Tutorium 13&15 Klausurwiederholung WS2007/2008 Informatik III 13.02.2008

Aufgabe 1 3+3+4+4 Punkte

a)

Gegeben sei ein endlicher Akzeptor A mit

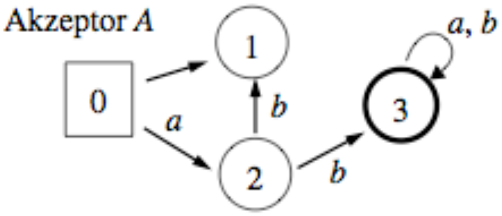
den Zuständen $Q = \{0, 1, 2, 3\}$,

dem Startzustand 0,

dem Eingabealphabet $T = \{a, b\}$

der Finalmenge $F = \{3\}$ sowie

den Produktionen $\Pi = \{0 ::= 1,\ 0a ::= 2,\ 2b ::= 1,\ 2b ::= 3,\ 3a ::= 3,\ 3b ::= 3\}$.



Bestimmen Sie den vollstaendigen DEA zum NEA A mithilfe einer syntaktischen Konstruktion.

A'=({{0},{1,2},{1},{1,3},{3},ø},{a,b},delta,{0},{{1,3},{3}}) mit delta:

	a	b
{0}	{1,2}	{1}
{1,2}	ø	{1,3}
{1}	ø	ø
Ø	ø	ø
{1,3}	{3}	{3}
{3}	{3}	{3}

Hinweis: Dieser DEA ist nicht minimal, was auch nicht gefordert ist!

b)

Gegeben seien die beiden folgenden Sprachen über dem Alphabet $\Sigma = \{a, b\}$:

$L_1 = \{$alle Wörter, die aa oder bb enthalten$\}$

$L_2 = \{$alle Wörter, in denen höchstens einmal aa und nie bb vorkommt$\}$

Geben Sie reguläre Ausdrücke für L_1 und L_2 an.

L1=(a + b)* (aa + bb) (a + b)*
L2=(epsilon + b) (ab)* (a + epsilon) (ab)* (epsilon + a)

c)

Beweisen Sie, dass die Sprache $L = \{a^n b a^m b a^{n+m} \mid n, m \geq 1\}$ nicht regulär ist.

Annahme: L regulaer. Mit dem Pumping Lemma folgt dann: ∃n ∈ N, sodass ∀w ∈ L mit |w| > n gilt: ∃ Zerlegung w = uvx mit v = ε, |uv| ≤ n und uv^i x ∈ L fuer alle i ∈ N.
Betrachte w = a^n ba^n ba^2n. Dann ist |w| > n. Sei w = uvx eine Zerlegung gem¨ass dem Pumping-Lemma. Wegen |uv| < n und v /= ε ist v = a^k fuer ein k mit 1 <= k <= n. Dann ist aber uv^0 x = a^n-k ba^n ba^2n ∈ L, da n – k + n /= 2n. Widerspruch! --> L nicht regulaer!

d)

Sei $\Sigma = \{a, b\}$ und $L := \{w \in \Sigma^* \mid w$ enthält $ab\}$. Bestimmen Sie die Äquivalenzklassen der Nerode-Relation bezüglich L auf Σ^*.

[ab]={x element von Sigma^* | x enthaelt ab}
[a]={x element von Sigma^* | x enthaelt ab nicht und endet auf a}
[epsilon]={x element von Sigma^* | x enthaelt ab nicht und endet nicht auf a}

Aufgabe 2 — 4+2+4 Punkte

a)

Betrachten Sie folgende Instanz des Post'schen Korrespondenzproblems über dem Alphabet $\{0, 1\}$:

$$K = ((100, 001), (100, 1), (10, 001), (10, 010))$$

(i) Geben Sie eine Indexfolge an, die K löst.

2,3,4

(ii) Modifizieren Sie K durch Entfernen *eines* Paares, so dass die Instanz nicht mehr lösbar ist. Begründen Sie, dass es für Ihre Modifikation keine Lösung gibt.

Entferne (100,1), dann beginen alle Paare mit verschiednen Symbolen.

b)

Gegeben sei eine Turingmaschine M mit folgender Eigenschaft: Wenn M ein Wort akzeptiert, dann geschieht das in weniger als 1000 Schritten. Kreuzen Sie für folgende Aussagen an, ob diese wahr oder falsch sind.

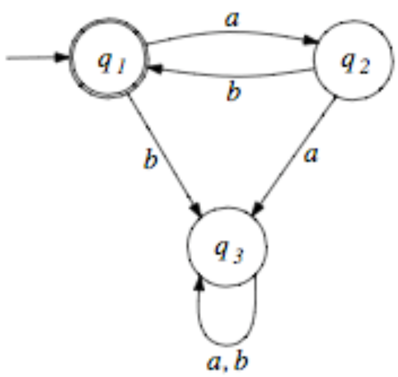
Hinweis: Für jede richtige Antwort gibt es einen Punkt, für jede falsche Antwort wird ein Punkt abgezogen. Es wird keine negative Gesamtpunktzahl für diese Teilaufgabe geben.

