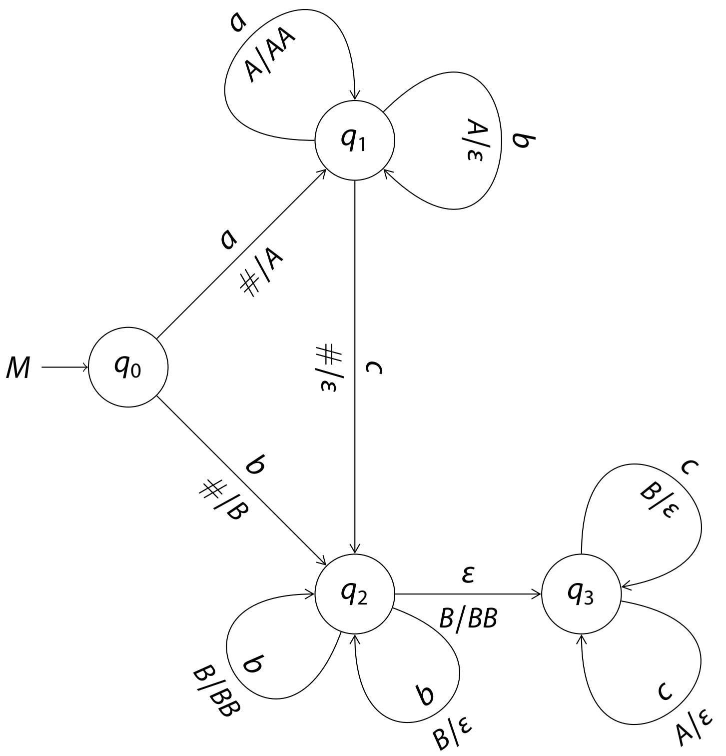
Geben Sie einen regulären Ausdruck an, der die Sprache $\mathcal{L}(A)$ des folgenden NFA A über $\Sigma = \{a, b, c\}$ beschreibt. Stellen Sie hierzu ein Gleichungssystem auf und lösen Sie es unter Verwendung von Ardens Lemma.



4. Tripelkonstruktion

10 Punkte

Betrachten Sie den Pushdown-Automaten $M = (\{q_0, q_1, q_2, q_3\}, \{a, b, c\}, \{\#, A, B\}, q_0, \#, \delta)$, der mit leerem Stack akzeptiert und dessen Transitionsrelation δ durch das folgende Diagramm definiert ist. Verwenden Sie die Tripelkonstruktion aus der Vorlesung, um eine kontextfreie Grammatik G mit $\mathcal{L}(M) = \mathcal{L}(G)$ zu bestimmen.



5. Teilerordnung auf den ganzen Zahlen $\boxed{2+3+5=10 \text{ Punkte}}$

Sei $M = \mathbb{Z} \setminus \{0\}$ die Menge aller ganzen Zahlen ohne 0. Wir definieren die Teilerrelation $|$ auf M mit $p|q$ gdw. $\frac{q}{p} \in M$.

a) Zeigen Sie, dass $(M, |\,)$ **keine** partielle Ordnung ist.

b) Wir definieren die Relation $||$ auf M mit $p||q$ gdw. $p|q$ und $q|p$. Zeigen Sie, dass $||$ eine **Äquivalenzrelation** auf M ist.

c) Wir definieren $\overline{M} = \{[p]_{||} \mid p \in M\}$ als die **Menge aller Äquivalenzklassen** von M unter $||$.

   Zeigen Sie nun: $(\overline{M}, |\,)$ ist eine partielle Ordnung. Dabei definieren wir $[p]_{||} \mid [q]_{||}$ gdw. $p|q$.

   Zeigen Sie zuerst, dass $|$ auf $\overline{M}$ **wohldefiniert** ist, also nicht von den benutzten Repräsentanten abhängt.

Erinnerung: Eine Äquivalenzrelation ist eine Relation, die reflexiv, symmetrisch und transitiv ist.

6. Automatenkonstruktion

10 Punkte

Betrachten Sie die Sprache

$$\mathcal{L} = \left\{ w \in \{a,b\}^* \;\middle|\; |w|_b \text{ ungerade } \textbf{ODER } (|w|_{bb} \leq |w|_{ab} \textbf{ UND } w \in a.\Sigma^*) \right\} \subseteq \{a,b\}^* .$$

Konstruieren Sie einen PDA M, der die Sprache $\mathcal{L}$ akzeptiert.

