2. Klausur zur Vorlesung
Theoretische Grundlagen der Informatik
Wintersemester 2016/2017

Hier Aufkleber mit Name und Matrikelnummer anbringen

Vorname: ____________________________

Nachname: ____________________________

Matrikelnummer: ____________________________

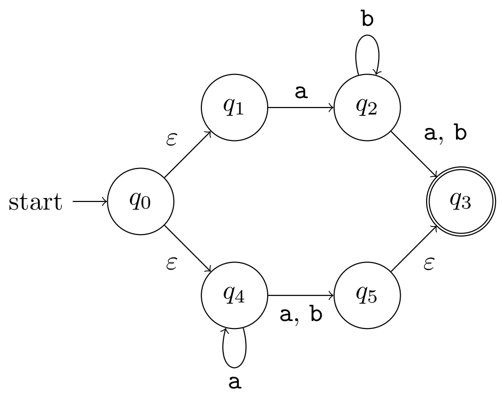
Beachten Sie:

- Bringen Sie den Aufkleber mit Ihrem Namen und Matrikelnummer auf diesem Deckblatt an und beschriften Sie jedes Aufgabenblatt mit Ihrem Namen und Matrikelnummer.
- Schreiben Sie die Lösungen auf die Aufgabenblätter und Rückseiten. Zusätzliches Papier erhalten Sie bei Bedarf von der Aufsicht.
- Es sind keine Hilfsmittel zugelassen.

Aufgabe	Mögliche Punkte						Erreichte Punkte					
	a	b	c	d	e	Σ	a	b	c	d	e	Σ
1	2	3	3	–	–	8				–	–	
2	4	–	–	–	–	4		–	–	–	–	
3	7	2	3	2	–	14					–	
4	6	–	–	–	–	6		–	–	–	–	
5	2	3	4	3	3	15						
6	1	2	3	–	–	6				–	–	
7	7×1					7						
Σ						60						

Problem 1: Automatenkonstruktion $2 + 3 + 3 = 7$

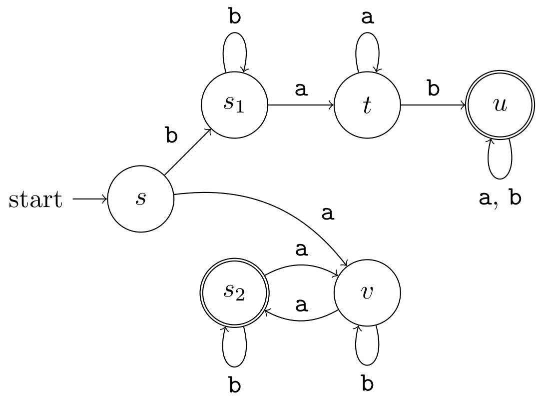
(a) Sei folgender nichtdeterministischer endlicher Automat $\mathcal{A}$ gegeben:



Dabei führen alle nicht explizit angegebenen Übergänge in einen impliziten Fehlerzustand. Geben Sie einen regulären Ausdruck an, der die Sprache $L(\mathcal{A})$ erzeugt und dabei höchstens zweimal das Vereinigungssymbol $\cup$ enthält.

(b) Geben Sie für jeden Zustand dessen ε-Abschluss an und geben Sie für $\mathcal{A}$ einen äquivalenten endlichen nichtdeterministischen Automaten $\mathcal{A}'$ an, der keinen ε-Übergang enthält. Der Automat $\mathcal{A}'$ soll dieselbe Zustandsmenge wie $\mathcal{A}$ haben. Markieren Sie alle überflüssigen Zustände.

(c) Zeigen Sie mit einem der aus Vorlesung oder Übung bekannten Verfahren, dass der folgende deterministische endliche Automat minimal ist.