$L(M)$ ist entscheidbar. — Sobald TM mehr als 1000 Schritte, dann wird Wort nicht akzeptiert. — ☒ Wahr ☐ Falsch

$L(M)$ ist notwendigerweise endlich. — ☐ Wahr ☒ Falsch

c)

Geben Sie graphisch eine deterministische Turingmaschine nach Definition aus der Vorlesung an, die auf allen Eingaben über dem Alphabet $\{a, b\}$ hält und dieselbe Sprache akzeptiert wie der folgende deterministische endliche Automat:



Aufgabe 3 — 4+(2+6) Punkte

a)

Sei $A, B \in \mathrm{NP}$. Zeigen Sie, dass $A \cdot B := \{v \cdot w : v \in A \wedge w \in B\} \in \mathrm{NP}$ (Abgeschlossenheit von NP unter Konkatenation).

Da A, B in NP, existieren zwei nichtdeterministische Turing-Maschinen M_A und M_B, die A bzw. B in polynomieller Zeit akzeptieren. Wir konstruieren nun eine nichtdeterministische Turing-Maschine M, die – gestartet auf der Eingabe w der Länge n – folgendermaßen arbeitet: Wähle nichtdeterministisch eine Trennstelle i zwischen 0 und n. Simuliere nacheinander M_A auf den ersten i Buchstaben von w und M_B auf den verbleibenden Buchstaben. M akzeptiert die Eingabe w gdw. beide Simulation ihre jeweilige Eingabe akzeptieren. Die Laufzeit von M ist im wesentlichen die Summe zweier Polynome, nämlich der Laufzeiten von M_A und M_B was wiederum ein Polynom ist. Also akzeptiert M in polynomieller Zeit die Sprache $A * B$ und somit folgt $A * B$ in NP.

b)

MAX CLIQUE sei folgendes Entscheidungsproblem:

Gegeben: ein ungerichteter Graph $G = (V, E)$ und eine natürliche Zahl $k \leq |V|$.

Frage: Hat die *größte* Clique in G *genau* die Größe k?

Wir nehmen an, dass P = NP. (Diese Annahme gilt für beide Teilaufgaben.)

Zeigen Sie, dass NP = co-NP.

Aus P = NP und der Abgeschlossenheit von P bezüglich der Komplementbildung folgt NP = co-NP.

Zeigen Sie unter Verwendung von NP = co-NP, dass das Problem MAX CLIQUE in P liegt.

Zur Erinnerung, Hinweise

– Eine Clique ist eine Knotenteilmenge $V' \subseteq V$, so dass $\forall u \neq v \in V'$: $\{u, v\} \in E$.

– Beim CLIQUE Problem muss man entscheiden, ob ein gegebener Graph eine Clique einer gegebenen Größe enthält. Bekannt ist, dass CLIQUE $\in$ NP.

– co-NP := $\{L^C: L \subseteq \Sigma^* \wedge L \in \text{NP}\}$

NICHT empfohlen: Der Lösungsansatz, als erstes MAX CLIQUE $\in$ NP zu beweisen, ohne die Annahme P = NP zu verwenden, und daraus MAX CLIQUE $\in$ P mit Hilfe der Annahme zu folgern.

Wir testen, 1. ob (G, k) in CLIQUE und 2. ob $(G, k + 1)$ nicht in CLIQUE.
Wenn beides erfüllt ist, wissen wir, dass G eine Clique der Größe k enthält, aber keine Clique der Größe $k + 1$. Letzteres impliziert auch, dass es keine Clique der Größe $k' > k + 1$ geben kann, da jede solche Clique auch eine Clique der Größe $k + 1$ enthielte. Also können wir folgern, dass die größte Clique in G genau die Größe k hat, d. h. es gilt (G, k) in MAX CLIQUE. Offensichtlich gilt auch (G, k) nicht in MAX CLIQUE, wenn 1. oder 2. nicht erfüllt ist.
Bekannt ist, dass CLIQUE in NP. Also liegt der 1. Test in NP und der 2. in co-NP. Wegen P = NP = co-NP, kann daher das gesamte Verfahren deterministisch in polynomieller Zeit durchgeführt werden. Daraus folgt: MAX CLIQUE in P.

