Karlsruher Institut für Technologie

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

Klausur zur Vorlesung
Theoretische Grundlagen der Informatik
Wintersemester 2010/2011

Hier Aufkleber mit Name und Matrikelnr. anbringen

Vorname: ____________________

Nachname: ____________________

Matrikelnummer: ____________________

Beachten Sie:

- Bringen Sie den Aufkleber mit Ihrem Namen und Ihrer Matrikelnummer auf diesem Deckblatt an und beschriften Sie jedes Aufgabenblatt mit Ihrem Namen und Ihrer Matrikelnummer.
- Schreiben Sie die Lösungen auf die Aufgabenblätter und Rückseiten. Zusätzliches Papier erhalten Sie bei Bedarf von der Aufsicht.
- Zum Bestehen der Klausur sind **20** der möglichen **60** Punkte hinreichend.
- Es sind keine Hilfsmittel zugelassen.

Aufgabe	Mögliche Punkte				Erreichte Punkte			
	a	b	c	Σ	a	b	c	Σ
1	4	-	-	4		-	-	
2	4	-	-	4		-	-	
3	3	-	-	3		-	-	
4	4	-	-	4		-	-	
5	3	4	-	7			-	
6	6	-	-	6		-	-	
7	4	2	-	6			-	
8	1	2	3	6				
9	3	-	-	3		-	-	
10	2	5	-	7			-	
11	10x1			10				
Σ				60				

Aufgabe 1: (4 Punkte)

Beweisen Sie, dass die Sprache $L = \{w_1 w_2 \mid w_1 \in \{a,b\}^k,\ w_2 = c^j,\ k < j,\ k,j \in \mathbb{N}_{>0}\}$ über dem Alphabet $\Sigma = \{a,b,c\}$ nicht regulär ist.

Aufgabe 2: (4 Punkte)

Gegeben sei das Alphabet $\Sigma = \{a,b,c,d\}$.
Definitionen:

- Es sei L die Sprache aller Wörter über Σ in denen a nie neben b, b nie neben c und c nie neben d steht, wobei gilt:
- In einem Wort $w = w_1 \dots w_n$ *steht Zeichen x neben Zeichen y*, wenn es ein $i \in \{1,\dots,n-1\}$ gibt, so dass $x = w_i$ und $y = w_{i+1}$ oder $y = w_i$ und $x = w_{i+1}$.

Entwerfen Sie eine Grammatik, die die Sprache L erzeugt.

Aufgabe 3: (3 Punkte)

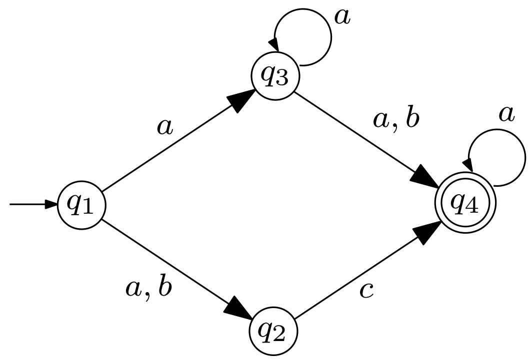
Gegeben sei die Grammatik $G = (\Sigma, V, S, R)$ mit Terminalalphabet $\Sigma = \{0, 1\}$, Variablen $V = \{S, A, B\}$ und Produktionen

$$\begin{aligned} R = \{ \quad S &\rightarrow \quad 0B|1A, \\ A &\rightarrow \quad 0|0S|1AA \\ B &\rightarrow \quad 1|1S|0BB \quad \} \; . \end{aligned}$$

Welche Sprache erzeugt G?

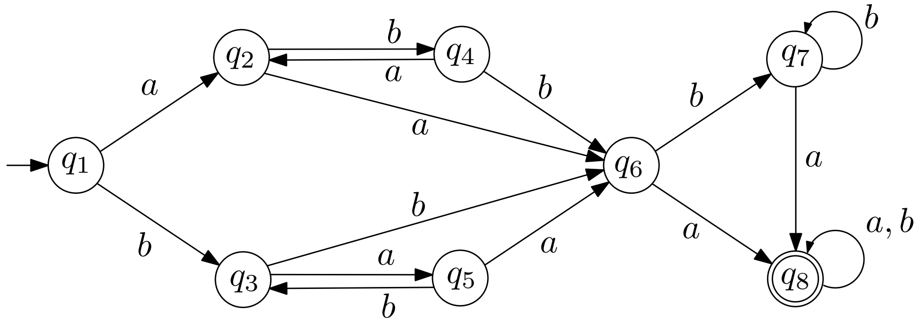
Aufgabe 4: (4 Punkte)

Konstruieren Sie zu folgendem NEA über dem Alphabet $\Sigma = \{a, b, c\}$ einen DEA, der dieselbe Sprache akzeptiert und zeichnen Sie das entsprechende Zustandsübergangsdiagramm.