Problem 2: Kontextfreie Grammatiken 4

Gegeben ist folgende kontextfreie Grammatik $G = (\{a, b, c, d\}, \{A, B, C, D, E\}, A, R)$. Die Regelmenge R enthält folgende Regeln:

$$
\begin{aligned}
A &\to AB \mid EC \mid a \\
B &\to CA \\
C &\to CC \mid DC \mid c \\
D &\to d \\
E &\to CA \mid DA
\end{aligned}
$$

Überprüfen Sie mittels des aus der Vorlesung bekannten CYK-Algorithmus, ob das Wort *adcaca* in $L(G)$ enthalten ist.

a	d	c	a	c	a

Problem 3: $\mathcal{NP}$-Vollständigkeit $7 + 2 + 3 + 2 = 14$

(a) Sei M ein Spielbrett mit jeweils $m \in \mathbb{N}$ Spalten und Zeilen, die das Spielbrett in m^2 Zellen aufteilen. In jeder Zelle liegt entweder ein roter, ein blauer, oder kein Stein. Eine Belegung der Zellen durch Steine heißt Konfiguration. Eine Konfiguration ist *gewinnend*, wenn jede Zeile mindestens einen Stein enthält und jede Spalte nur Steine gleicher Farbe enthält. Betrachten Sie folgendes Problem.

Problem Solitaire

Gegeben: Eine Konfiguration M.

Frage: Kann M durch Entfernen von Steinen zu einer gewinnenden Konfiguration M' transformiert werden?

Zeigen Sie, dass Solitaire $\mathcal{NP}$-vollständig ist. Verwenden Sie dazu, dass das 3-SAT-Problem $\mathcal{NP}$-vollständig ist.

Doktor Meta forscht in seinem Labor am Geheimproblem X. Er hat bereits eine polynomielle Transformation von SAT auf X gefunden, die SAT-Instanzen der Größe n auf X-Instanzen der Größe n^3 abbildet. In einem Moment der Genialität gelingt es Doktor Meta nun, eine Laufzeitschranke für SAT von $\Omega(c^n)$ für eine Konstante $c > 1$ zu beweisen.

(b) Was folgt aus der Laufzeitschranke für SAT über die Frage $\mathsf{P} \stackrel{?}{=} \mathcal{NP}$? Begründen Sie!

(c) Welche Laufzeitschranke ergibt sich für X? Begründen Sie!

(d) Können Sie mit der Laufzeitschranke für X die Frage $\mathsf{P} \stackrel{?}{=} \mathcal{NP}$ beantworten? Begründen Sie!

Problem 4: Berechenbarkeit 6

Sei T_w die Turing-Maschine mit Gödelnummer w. Zeigen Sie, dass die Sprache

$$L = \big\{(u, v) \in \{0,1\}^\star \times \{0,1\}^\star \mid w \in L(T_u) \text{ genau dann, wenn } w^R \in L(T_v)\big\}$$

nicht entscheidbar ist. Verwenden Sie für Ihren Beweis, dass das Halteproblem nicht eintscheidbar ist.

Hinweis: Das Wort w^R bezeichnet das Spiegelwort von w.

Problem 5: Approximationsalgorithmen $2 + 3 + 4 + 3 + 3 = 15$

Aus Vorlesung und Übung kennen Sie das $\mathcal{NP}$-vollständige Problem VERTEX COVER:

Problem VERTEX COVER

Gegeben: ungerichteter Graph $G = (V, E)$ und $k \in \mathbb{N}$

Gesucht: Menge $V' \subseteq V$ mit $|V'| \leq k$, sodass für alle Kanten $(u, v) \in E$ gilt: $u \in V' \lor v \in V'$.

(a) Ist das Problem in der oben gegebenen Formulierung ein Optimierungsproblem, ein Optimalwertproblem, oder ein Entscheidungsproblem? Geben Sie auch die beiden anderen Formulierungen an. *Anmerkung für Lernende: Die Frage lässt sich nicht wie gefordert beantworten.*

Es soll eine approximative Lösung für das VERTEX COVER Problem berechnet werden. Dazu wird zunächst $V' = \emptyset$ gesetzt. Solange G eine Kante enthält wird dazu eine beliebige Kante $(u, v) \in E$ gewählt, dann beide Knoten u, v zu V' hinzugefügt und aus G entfernt. Um einen Knoten w aus G zu entfernen werden alle zu w inzidenten Kanten und w selbst entfernt.

