UNIVERSITÄT KONSTANZ
Fachbereich Informatik & Informationswissenschaft
Prof. Dr. D. Wagner / C. Fieß / M. Gaertler / F. Schulz

Lösungen zum 12. Übungsblatt

Aufgabe 1:
Beh.: Im Falle $\mathcal{P} \neq \mathcal{NP}$ gibt es keinen polynomialen ε-approximativen Algorithmus für die Optimierungsversion des TSP.

Bew. (sehr ausführlich): **Angenommen**, es gibt einen ε-approximativen Algorithmus $\mathcal{A}$ für die Optimierungsversion des TSP. Also (TSP ist Minimierungsproblem!) gilt für jedes Problembeispiel (G', c)

$$\frac{\mathcal{A}(G', c)}{OPT_{TSP}(G', c)} \leq (1 + \varepsilon)$$

Wir konstruieren nun einen polynomialen Algorithmus $\mathcal{B}$ für das Entscheidungsproblem HAMILTON CYCLE, im Widerspruch zu $\mathcal{P} \neq \mathcal{NP}$. Sei also ein Problembeispiel für HAMILTON CYCLE gegeben, ein Graph $G = (V, E)$. $\mathcal{B}$ muss entscheiden, ob G einen Hamilton Cycle enthält. In **Schritt 1** konstruieren wir einen Graph G' und eine Kostenfunktion c, was wir als Eingabe für $\mathcal{A}$ verwenden werden:

$$\begin{aligned}
G' &= (V', E') \\
V' &= V \\
E' &= \{\{u, v\} : u, v \in V, u \neq v\} \qquad (G' \text{ ist also vollständiger Graph}) \\
c(\{u, v\}) &= \begin{cases} 1 & \text{falls } \{u, v\} \in E \\ 2 + \varepsilon \cdot |V| & \text{sonst} \end{cases}
\end{aligned}$$

Im Anschluss an diese Konstruktion wird in **Schritt 2** des Algorithmus $\mathcal{B}$ der Algorithmus $\mathcal{A}$ ausgeführt, mit Eingabe (G', c). Es gilt dann:

- **Fall I:** G enthält einen Hamilton Cycle. Dann ist dieser Kreis eine Rundreise in G' mit Kosten $|V|$ (ein Kreis hat genauso viele Kanten wie Knoten und alle Kanten des Kreises sind in E, also haben alle Kanten in diesem Kreis Gewicht 1), also ist das Optimum für TSP sicher kleiner als $|V|$: $OPT_{TSP}(G', c) \leq |V|$. Weil $\mathcal{A}$ ε-approximativ ist, gilt weiter

$$\begin{aligned}
& \tfrac{\mathcal{A}(G', c)}{OPT_{TSP}(G', c)} \leq (1 + \varepsilon) \\
\Rightarrow \quad & \mathcal{A}(G', c) \leq (1 + \varepsilon) OPT_{TSP}(G', c) \\
\Rightarrow \quad & \mathcal{A}(G', c) \leq (1 + \varepsilon)|V|
\end{aligned}$$

- **Fall II:** G enthält keinen Hamilton Cycle. Dann ist für die optimale Rundreise in (G', c) nicht jede Kante in E (sonst hätten wir einen Hamilton Cycle in G gefunden!) Die Rundreise enthält also alle $|V|$ Knoten und $|V|$ Kanten (die Rundreise ist ein Kreis), wobei eine Kante nicht in E ist. Für mindestens eine Kante e in der Rundreise ist also $c(e) = 2 + \varepsilon \cdot |V|$, und damit $OPT_{TSP}(G', c) \geq |V| - 1 + (2 + \varepsilon \cdot |V|)$. Da $\mathcal{A}(G', c)$ sicher größer als das Optimum ist (nur Approximation), folgt weiter

$$\begin{aligned}
\mathcal{A}(G', c) &\geq OPT_{TSP}(G', c) \geq |V| + 1 + \varepsilon \cdot |V| \\
&= (1 + \varepsilon)|V| + 1 \\
&> (1 + \varepsilon)|V|
\end{aligned}$$

