Universität Karlsruhe
Fakultät für Informatik
ILKD Prof. Dr. D. Wagner

Theoretische Informatik
WS 2003/04
3. Februar 2004

Musterlösungen zur Musterklausur

Aufgabe 1: (3+3+6=12 Punkte)

(a) $01010 \in L$ ☐ Wahr ☒ Falsch

$1^n \in L$ für alle $n \in \mathbb{N}_0$ ☒ Wahr ☐ Falsch

Es gibt ein $m \in \mathbb{N}_0$ mit $(10)^m 01^m \in L$ ☐ Wahr ☒ Falsch

(b) Mögliche Zustände des DEA sind Teilmengen von $\{0,1,2,3\}$. Startzustand ist $\{0,2,3\}$. Die Übergangsfunktion der erreichbaren Zustände sind in folgender Tabelle dargestellt. Endzustände sind $\{\{0,2,3\},\{1,2,3\},\{2,3\}\}$.

$\widetilde{q}$	$a \in \Sigma$	$\widetilde{\delta}(\widetilde{q},a)$
$\{0,2,3\}$	0	$\{1,2,3\}$
$\{0,2,3\}$	1	$\emptyset$
$\emptyset$	0	$\emptyset$
$\emptyset$	1	$\emptyset$
$\{1,2,3\}$	0	$\{2,3\}$
$\{1,2,3\}$	1	$\{2\}$
$\{2,3\}$	0	$\{2,3\}$
$\{2,3\}$	1	$\emptyset$
$\{2\}$	0	$\{2,3\}$
$\{2\}$	1	$\emptyset$

(c) Angenommen, die Sprache sei regulär.

Sei m die Konstante aus dem Pumping Lemma. Das Wort 0^{m^3} hat Länge größer als m. Betrachte für jede mögliche Zerlegung in uvx das Wort uv^2x. Für die Länge dieses Wortes gilt $|uv^2x| = m^3+k$, für ein k mit $0 < k < m$, und damit gilt weiter

$$\begin{aligned} m^3 &< |uv^2x| = m^3 + k \\ &< (m+1)^3 \end{aligned}$$

Falls das Wort uv^2x zur Sprache gehört, so gibt es eine natürliche Zahl l so dass $|uv^2x| = l^3$. Dann gilt $m^3 < l^3 < (m+1)^3$, was offensichtlich nicht möglich ist (denn $m^3 < l^3 \Rightarrow m < l \Rightarrow m+1 \leq l \Rightarrow (m+1)^3 \leq l^3$). Das Wort uv^2x gehört also nicht zur Sprache. Dies ist ein Widerspruch zur Aussage des Pumping Lemma, und die Annahme, dass die Sprache regulär ist, ist falsch.

Aufgabe 2: (2+2+2+6=12 Punkte)

(a) $(q_1)1010, (q_4)010, 0(q_4)10, 01(q_4)0, 010(q_4), 01(q_5)0$. Das Wort wird nicht akzeptiert.

(b) $f(0) = 1$ und $f(11111011111) = 1$.

(c) Beginnt man mit $(10, 100)$, so ist das zweite Wort länger. Egal, welches Paar man als nächstes nimmt, die beiden gebildeten Wörter stimmen am dritten Buchstaben nicht überein. Beginnt man mit $(100, 10)$, so muss man ebenfalls ein zweites Paar nehmen, und ebenfalls stimmen in beiden möglichen Fällen die Wörter an der dritten Stelle nicht überein.

(d) Angenommen, die Sprache L ist entscheidbar. Dann gibt es eine Turingmaschine $\mathcal{M}_L$, die auf allen Eingaben hält und genau die Sprache L akzeptiert. Betrachte nun das Wort $\langle\mathcal{M}_L\rangle$. Es gilt entweder $\langle\mathcal{M}_L\rangle \in L$ oder $\langle\mathcal{M}_L\rangle \notin L$.