Geben Sie insbesondere die Akzeptanzbedingung Ihres Automats an und erklären Sie Ihre Konstruktion.

Achten Sie auf die Klammerung in der Definition der Sprache!

Definition: $y \in \Sigma^*$ ist ein **Infix** eines Wortes $w \in \Sigma^*$, wenn es $x, z \in \Sigma^*$ gibt mit $w = x.y.z$.

$|w|_y$ ist die **Anzahl an Infixen** $y \in \Sigma^*$ in w. Beispielweise gilt für das Wort $w = aabbba$: $|w|_{ab} = 1$, $|w|_{bb} = 2$ und $|w|_b = 3$.

7. Nichtregularität

10 Punkte

Betrachten Sie wieder die Sprache

$$\mathcal{L} = \left\{ w \in \{a,b\}^* \;\middle|\; |w|_b \text{ ungerade ODER } (|w|_{bb} \leq |w|_{ab} \text{ UND } w \in a.\Sigma^*) \right\} \subseteq \{a,b\}^* .$$

aus Aufgabe 6.

Zeigen Sie, dass $\mathcal{L}$ nicht regulär ist. Sie dürfen dazu alle Ihnen bekannten Methoden verwenden.

8. Quiz

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

Bestimmen Sie zu jeder der folgenden Aussagen, ob sie wahr oder falsch ist. Geben Sie jeweils einen kurzen Beweis oder ein Gegenbeispiel an.

a) Das Komplement einer nicht-kontextfreien Sprache $\mathcal{L}$ ist immer nicht-kontextfrei.

b) Eine Sprache $\mathcal{L} \subseteq \Sigma^*$ heißt co-endlich, wenn sie **alle bis auf endlich viele** Wörter enthält. Jede co-endliche Sprache ist regulär.

c) Sei $\mathcal{L} \subseteq \Sigma^*$ eine Sprache und sei $h : \Sigma^* \to \Gamma^*$ ein Homomorphismus, der kein Wort auf ε abbildet. Wenn $h(\mathcal{L})$ regulär ist, dann ist $\mathcal{L}$ auch regulär.

d) Die Sprache $\mathcal{L} = \{w \in \{0,1\}^* \mid |w|_{01} = |w|_{10}\}$ ist regulär.

   Bemerkung: Die Notation $|w|_x$ ist in Aufgabe 6 erklärt.

9. Regular-Stack Pushdowns

10 Punkte

Ein **Regular-Stack Pushdown** (Reg-PDA) $M = (Q, \Sigma, \Gamma, q_0, \delta, Q_F)$ ist wie ein Pushdown-Automat mit Akzeptanz bei Finalzustand definiert, aber mit folgender Besonderheit: Transitionen sind von der Form

$$q_1 \xrightarrow[\gamma_1/\mathcal{L}]{\sigma} q_2$$

wobei $\mathcal{L} \subseteq \Gamma^*$ eine **nicht-leere reguläre Sprache** ist. Bei einer solchen Transition wird das Stacksymbol γ_1 verbraucht und durch ein **beliebiges** Wort $\gamma_2 \in \mathcal{L}$ ersetzt.

Zeigen Sie, dass Reg-PDAs genau die kontextfreien Sprachen akzeptieren.

Bemerkung: Damit zeigen Sie, dass Reg-PDAs nicht stärker sind als übliche PDAs.

10. Bounded-Difference Sprachen

10 Punkte

Sei $w \in \Sigma^*$ ein Wort. Wir sagen w hat **k-bounded-differences**, wenn gilt: Für alle Präfixe y von w, und alle Paare von Buchstaben $a, b \in \Sigma$ haben wir $\big|\, |y|_a - |y|_b \,\big| \leqslant k$.

Hierbei ist y ein **Präfix** von w, wenn es ein $z \in \Sigma^*$ gibt mit $w = y.z$.

Sei $\Sigma = \{a, b, c\}$. Zeigen Sie, dass für jedes $k \in \mathbb{N}$ die Sprache

$$\mathcal{L}_k = \left\{ w \in \Sigma^* \;\middle|\; |w|_a = |w|_b = |w|_c \text{ und w hat k-bounded-differences.} \right\}$$

regulär ist.