Aufgabe 5: (3+4 Punkte)

Gegeben ist folgender endlicher Automat $\mathcal{A}$ über dem Alphabet $\Sigma = \{a, b\}$.



(a) Geben Sie einen regulären Ausdruck für die Sprache an, die $\mathcal{A}$ akzeptiert. Hinweis: Hier ist nicht verlangt, dass sie das Verfahren aus der Vorlesung benutzen.

(b) Konstruieren Sie den Minimalautomaten zu $\mathcal{A}$. Sie dürfen benutzen, dass der Minimalautomat aus 5 Zuständen besteht.

Aufgabe 6: (6 Punkte)

Problem SUBSET 3-INTERVALL (S-3INT)

Gegeben:	Endliche Menge M, Gewichtsfunktion $w : M \to \mathbb{N}_{>0}$, Zahl $k \in \mathbb{N}_{>0}$.
Frage:	Gibt es eine Teilmenge $M' \subseteq M$, so dass $\sum_{m \in M'} w(m) \in [k, k+3]$ gilt?

Zeigen Sie, dass das Problem SUBSET 3-INTERVALL NP-vollständig ist. Benutzen Sie dazu, dass das Problem SUBSET SUM NP-vollständig ist:

Problem SUBSET SUM (S-SUM)

Gegeben:	Endliche Menge M, Gewichtsfunktion $w : M \to \mathbb{N}_{>0}$, Zahl $k \in \mathbb{N}_{>0}$.
Frage:	Gibt es eine Teilmenge $M' \subseteq M$, so dass $\sum_{m \in M'} w(m) = k$ gilt?

Aufgabe 7: (4+2 Punkte)

(a) Sei L eine nicht entscheidbare Sprache über dem Alphabet $\Sigma = \{0, 1\}$ und sei $(01)^k \notin L$ für jedes $k \in \mathbb{N}_{>0}$. Zeigen Sie: Die Sprache

$$L' = \{w_1 \# w_2 \mid (w_1 \in L \wedge w_2 \notin L) \text{ oder } (w_1 \notin L \wedge w_2 \in L)\}$$

über dem Alphabet $\Sigma' = \{0, 1, \#\}$ ist nicht entscheidbar.

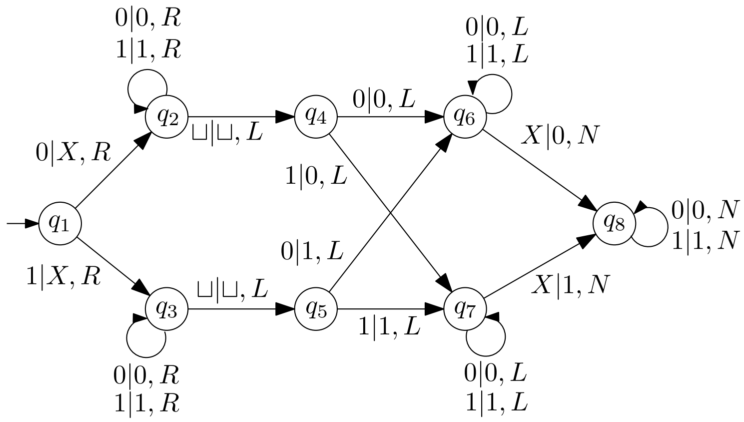
(b) Sei $\mathcal{A} = (Q = \{s, f\}, \Sigma = \{a, b\}, \Gamma = \{Y, Z\}, \delta, s, Z, \{f\})$ der Kellerautomat mit Zustandsmenge Q, Eingabealphabet Σ, STACK-Alphabet Γ, Anfangszustand s, Stack-Initialisierung Z, akzeptierendem Endzustand F und der folgenden Übergangsrelation δ:

$$\begin{array}{rcl} (s, a, Z) & \mapsto & (s, YZ) \\ (s, a, Y) & \mapsto & (s, YY) \\ (s, b, Y) & \mapsto & (s, \epsilon) \\ (s, \epsilon, Z) & \mapsto & (f, Z) \end{array}$$

Sei L die Sprache, die $\mathcal{A}$ durch akzeptierenden Endzustand erkennt. Formen Sie $\mathcal{A}$ in einen Kellerautomaten $\mathcal{A}'$ um, der L durch leeren Stack erkennt.

Aufgabe 8: (1+2+3 Punkte)

Gegeben sei die folgende Turingmaschine M mit Eingabealphabet $\{0,1\}$ und Bandalphabet $\{0,1,\sqcup,X\}$ wobei $\sqcup$ das Blanksymbol ist.



(a) Ist die Turingmaschine M deterministisch?

(b) Rechnen Sie das Verhalten der Turingmaschine M bei Eingabe des Wortes 0101 durch. Geben Sie dazu alle auftretenden Konfigurationen an.

(c) Was berechnet die Turingmaschine M für eine Eingabe der Länge mindestens 2?