- **Fall I:** $\langle\mathcal{M}_L\rangle \in L$. Nach Definition von L wird das Wort $\langle\mathcal{M}_L\rangle$ von $\mathcal{M}_L$ nicht akzeptiert, aber nach Definition von $\mathcal{M}_L$ werden genau die Wörter aus L akzeptiert, dieser Fall führt also zum Widerspruch.
- **Fall II:** $\langle\mathcal{M}_L\rangle \notin L$. Nach Definition von L wird nun das Wort $\langle\mathcal{M}_L\rangle$ von $\mathcal{M}_L$ akzeptiert, aber nach Definition von $\mathcal{M}_L$ wird jedes Wort, das nicht in L ist, nicht akzeptiert, wir erhalten also wieder einen Widerspruch.

Da beide Fälle zum Widerspruch führen, kann die Annahme nicht stimmen, und die Sprache L ist nicht entscheidbar.

Aufgabe 3: (6+6=12 Punkte)

(a) Es genügt zu zeigen, dass TWO-CLIQUE $\in \mathcal{NP}$ und CLIQUE $\propto$ TWO-CLIQUE. Da wir wissen, dass CLIQUE $\mathcal{NP}$-vollständig ist, ist damit auch TWO-CLIQUE $\mathcal{NP}$-vollständig.

Für zwei Teilmengen der Knoten überprüft man, ob die Teilmengen jeweils aus k verschiedenen Knoten bestehen, und ob im Graph alle Knoten der Teilmenge paarweise durch Kanten verbunden sind. Die Anzahl der zu überprüfenden Knotenpaare ist höchstens quadratisch in der Eingabegröße, also ist die Laufzeit polynomial in der Eingabegröße. Für Instanzen, die zwei Cliquen der Größe k enthalten, gibt es natürlich zwei solche Teilmengen, für die die Antwort „Ja" lautet. Für Instanzen, die nicht zu TWO-CLIQUE gehören ist die Antwort immer „Nein". Damit ist TWO-CLIQUE $\in \mathcal{NP}$.

Konstruiere nun einen Graph G', der aus zwei Kopien von G besteht. Bilde $\langle G, k\rangle$ auf $\langle G', k\rangle$ ab. Der Graph wird nur um eine Kopie von sich selbst erweitert, also ist die Abbildung in Polynomialzeit machbar. Wenn G eine Clique der Größe k enthält, so enthält G' sicherlich zwei Cliquen der Größe k – je eine in jeder Kopie von G. Umgekehrt, wenn G' zwei Cliquen der Größe k enthält, so kann jede dieser Cliquen nur Teil *einer* Kopie von G sein, denn die beiden Kopien sind durch keine Kanten verbunden. Also muss G selbst eine Clique der Größe k enthalten. Also CLIQUE $\propto$ TWO-CLIQUE.

(b) Angenommen, es gibt einen absoluten Approximationsalgorithmus $\mathcal{A}$, mit $OPT(I) - \mathcal{A}(I) \leq K$. Zu einer Instanz I konstruiere eine Instanz I', die aus $(K+1)$ Kopien des Graphen in I besteht. Die Laufzeit der Konstruktion ist polynomial, da die konstruierte Instanz $(K+1)$ mal so groß wie I ist. Es gilt $OPT(I') = (K+1)OPT(I)$, da sich eine unabhängige Menge in I' aus den $(K+1)$ unabhängigen Mengen der einzelnen Kopien zusammensetzen lässt. Nun gilt

$$OPT(I') - \mathcal{A}(I') = (K+1)OPT(I) - \mathcal{A}(I') \leq K. \tag{1}$$

In einer unabhängige Menge für I' der Größe $\mathcal{A}(I')$ müssen mindestens $\mathcal{A}(I')/(K+1)$ Knoten in einer der Kopien des Ausgangsgraphen enthalten sein, es gibt also eine unabhängige Menge für I der Größe $\mathcal{L}(I) = \lceil \mathcal{A}(I')/(K+1)\rceil$, damit gilt also

$$\mathcal{A}(I') \leq (K+1)\mathcal{L}(I). \tag{2}$$

Einsetzen von (2) in (1) ergibt

$$(K+1)OPT(I) - (K+1)\mathcal{L}(I) \leq K \tag{3}$$

und weiter

$$OPT(I) - \mathcal{L}(I) \leq K/(K+1) < 1. \tag{4}$$

Da nur ganzzahlige Werte möglich sind, bedeutet (4), dass $\mathcal{L}(I)$ der Wert einer optimalen Lösung ist, und mit Hilfe des Algorithmus $\mathcal{A}$ kann also INDEPENDENT SET in polynomialer Laufzeit optimal gelöst werden. Da $\mathcal{P} \neq \mathcal{NP}$ vorausgesetzt wurde, ist dies ein Widerspruch zur Annahme.