Damit können wir Algorithmus $\mathcal{B}$ mit folgendem **Schritt 3** beenden:

1. falls $\mathcal{A}(G', c) \leq (1 + \varepsilon)|V|$, so enthält G einen Hamilton Cycle (angenommen G enthielte keinen, so wäre laut Fall II $\mathcal{A}(G', c) > (1 + \varepsilon)|V|$). Die Ausgabe ist also **ja**.

2. falls $\mathcal{A}(G', c) > (1 + \varepsilon)|V|$, so enthält G keinen Hamilton Cycle (angenommen G enthielte einen, so wäre laut Fall I $\mathcal{A}(G', c) \leq (1 + \varepsilon)|V|$). Die Ausgabe ist also **nein**.

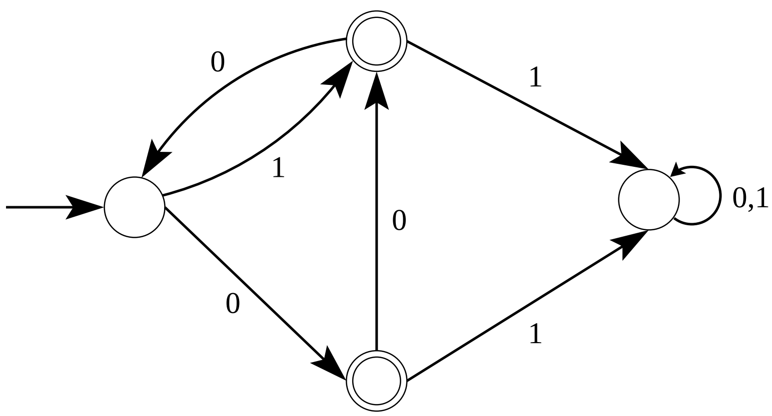
Der Algorithmus $\mathcal{B}$ liefert also wie gewünscht die Entscheidung, ob G einen Hamilton Cycle enthält. Es bleibt noch zu zeigen, dass $\mathcal{B}$ eine Laufzeit hat, die polynomial in der Eingabegröße $G = (V, E)$ ist.

Der erste Schritt ist beschränkt durch $\mathcal{O}(|V|^2)$, da ein vollständiger Graph mit Knoten $V' = V$ konstruiert wird, und dieser Graph G' genau $|V|$ Knoten und $|V|^2$ Kanten enthält. Für jede Kante wird ein Kantengewicht $c(e)$ gesetzt, also auch dafür Laufzeit in $\mathcal{O}(|V|^2)$. Im zweiten Schritt wird $\mathcal{A}(G', c)$ ausgeführt, was nach Voraussetzung ($\mathcal{A}$ ist ε-approximativ, d.h. insbesondere polynomial in der Größe der Eingabe (G', c)) auch polynomial in der Größe von G ist, weil die Größe von (G', c) polynomial in der Größe von G ist. Schließlich wird nur noch ein Vergleich ausgeführt, nämlich das Ergebnis von $\mathcal{A}$ wird verglichen mit $(1 + \varepsilon)|V|$. Insgesamt ist also die Laufzeit von $\mathcal{B}$ polynomial in der Größe der Eingabe G.

Wir haben also einen polynomialen Algorithmus $\mathcal{B}$ gefunden, der das Problem HAMILTON CYCLE in polynomialer Laufzeit löst, also HAMILTON CYCLE $\in \mathcal{P}$, und damit $\mathcal{P} = \mathcal{NP}$ (HAMILTON CYCLE ist $\mathcal{NP}$-vollständig, und siehe Vorlesung, Lemma 4.18). $\lightning$ Widerspruch zur Voraussetzung $\mathcal{P} \neq \mathcal{NP}$, und damit gilt die Annahme nicht, es gibt also keinen ε-approximativen Algorithmus für TSP.