(b) Zeigen Sie, dass dieser Algorithmus tatsächlich ein VERTEX COVER berechnet.

(c) Zeigen Sie, dass dieser Algorithmus ein Approximationsalgorithmus für das Vertex Cover Problem mit einer relativen Gütegarantie von 2 ist.

Hinweis: Die Menge der gewählten Kanten bildet ein Matching.

(d) Zeigen Sie, dass unter der Annahme dass $\mathcal{P} \neq \mathcal{NP}$ kein Polynomialzeitapproximationsalgorithmus mit absoluter Gütegarantie für das Vertex Cover Problem existiert.

(e) Der Algorithmus soll angepasst werden, um eine approximative Lösung für das INDEPENDENT SET Problem zu finden.

Problem INDEPENDENT SET

Gegeben: ungerichteter Graph $G = (V, E)$ und $k \in \mathbb{N}$

Gesucht: Menge $V' \subseteq V$ mit $|V'| \geq k$, wobei aus $v \in V'$ für alle Kanten $(u, v) \in E$ folgt, dass $u \notin V'$.

Dazu wird anstatt der Menge V' die Menge $V \setminus V'$ ausgegeben. Ist der so abgewandelte Algorithmus ein Approximationsalgorithmus für das INDEPENDENT SET Problem mit relativer Gütegarantie 2? Begründen Sie!

Problem 6: Kellerautomaten $1 + 2 + 3 = 6$

(a) Die kontextfreie Grammatik G_1 über dem Eingabealphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ sei definiert durch die Menge der Nichtterminalsymbole $\{S, L, R\}$ mit dem Startsymbol S und folgenden Ableitungsregeln:

$$\begin{aligned} S &\to LR \mid L \mid R \\ L &\to \mathtt{a}L\mathtt{b} \mid \mathtt{ab} \\ R &\to \mathtt{b}R\mathtt{a} \mid \mathtt{ba} \end{aligned}$$

Geben Sie die Sprache $L(G_1)$ in Mengenschreibweise an, die von G_1 erzeugt wird.

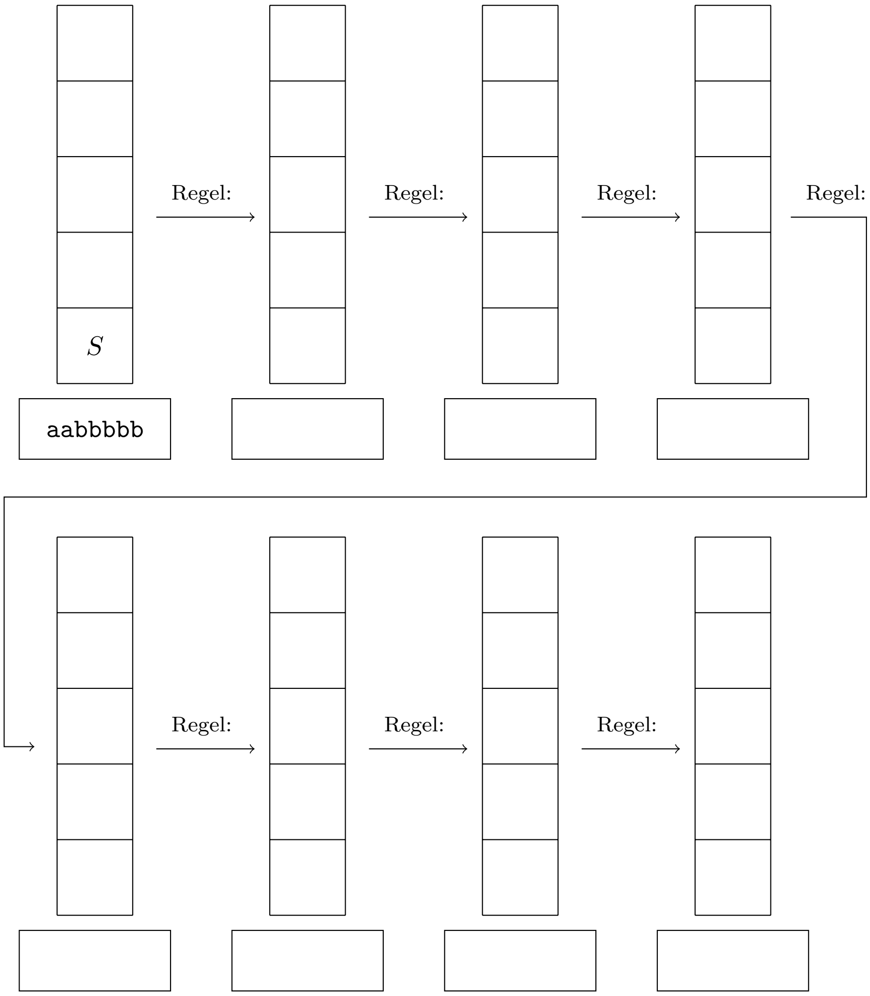
(b) Geben Sie eine Grammatik G_2 in Greibach-Normalform an, sodass $L(G_2) = L(G_1)$ gilt, wobei G_1 die Grammatik aus Teilaufgabe a ist.