Aufgabe 4: (2+4+4+2=12 Punkte)

(a)

$$S \to aaSb,\ S \to aab$$

(b)

$$\begin{aligned}
\delta(q_1, a, a) &= \{(q_2, aa)\}\\
\delta(q_1, a, Z_0) &= \{(q_2, aZ_0)\}\\
\delta(q_1, b, a) &= \{(q_3, \varepsilon)\}\\
\delta(q_2, a, a) &= \{(q_1, a)\}\\
\delta(q_3, b, a) &= \{(q_3, \varepsilon)\}\\
\delta(q_3, \varepsilon, Z_0) &= \{(q_3, \varepsilon)\}
\end{aligned}$$

(c)

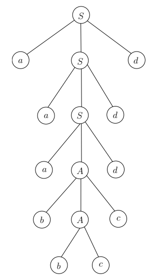
- Ersetze Buchstaben durch Variablen Y.
  $S \to Y_aSY_d | Y_aAY_d,\ A \to Y_bAY_c | Y_bY_c | \varepsilon,\ Y_a \to a,\ Y_b \to b,\ Y_c \to c,\ Y_d \to d.$

- Entferne zu lange Regeln.
  $S \to Y_aC_1,\ C_1 \to SY_d,\ S \to Y_aC_2,\ C_2 \to AY_d,\ A \to Y_bC_3,\ C_3 \to AY_c,$
  $A \to Y_bY_c | \varepsilon,\ Y_a \to a,\ Y_b \to b,\ Y_c \to c,\ Y_d \to d.$

- Entferne $A \to \varepsilon$.
  $S \to Y_aC_1,\ C_1 \to SY_d,\ S \to Y_aC_2,\ C_2 \to AY_d,\ A \to Y_bC_3,\ C_3 \to AY_c,$
  $A \to Y_bY_c,\ Y_a \to a,\ Y_b \to b,\ Y_c \to c,\ Y_d \to d,\ C_2 \to Y_d,\ C_3 \to Y_c.$

- Entferne Kettenregeln. Die Lösung lautet:
  $S \to Y_aC_1,\ C_1 \to SY_d,\ S \to Y_aC_2,\ C_2 \to AY_d,\ A \to Y_bC_3,\ C_3 \to AY_c,$
  $A \to Y_bY_c,\ Y_a \to a,\ Y_b \to b,\ Y_c \to c,\ Y_d \to d,\ C_2 \to d,\ C_3 \to c.$

(d)



Aufgabe 5: (12 Punkte)

Aussage	Wahr	Falsch
Jede endliche Sprache ist regulär.	X	
Falls $\mathcal{P} = \mathcal{NP}$, so ist 2SAT $\mathcal{NP}$-vollständig.	X	
Jedes $\mathcal{NP}$-vollständige Problem ist entscheidbar.	X	
Der Index der Nerode-Relation einer endlichen Sprache L ist echt kleiner als die Anzahl der Wörter in L.		X
Alle Sprachen sind semientscheidbar.		X
Die regulären Ausdrücke $b^*(ab^*)^*$ und $(a^* \cup b)^*$ sind äquivalent.	X	
Es gibt eine Sprache, die entscheidbar und semientscheidbar ist.	X	
Zu jeder regulären Sprache gibt es einen PDA, der sie erkennt.	X	
Das Wortproblem für kontextfreie Sprachen ist entscheidbar.	X	
Das Komplement jeder semientscheidbaren Sprache ist semientscheidbar.		X
MAX3SAT ist $\mathcal{N}P$-vollständig.	X	
Um zu zeigen, dass eine Sprache nicht regulär ist, genügt es zu zeigen, dass sie nicht kontextfrei ist.	X	