Aufgabe 4 4+3+3+2 Punkte

a)

Sei $\mathcal{G}$ die Menge aller kontextfreien Grammatiken, die mindestens ein Palindrom erzeugen, und

$$L_P = \{\text{code}(G) \mid G \in \mathcal{G}\},$$

wobei code(G) eine geeignete Kodierung von G über einem festen Alphabet bezeichne.

Zeigen Sie, dass L_P *nicht* entscheidbar ist.

Hinweis: Zu einer PKP-Instanz I kann eine kontextfreie Grammatik konstruiert werden, deren Sprache genau dann ein Palindrom enthält, wenn I eine Lösung hat.

siehe Klausur1 2004/2005 Aufgabe 2d

b)

Die Grammatiken G bzw. G' seien gegeben durch das Alphabet $\{a, b, c\}$, die Variablenmenge $\{S, T\}$, das Startsymbol S und die folgenden Regeln R bzw. R':

$$R = \{S \to aSc \mid T \mid ac, \quad T \to aTb \mid ab\}, \qquad R' = R \cup \{T \to aTc\}.$$

(a) Geben Sie einen Kellerautomaten an, der genau die Sprache $L(G)$ akzeptiert (ohne Begründung).

$Q = \{q_a, q_b, q_c, q_J\}$, Startzustand q_a, Stack-Alphabet $\{Z_0, A\}$, Stack-Init Z_0

$\delta(q_a, a, X) \to (q_a, AX)$ fuer $X \in \{Z_0, A\}$ $\qquad \delta(q_a, b, A) \to (q_b, \varepsilon)$

$\delta(q_a, c, A) \to (q_c, \varepsilon)$ $\qquad \delta(q_b, b, A) \to (q_b, \varepsilon)$

$\delta(q_b, c, A) \to (q_c, \varepsilon)$ $\qquad \delta(q_b, \varepsilon, Z_0) \to (q_J, \varepsilon)$

$\delta(q_c, c, A) \to (q_c, \varepsilon)$ $\qquad \delta(q, \varepsilon, Z) \to (q, \varepsilon)$

c)

Bringen Sie G' durch eine systematische Konstruktion auf Chomsky-Normalform.

Regelmenge: { $S \to Y_a\, C_1 \mid Y_a\, Y_c \mid Y_a\, C_2 \mid Y_a\, Y_b \mid Y_a\, C_3$, $T \to Y_a\, C_2 \mid Y_a\, Y_b \mid Y_a\, C_3$,
$C_1 \to S\, Y_c$, $C_2 \to T\, Y_b$, $C_3 \to T\, Y_c$, $Y_x \to x$ fuer $x \in \{a, b, c\}$}

d)

Zeigen Sie mit Hilfe des CYK-Algorithmus, dass das Wort *aabc* in $L(G')$ enthalten ist.

a	a	b	c
Y_a	Y_a	Y_b	Y_c
$\varnothing$	{S,T}	$\varnothing$	
$\varnothing$	$\{C_1,C_3\}$		
{S,T} --> aabc in L			

Aufgabe 5 12 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
Seien L_1, L_2 reguläre Sprachen. Dann ist auch $L_1 \cap L_2$ regulär.	X	
Die regulären Ausdrücke $(a^*b^*)^*$ und $(a^*b)^*$ sind äquivalent. a nicht erzeugbar		X
Jede Turingmaschine mit zwei Bändern kann durch eine Turingmaschine mit einem Band simuliert werden.	X	
Jeder Kellerautomat mit zwei Stacks kann durch einen Kellerautomat mit einem Stack simuliert werden. 2-PDA ist aequivalent zu TM		X
Sei L_1 semientscheidbar und L_2 entscheidbar. Dann ist $L_1 \cap L_2$ immer entscheidbar.		X
Σ^* und $\emptyset$ sind $\mathcal{NP}$-vollständig.		X
Für jede Sprache L gilt: Aus $L \in \mathcal{P}$ folgt $L^C \in \mathcal{P}$.	X	
Für jedes Optimierungsproblem Π gilt: Es gibt für Π einen Approximationsalgorithmus mit relativer Gütegarantie 2 wenn es ein PAS für Π gibt.	X	
Das KNAPSACK-Problem, eingeschränkt auf Kosten aus $\{0,1\}$ und Gewichte aus $\{3,5,7\}$ ist in $\mathcal{P}$.	X	
Sei Σ ein endliches Alphabet. Dann ist Σ^* kontextfrei.	X	
Es existiert ein NEA, der die Sprache aller Java-Programme erkennt.		X
Wenn für eine Sprache L gilt: $\exists n \in \mathbb{N} : \forall w \in L, \lvert w\rvert > n : \exists$ Zerlegung $w = uvx, \lvert uv\rvert \leq n, v \neq \varepsilon : \forall i \in \mathbb{N}_0 : uv^ix \in L$, dann ist L regulär.		X

Summe: 60 von 60 Punkten.