Lehrstuhl für Informatik 1
Prof. Dr. Gerhard Woeginger
Jan Böker, Tim Hartmann

WS 2018/19

Übung zur Vorlesung
Berechenbarkeit und Komplexität

Lösung Blatt 9

Tutoriumsaufgabe 9.1

Wir geben eine alternative Laufzeitdefinition für nichtdeterministische Turingmaschinen an. Für eine NTM M und eine Eingabe x sei

$$T'_M(x) := \textbf{Länge des längsten Rechenweges von } M \textbf{ auf } x.$$

Falls ein nicht-terminierender Rechenweg auf x existiert, setzen wir $T'_M(x) := \infty$. Daraus erhalten wir die Worst-Case-Laufzeit $t'_M(n) := \max\{T'_M(x) \mid x \in \Sigma^n\}$. Sei nun NP' die Klasse der Entscheidungsprobleme, die durch eine NTM M erkannt werden, deren Worst-Case-Laufzeit $t'_M(n)$ polynomiell beschränkt ist.

Zeigen Sie, dass $\mathsf{NP} = \mathsf{NP}'$.

Wir zeigen zuerst die Inklusion $\mathsf{NP}' \subseteq \mathsf{NP}$. Aus der Definition von $T_M(x)$ und $T'_M(x)$ folgt $T_M(x) \leq T'_M(x)$ und daher auch $t_M(n) \leq t'_M(n)$. Für jede NTM M gilt also: Wenn $t'_M(n)$ polynomiell beschränkt ist, dann ist auch $t_M(n)$ polynomiell beschränkt. Daher gilt $\mathsf{NP}' \subseteq \mathsf{NP}$.

Nun zeigen wir die Inklusion $\mathsf{NP} \subseteq \mathsf{NP}'$. Sei L eine Sprache mit $L \in \mathsf{NP}$. Diese Sprache wird also durch eine NTM M erkannt, deren Worst-Case-Laufzeit $t_M(n)$ durch ein Polynom $p(n)$ beschränkt ist. Wir konstruieren eine NTM M' mit $L(M') = L(M) = L$, so dass $t'_{M'}(n)$ polynomiell beschränkt ist, was $L \in \mathsf{NP}'$ impliziert.

Auf einer Eingabe x simuliert M' das Verhalten von M, zählt dabei aber die Schritte des Rechenweges mit und verwirft die Eingabe, sobald $p(|x|)$ Schritte simuliert wurden (und M noch nicht akzeptiert hat): Falls M die Eingabe x akzeptiert, so existiert **ein** Rechenweg, auf dem sie dies nach höchstens $p(|x|)$ Schritten tut. Damit akzeptiert auch M' diese Eingabe x. Die Umkehrung gilt auch: Wenn M' eine Eingabe x akzeptiert, so akzeptiert auch M diese Eingabe. Folglich gilt $L(M') = L(M) = L$.

Detailliertere Konstruktion: Wir zunächst eine 3-Band-NTM, die sich folgendermaßen verhält: Für eine Eingabe x berechnet sie zuerst auf dem zweiten Band $p(|x|)$, was in Polynomialzeit möglich ist. Dann arbeitet sie auf dem ersten Band wie M und zählt dabei auf dem dritten Band die Rechenschritte. Sobald die Schrittanzahl größer ist als die auf dem zweiten Band gespeicherte Zahl, verwirft die Maschine die Eingabe. Der Laufzeitverlust ist dabei in jedem Schritt höchstens $O(p(|x|))$. Analog zum in der Vorlesung vorgestellten Verfahren kann die 3-Band-NTM in eine (1-Band-)NTM umgewandelt werden, wobei die Länge jedes Rechenwegs nur quadratisch wächst.

Weil die NTM M' auf einer Eingabe der Länge n **jeden** Rechenweg von M nur für $p(n)$ Schritte simuliert und dabei nur ein polynomieller Laufzeitverlust entsteht, ist $t'_{M'}(n)$ polynomiell beschränkt. Daher gilt $L(M') = L(M) = L \in \mathsf{NP}'$.

Tutoriumsaufgabe 9.2

Zeigen Sie, dass folgendes Problem in NP liegt:

> **Problem: GI**
> **Eingabe: Zwei ungerichtete Graphen $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$**
> **Frage: Ist G_1 isomorph zu G_2, d. h., existiert eine Bijektion $f : V_1 \to V_2$, sodass $\{u, v\} \in E_1$ genau dann wenn $\{f(u), f(v)\} \in E_2$?**

Wir geben ein Zertifikat polynomieller Länge und einen Polynomialzeit-Verifizierer an.

Zertifikat: Wir kodieren den Isomorphismus f durch den String

$$bin(v_1)\#bin(f(v_1))\#\#bin(v_2)\#bin(f(v_2))\#\# \ldots \#\#bin(v_n)\#bin(f(v_n))$$

wobei $V_1 = \{v_1, \ldots, v_n\}$ ist.

Die Länge des Zertifikats ist $\mathcal{O}(n \log n)$, also polynomiell in der Eingabelänge.

Verifizierer: Prüfe zunächst ob die Eingabe das richtige Format hat. Dann prüfe ob f bijektiv ist. Dazu teste zunächst ob $|V_1| = |V_2|$ (verwirf falls nicht). Dann durchlaufe das Zertifikat und prüfe ob $i \neq j \in \{1, \ldots, n\}$ existieren mit $f(v_i) = f(v_j)$. Wenn ja, dann verwirf. Nun prüfe für jedes Paar $u, v \in V_1$, ob $\{u, v\} \in E_1$ genau dann, wenn $\{f(u), f(v)\} \in E_2$. Wenn ja, dann akzeptiere.