$\blacksquare$

Aufgabe 2: Achtung: Dies sind nur die Ergebnisse ohne Konstruktionsschritte und Begründungen.

a) 

b) $(000 \cup 10)^*(0 \cup 00 \cup 1)$

c) Der Index der Nerode-Relation ist 4

Aufgabe 3:

Beh.: $L = \{0^{(k^3)}; k \in I\!N_0\}$ ist nicht regulär.

Bew.: Wir benützen zum Beweis das Pumping Lemma für reguläre Sprachen. Dazu **nehmen wir an**, L sei regulär. Dann gibt es laut Pumping Lemma eine natürliche Zahl n, so dass für alle Wörter w aus L, deren Länge größer als n ist, gilt: $w = uvx$, wobei $|uv| \leq n$ und $v \neq \varepsilon$, und für alle $i \in I\!N_0$ ist $uv^ix \in L$.

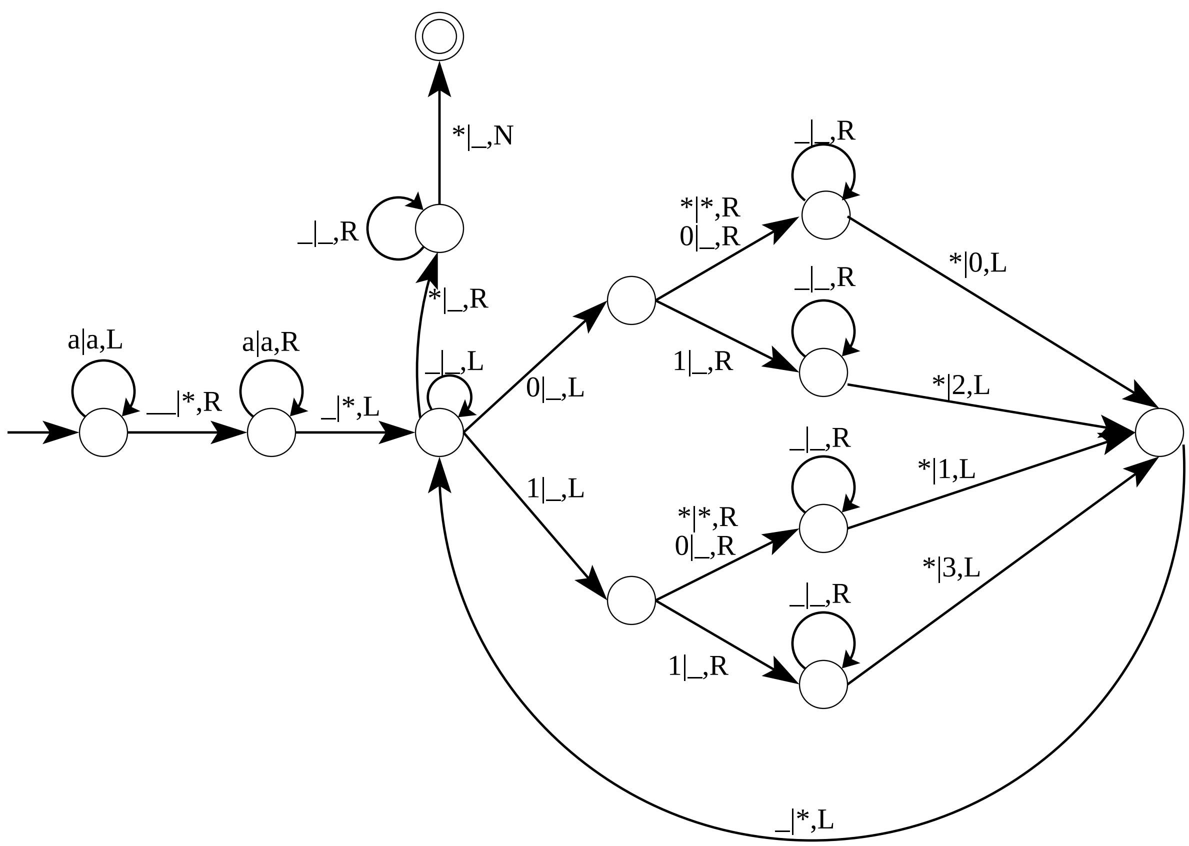
Wir betrachten nun das Wort $w' = 0^{(n^3)}$ (wichtig: n ist das n von oben, das es laut Pumping Lemma gibt!) Es ist natürlich $w' \in L$, und $|w'| > n$. Also trifft obige Zerlegung auch für unser gewähltes w' zu, und es gibt eine Zerlegung $w' = uvx$ mit $|uv| \leq n$ und $v \neq \varepsilon$. Betrachte nun das Wort uv^2x, für welches laut Pumping Lemma gilt $uv^2x \in L$ ist. Für die Länge von uv^2x gilt