Hinweis: Sie dürfen das Verfahren aus der Vorlesung nutzen, müssen aber nicht.

(c) Die kontextfreie Grammatik G_3 in Greibach-Normalform über dem Eingabealphabet $\Sigma = \{\texttt{a}, \texttt{b}\}$ sei definiert durch die Menge der Nichtterminalsymbole $\{S, B, X\}$, Startsymbol S und folgenden Ableitungsregeln:

$$S \to \texttt{a}SB \tag{1}$$
$$S \to \texttt{b}X \tag{2}$$
$$X \to \texttt{b}X \tag{3}$$
$$X \to \texttt{b} \tag{4}$$
$$B \to \texttt{b} \tag{5}$$

In Vorlesung und Übung wurde ein nichtdeterministischer Kellerautomat vorgestellt, der Sprachen erkennen kann, die von Grammatiken in Greibach-Normalform erzeugt werden. Geben Sie die Konfigurationen dieses Automaten an, die bei Abarbeitung der Eingabe `aabbbbb` entstehen. Vervollständigen Sie dazu folgendes Schema mit den zu lesenden Worten, den Stackinhalten und den Regeln, die beim Übergang verwendet werden. Ist das Eingabewort in der Sprache $L(G_3)$ enthalten?



Problem 7: Gemischtes 7

Sind die folgende Aussagen korrekt? Begründen Sie jeweils kurz Ihre Antwort.

(a) Sei ein nichtdeterministischer endlicher Automat $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ gegeben. Dann akzeptiert der Automat $(Q, \Sigma, \delta, s, Q \setminus F)$ die Sprache $L(\mathcal{A})^c$.

(b) Sei ein nichtdeterministischer endlicher Kellerautomat $\mathcal{A}$ gegeben. Dann existiert ein nichtdeterministischer endlichernichtdeterministischer endlicher Kellerautomat, der die Sprache $L(\mathcal{A})^c$ akzeptiert.

(c) Die Sprache aller binär kodierten Primzahlen kleiner als 2017^{2017} ist regulär.

(d) Mit dem CYK-Algorithmus kann die Zugehörigkeit eines Wortes zu einer Typ-1 Grammatik überprüft werden.

(e) Zu jeder entscheidbaren Sprache L existiert eine Typ-0 Grammatik, die L erzeugt.

(f) Es existiert eine Sprache in $\mathcal{NP}$, die in Polynomialzeit entschieden werden kann.

(g) Mit dem Pumpinglemma kann gezeigt werden, dass eine Sprache regulär ist.