Aufgabe 9: (3 Punkte)

Gegeben sei die Grammatik $G = (\Sigma, V, S, R)$ mit Terminalen $\Sigma = \{a, b, c, d, x, f\}$, Nichtterminalen $V = \{S, A, B, C\}$ und Produktionen

$$R = \{ \quad \begin{aligned} S &\to ABx|BCd|A \\ A &\to AAf|a|CC \\ B &\to b \\ C &\to c \quad \} . \end{aligned}$$

Berechnen Sie durch ein systematisches Verfahren eine Grammatik G' in Chomsky-Normalform, die die gleiche Sprache wie G erzeugt.

Aufgabe 10: (2+5 Punkte)

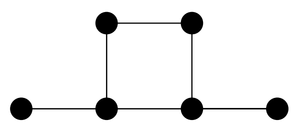
Gegeben sei ein ungerichteter Graph $G = (V, E)$.

- Zwei Kanten $e_1, e_2 \in E$ sind **adjazent**, wenn sie einen gleichen Endknoten haben.
- Ein **Matching** $E' \subseteq E$ von G ist eine Menge von Kanten die paarweise nicht adjazent sind.
- Ein **kardinalitätsmaximales Matching** E^* für G ist ein Matching mit maximal vielen Kanten, d.h. mit maximalem $|E^*|$.
- Ein **inklusionsmaximales Matching** E^{**} für G ist ein Matching, das nicht echte Teilmenge eines anderen Matchings ist, d.h. für das es kein Matching E'' für G gibt mit $E^{**} \subsetneq E''$.

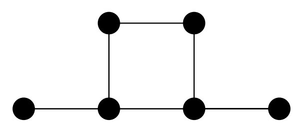
(a) Zeichnen Sie in

- Graphen (i) ein kardinalitätsmaximales Matching
- Graphen (ii) ein inklusionsmaximales Matching, das nicht kardinalitätsmaximal ist,

ein.



(i) kardinalitätsmaximales Matching



(ii) inklusionsmaximales, nicht kardinalitätsmaximales Matching

(b) Sei das folgende Maximierungsproblem MAXIMUM MATCHING gegeben:

Problem MAXIMUM MATCHING

Gegeben: Ein ungerichteter Graph $G = (V, E)$, der mindestens eine Kante enthält.

Gesucht: Ein Matching E' möglichst großer Kardinalität in G, d.h. wir haben ein Maximierungsproblem mit Zielfunktion $|E'|$ und der Bedingung, dass E' ein Matching in G ist.

Sei $\mathcal{A}$ ein Algorithmus, der für einen Eingabegraphen G ein beliebiges inklusionsmaximales Matching zurückliefert. Zeigen Sie: $\mathcal{A}$ ist 1-approximativ für Problem MAXIMUM MATCHING (d.h. $\mathcal{A}$ hat eine relative Gütegarantie von $1 + 1 = 2$).

Aufgabe 11: (10 Punkte)

Kreuzen Sie für folgende Aussagen an, ob diese wahr oder falsch sind.
Hinweis: Für jede richtige Antwort gibt es einen Punkt, für jede falsche Antwort wird ein Punkt abgezogen. Es wird keine negative Gesamtpunktzahl für diese Aufgabe geben.

Aussage	Wahr	Falsch
Die Sprache $L = \{(xyz)^i(abc)^i \mid i \in \mathbb{N}_{>0},\ i < 27\}$ ist regulär.	☐	☐
Für jeden Kellerautomaten $\mathcal{K}$ gilt: Die Sprache, die $\mathcal{K}$ durch akzeptierenden Endzustand erkennt, ist gleich der Sprache, die $\mathcal{K}$ durch leeren Stack erkennt.	☐	☐
Seien L_1, L_2 Sprachen vom Chomsky-Typ 2. Dann ist $L_1 \cap L_2$ vom Chomsky Typ 2.	☐	☐
Sei K ein Kellerautomat, der nie ein Zeichen vom Stack löscht oder auf den Stack schreibt und durch Endzustand akzeptiert. Dann gibt es einen NEA, der die gleiche Sprache akzeptiert wie K.	☐	☐
Wenn es für jedes Wort $w \in L$ einer Sprache L einen DEA A_w gibt, der w akzeptiert, dann ist L regulär.	☐	☐
Es gibt ein Entscheidungsproblem $\Pi \in NP$, für das es keine polynomiale Transformation $\Pi \propto SAT$ gibt.	☐	☐
Jede Sprache, die von einer Grammatik erzeugt wird, ist entscheidbar.	☐	☐
Sei Π ein Optimalwertproblem und $\mathcal{A}$ ein 4-Approximationsalgorithmus für Π. Dann liefert $\mathcal{A}$ mit Wahrscheinlichkeit 1/4 die richtige Lösung für Π.	☐	☐
Das Problem 2-SAT liegt in NP.	☐	☐
Sei L eine Sprache, die von einer nichtdeterministischen Turingmaschine $\mathcal{M}$ in Polynomialzeit akzeptiert wird. Dann folgt daraus, dass es eine deterministische Turing-Maschine gibt, die L entscheidet.	☐	☐