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

Theoretische Informatik
WS 2003/04
14. Januar 2003

# Lösungen zu den Musteraufgaben vom 19.12.2003

## Aufgabe 1

Die ersten drei Aussagen sind richtig, die letzte ist falsch.

*Begründung (war nicht gefragt):* Zu einer endlichen Sprache kann man leicht einen regulären Ausdruck erhalten, indem man jedes Wort als Konkatenation der einzelnen Buchstaben, und die endliche Menge der Wörter als endliche Vereinigung der einzelnen Wörter darstellt. Eine Übungsaufgabe hat gezeigt, dass falls $\mathcal{P} = \mathcal{NP}$ gilt, jede nichttriviale Sprache $\mathcal{NP}$-vollständig ist, also auch 2SAT. Da jede Sprache aus $\mathcal{NP}$ entscheidbar ist (man kann eine deterministische Turingmaschine konstruieren, die stets hält und genau die Sprache akzeptiert), ist auch jede $\mathcal{NP}$-vollständige Sprache entscheidbar. Schließlich ist z.B. der Index der Nerode-Relation zur Sprache $\{a\}$ mit einem Wort gleich drei (die Äquivalenzklassen sind $[\varepsilon]$, $[a]$ und $[aa]$).

## Aufgabe 2

Die Wörter aus a), c), e) und f) werden akzeptiert.

## Aufgabe 3

Angenommen, die Sprache sei regulär. Dann gilt die Aussage des Pumping Lemma für reguläre Sprachen, und es gibt eine natürliche Zahl $n$, so dass sich alle Wörter $w$ in der Sprache, deren Länge größer als $n$ ist, darstellen lassen als $w = uvx$, mit $|uv| \leq n$ und $v \neq \varepsilon$, so dass auch $uv^i x$ in der Sprache ist, für alle $i \geq 0$.
Wir konstruieren einen Widerspruch zu dieser Aussage: das Wort $0^{n^3}$ hat Länge größer als $n$. Betrachte für jede mögliche Zerlegung in $uvx$ das Wort $uv^2x$. Für die Länge dieses Wortes gilt $|uv^2x| = n^3 + r$, für ein $r$ mit $0 < r < n$, und damit gilt weiter

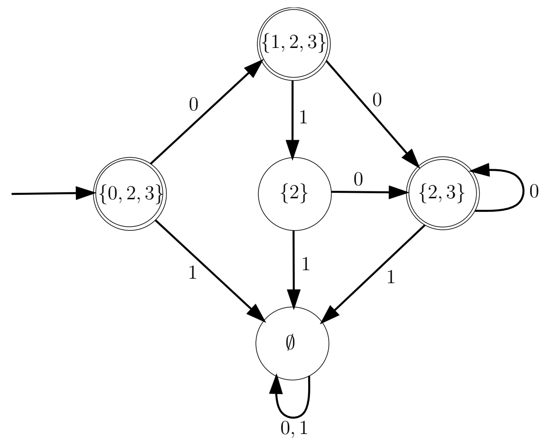
$$\begin{aligned} n^3 &< |uv^2x| = n^3 + r \\ &< (n+1)^3 \end{aligned}$$

Falls das Wort $uv^2x$ zur Sprache gehört, so gibt es eine natürliche Zahl $l$ so dass $|uv^2x| = l^3$. Dann gilt $n^3 < l^3 < (n+1)^3$, was offensichtlich nicht möglich ist (denn $n^3 < l^3 \Rightarrow n < l \Rightarrow n+1 \leq l \Rightarrow (n+1)^3 \leq l^3$). 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 4

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 und Übergangsdiagramm 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 | $\emptyset$ |




## Aufgabe 5

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 6

Wir zeigen die $\mathcal{NP}$-Vollständigkeit von TWO-CLIQUE indem wir zeigen, dass (i) TWO-CLIQUE $\in \mathcal{NP}$ und (ii) CLIQUE $\propto$ TWO-CLIQUE gilt.
Zu (i) seien zusätzlich zu dem gegebenen Graphen und der Zahl $k$ zwei Teilmengen der Knoten gegeben. Es wird zuerst überprüft, ob die Teilmengen jeweils aus $k$ verschiedenen Knoten bestehen, und dann für jede Teilmenge überprüft, ob im Graph alle Knoten der Teilmenge paarweise durch Kanten verbunden sind. Insgesamt sind dies also höchstens $O(|V|^2)$ Überprüfungen, der Aufwand ist also polynomial 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 $\in \mathcal{NP}$.
Zu (ii) sei eine Instanz $\langle G, k \rangle$ von CLIQUE gegeben. Der Graph $G'$ bestehe aus zwei Kopien von $G$, und wir bilden nun $\langle G, k \rangle$ auf $\langle G', k \rangle$ ab und betrachten das Bild $\langle G', k \rangle$ als Instanz von TWO-CLIQUE. Die Abbildung ist in Polynomialzeit möglich, da lediglich der Graph um eine Kopie von sich selbst erweitert wird. Es gilt nun

$$\langle G, k \rangle \in \text{CLIQUE} \;\Leftrightarrow\; \langle G', k \rangle \in \text{TWO-CLIQUE},$$

denn 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 (die beiden Kopien sind durch keine Kanten verbunden). Also muss $G$ selbst eine Clique der Größe $k$ enthalten.

## Aufgabe 7

Wir zeigen (analog zum Vorgehen bei KNAPSACK in der Vorlesung), dass wenn es einen solchen Approximationsalgorithmus gibt, das Optimalwertproblem INDEPENDENT SET polynomial lösbar ist, was $\mathcal{P} = \mathcal{NP}$ zur Folge hätte. Wir nehmen also an, dass es einen absoluten Approximationsalgorithmus $\mathcal{A}$ gibt, mit $OPT(I) - \mathcal{A}(I) \leq K$, wobei $OPT(I)$ der Wert einer optimalen Lösung für die Instanz $I$ ist, und $K$ eine natürliche Zahl größer als Null. Zu einer gegebenen Instanz $I$ konstruieren wir nun eine Instanz $I'$, die aus $(K+1)$ Kopien des Graphen in $I$ besteht. Die Laufzeit der Konstruktion und die Größe von $I'$ ist $(K+1)$ mal die Größe von $I$. Da $K$ fest vorgegeben ist, ist dies polynomial in der Größe von $I$. Für den Optimalwert von $I'$ 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. Wir wenden nun den Algorithmus $\mathcal{A}$ auf $I'$ an, und wissen, dass es eine unabhängige Menge in $I'$ gibt der Größe $\mathcal{A}(I')$, für die 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, ein Widerspruch zu $\mathcal{P} \neq \mathcal{NP}$.