$$n^3 < |uv^2x| \leq n^3 + n < n^3 + 3n^2 + 3n + 1 = (n+1)^3$$

und die Länge von uv^2x ist nicht gleich k^3 für ein $k \in I\!N_0$. Also $uv^2x \notin L$ ↯ Widerspruch zu $uv^2x \in L$, die Annahme gilt nicht, und damit ist L nicht regulär.

■

Aufgabe 4: Es gilt $\Sigma = \{0, 1\}$ und $\Gamma = \{0, 1, \_, *\}$. Zur Notation: _ ist das Blank-Symbol, und a steht für ein beliebiges Symbol des Bandalphabets Γ, für das aus einem Zustand kein anderer Übergang herausführt.



Aufgabe 5:

Beh.: Die Sprache

$$L = \{\langle\mathcal{M}\rangle\ ;\ \mathcal{M} \text{ gibt entweder } \texttt{0} \text{ oder } \texttt{1} \text{ aus auf beliebige Eingabe } w\},$$

deren Wörter die Gödelnummern $\langle\mathcal{M}\rangle$ sind, deren zugehörige Turingmaschine $\mathcal{M}$ auf eine beliebige Eingabe $w \in \{0,1\}^*$ entweder eine `0` oder eine `1` ausgibt, ist nicht entscheidbar.

Bew.: Wir verwenden den **Satz von Rice** (s. Vorlesung, Satz 3.15). Die Menge aller Turing-berechenbaren Funktionen war definiert als

$$R = \{f : \Sigma^* \to \Sigma^* \mid \text{ es gibt } \langle\mathcal{M}\rangle \text{ wobei } \mathcal{M} \text{ die Funktion } f \text{ berechnet}\}.$$

Da für Gödelnummern $\langle\mathcal{M}\rangle$ die Turingmaschinen „standardisiert” sind, ist das Alphabet hier $\Sigma = \{0,1\}$. Für den Satz von Rice verwenden wir nun weiter die Teilmenge von Turing-berechenbaren Funktionen

$$S := \{f : \Sigma^* \to \Sigma^* \mid f \in R \text{ und } f(w) = 0 \text{ oder } f(w) = 1 \text{ für alle } w \in \Sigma^*\}$$

Es gilt $S \neq \emptyset$ und $S \neq R$, und mit dem Satz von Rice ist die Sprache

$$L(S) = \{\langle\mathcal{M}\rangle\ ;\ \mathcal{M} \text{ berechnet eine Funktion aus } S\}$$

nicht entscheidbar. Da eine Turingmaschine $\mathcal{M}$ genau dann für alle Eingaben eine `0` oder eine `1` ausgibt, wenn sie eine Funktion aus S berechnet, ist die gesuchte Sprache L gleich der Sprache $L(S)$, und auch L ist nicht entscheidbar.

■