Wir analysieren die Laufzeit des Verifizierers. Das Format kann in polynomieller Zeit geprüft werden. Dies gilt auch für den Test, ob $|V_1| = |V_2|$. Dananch durchläuft der Algorithmus maximal n^2 Paare i, j und macht für jedes Paar zwei Lookups im Zertifikat. Dann werden nochmal n^2 Paare durchlaufen, wobei jeweils zwei Lookups in der Adjazenzmatrix durchgeführt werden. Insgesamt ergibt sich also eine polynomielle Laufzeit.

Korrektheit: Sind die Graphen isomorph, so bringt ein kodierter Isomorphismus den Verifizierer zum Akzeptieren. Gibt es umgekehrt ein Zertifikat, das den Verifizierer zum Akzeptieren bringt, so folgt direkt, dass dieses Zertifikat einen Isomorphismus kodiert; die Graphen sind also isomorph.

Also ist GI in NP.

Tutoriumsaufgabe 9.3

Es sei $G = (V_G, E_G)$ ein ungerichteter Graph. Eine Menge $D \subseteq V_G$ heißt *Dominating Set* von G, wenn jeder Knoten in V_G in D liegt oder zu einem Knoten in D benachbart ist. Eine Eingabe des Problems DOMINATING SET besteht aus einem Graphen G und einer Zahl k. Die Frage ist, ob G ein Dominating Set der Größe höchstens k besitzt.

Für einen Graphen $G = (V_G, E_G)$ konstruieren wir einen neuen Graphen H mit Knotenmenge $\{v \in V_G \mid v \text{ nicht isoliert in } G\} \cup \{w^e \mid e \in E_G\}$. Der Graph H enthält alle Kanten in E_G, und außerdem ist für jede Kante $e = \{u, v\} \in E_G$ der Knoten w^e zu den beiden Knoten u und v benachbart.

(a) **Zeigen Sie: Wenn der Graph $\boldsymbol{G}$ ein Vertex-Cover der Größe höchstens $\boldsymbol{k}$ besitzt, dann besitzt $\boldsymbol{H}$ ein Dominating Set der Größe höchstens $\boldsymbol{k}$.**

Es sei $S \subseteq V_G$ ein Vertex-Cover von G der Größe höchstens k. Es wird gezeigt, dass das Vertex-Cover $S' := S \setminus \{v \in V_G \mid v \text{ isoliert in } G\}$ von G (der Größe höchstens k) ein Dominating Set von H ist.

Bertrachte zunächste die „alten“ Knoten von H. Es sei $v \in \{v \in V_G \mid v \text{ nicht isoliert in } G\} \setminus S'$. Es ist zu zeigen, dass v einen Nachbarn in S' hat. Da v nicht isoliert in G ist, hat v einen Nachbarn u in G. Da S' ein Vertex-Cover von G ist, ist v oder u Element von S'; nach Annahme gilt $v \notin S'$, also muss dies bereits u sein.

Bertrachte nun die „neuen“ Knoten von H. Es sei w^e für $e = \{u, v\} \in E_G$ ein solcher Knoten von H. Da S' Vertex-Cover von G ist, ist u oder v Element von S'. Da w^e mit beiden Knoten verbunden ist, hat der Knoten also einen Nachbarn in S'.

Also ist S' ein Dominating Set von H.

(b) **Zeigen Sie: Wenn der Graph $\boldsymbol{H}$ ein Dominating Set der Größe höchstens $\boldsymbol{k}$ besitzt, dann besitzt $\boldsymbol{G}$ ein Vertex-Cover der Größe höchstens $\boldsymbol{k}$.**

Es sei $S \subseteq \{v \in V_G \mid v \text{ nicht isoliert in G}\} \cup \{w^e \mid e \in E_G\}$ ein Dominating Set von H der Größe höchstens k. Konstruiere aus S ein Vertex-Cover von G wie folgt: Für $e = \{u, v\} \in E_H$ sind die Knoten u und v die einzigen Nachbarn von w^e in H, weshalb einer dieser drei Knoten in S enthalten sein muss. Falls $w^e \notin S$ gilt, so muss nichts getan werden, da dann u oder v in S enthalten ist. Ist $w^e \in S$, so entferne w^e und füge entweder u oder v hinzu. (Das Hinzufügen ist nur wichtig, falls beide Knoten noch nicht vorhanden sind. Falls einer oder beide schon vorhanden sind, wird die Menge insgesamt aber auch nicht größer. Falls der hinzugefügte Knoten schon vorhanden ist, wird sie sogar kleiner.)

Wiederholt man obige Überlegung für jedes $e \in E_H$, so erhält man ein Vertex-Cover von G der Größe höchstens k.

(c) **Zeigen Sie: VERTEX COVER $\boldsymbol{\leq_p}$ DOMINATING SET**

Reduktion f:

- Ungültige Kodierungen für VERTEX COVER werden auf ungültige Kodierungen für DOMINATING SET abgebildet
- Eingaben (G, k) werden auf (H, k) abgebildet.

Die Reduktion ist in Polynomialzeit berechenbar: Für einen Graphen G mit n Knoten und m Kanten hat der Graph H höchstens $n + m$ Knoten und höchstens $m + m + n$ Kanten. H lässt sich einfach in polynomieller Zeit aus G berechnen, indem man eine Kopie der Adjazenzmatrix von G, aus der isolierte Knoten entfernt wurden, um die entsprechenden m neuen Knoten ergänzt.

Korrektheit: Für eine ungültige Kodierung gilt die Korrektheit trivialerweise. Für eine Eingabe (G, k) gilt nach (a) und (b), dass G ein Vertex-Cover der Größe höchstens k besitzt gdw. H ein Dominating Set der Größe höchstens k besitzt. Damit gilt auch in diesem Fall die Korrektheit.

Hausaufgabe 9.1 **(3 Punkte)**