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

WS 2018/19

Übung zur Vorlesung
Berechenbarkeit und Komplexität

Lösung Blatt 9

Hausaufgabe 9.1 (3 Punkte)

Die Sprache L enthält die Gödelnummern $\langle M \rangle$ jener Turingmaschinen M, für die ein Wort $w \in \{0,1\}^*$ existiert, auf dem M nach höchstens $|w|^3$ vielen Schritten anhält.

Beschreiben Sie eine NTM, die L erkennt. Da aus jeder NTM eine (deterministische) TM konstruiert werden kann, die dieselbe Sprache erkennt, genügt dies, um zu zeigen, dass L semi-entscheidbar ist.

Die NTM verhalte sich wie folgt:

(a) Stelle sicher, dass es sich bei der Eingabe um eine Gödelnummer einer Turingmaschine M handelt.

(b) „Rate“ ein Wort $w \in \{0,1\}^*$. Verwende dazu einen Zustand, von dem aus nichtdeterministisch entschieden wird, ob das Wort schon zu Ende geraten wurde oder nicht. Ist das Wort noch nicht zu Ende geraten, so entscheide nichtdeterministisch, ob als nächstes Symbol eine 0 oder eine 1 auf das Band geschrieben wird.

(c) Simuliere M auf w für höchstens $|w|^3$ viele Schritte und akzeptiere falls M hält.

Korrektheit: Wenn $\langle M \rangle \in L$ gilt, so gibt es ein Wort $w \in \{0,1\}^*$, auf dem M nach höchstens $|w|^3$ Schritten hält. Dieses Wort wird dann auch von der NTM erraten, die Simulation führt zur Akzeptanz und die NTM akzeptiert. Für die andere Richtung gilt, dass, wenn die NTM akzeptiert, die Eingabe eine Gödelnummer $\langle M \rangle$ einer Turingmaschine M ist, für die ein Wort $w \in \{0,1\}^*$ existiert, auf dem sie nach höchstens $|w|^3$ Schritten akzeptiert.

Hausaufgabe 9.2 (4 Punkte)

Zeigen Sie, dass folgendes Problem in NP liegt:

Problem: COMPOSITE
EINGABE: Eine natürliche Zahl n (kodiert als Binärzahl)
FRAGE: Ist n keine Primzahl?

Sie dürfen dabei nicht verwenden, dass ein Primzahltest in polynomieller Zeit möglich ist.

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

Zertifikat: Wir kodieren einen nichttrivialen Teiler $1 < a < n$ von n durch den String $bin(a)$. Da $a < n$, ist die Länge des Zertifikats $\mathcal{O}(log(n))$, also polynomiell in der Eingabelänge.

Verifizierer: Prüfe zunächst, ob die Eingabe das richtige Format hat. (Wenn man Vornullen ignoriert, dann ist jeder Binärstring eine gültige Kodierung einer Zahl. In diesem Fall muss man nur testen, ob genau ein # vorkommt, das das Zertifikat von der eigentlichen Eingabe trennt.) In den Fällen $n = 0$ und $n = 1$ wird direkt akzeptiert. Sonst wird geprüft, ob $1 < a < n$ und ob die Zahl a die Zahl n teilt. Wenn ja, dann akzeptiere. Sonst verwirf.

Wir analysieren die Laufzeit des Verifizierers. Das Format kann in Polynomialzeit geprüft werden. Dies gilt auch für den Test, ob $1 < a < n$, und für den Test, ob die Zahl a die Zahl n teilt; Division ist in polynomieller Zeit möglich (Schriftliche Division, wie in der Schule).

Korrektheit: Ist n keine Primzahl, so gilt $n = 0$, $n = 1$ oder, dass n einen nichttrivialen Teiler a mit $1 < a < n$ hat. In den Fällen $n = 0$ und $n = 1$ wird unabhängig vom Zertifikat akzeptiert. Im dritten Fall erhält man durch $bin(a)$ ein Zertifikat, das den Verifizierer zum Akzeptieren bringt. Existiert umgekehrt ein Zertifikat, das den Verifizierer zum Akzeptieren bringt, so gilt $n = 0$, $n = 1$ oder, dass dieses Zertifikat einen nichttrivialen Teiler von n kodiert, d. h., n ist nicht prim.

Also ist COMPOSITE in NP.

Hausaufgabe 9.3 (5 Punkte)

Da Weihnachten naht, muss der Weihnachtsmann seine Rentierschlitten auf das Geschenkeverteilen vorbereiten, wozu je zwei Rentiere auf jeweils einen Rentierschlitten verteilt werden müssen. Die Rentiere des Weihnachtsmannes sind allerdings bezüglich der Wahl ihres Schlittenpartners sehr wählerisch und teilen dem Weihnachtsmann mit, mit welchen anderen Rentieren sie als Schlittenpartner einverstanden sind. Der Weihnachtsmann möchte die Entscheidungen der Rentiere respektieren, aber dennoch so viele Schlitten wie möglich besetzen. Dazu modelliert er dieses Problem als ungerichteten Graphen: Die Knoten sind durch die Menge der Rentiere gegeben, und zwei Rentiere sind durch eine Kante verbunden, wenn sie dem Weihnachtsmann das gegenseitige Einverständnis mitgeteilt haben. Dies führt den Weihnachtsmann zu folgendem Entscheidungsproblem:

Es sei $G = (V_G, E_G)$ ein ungerichteter Graph. Eine Menge $M \subseteq E_G$ heißt *Matching* in G, wenn keine zwei Kanten in M einen gemeinsamen Knoten haben. Eine Eingabe des Problems MATCHING besteht aus einem Graphen G und einer Zahl k. Die Frage ist, ob G ein Matching der Größe mindestens k besitzt.

Für einen Graphen $G = (V_G, E_G)$ konstruieren wir einen neuen Graphen $H = (V_H, E_H)$ mit Knotenmenge $V_H = \{w^e \mid e \in E_G\}$. Zwei Knoten w^{e_1} und w^{e_2} in H sind adjazent genau dann, wenn die zugrundeliegenden Kanten e_1 und e_2 im Graphen G einen gemeinsamen Knoten haben.

(a) **Zeigen Sie: Wenn der Graph G ein Matching der Größe k besitzt, dann besitzt H ein Independent Set der Größe k.**

Es sei $M \subseteq E_G$ ein Matching von G der Größe k. Es wird gezeigt, dass die Menge $M' \coloneqq \{w^e \mid e \in M\}$ der Größe k ein Independent Set von H ist. Angenommen, M' ist kein Independent Set, d. h., es gibt zwei Knoten $w^{e_1}, w^{e_2} \in M'$, die in H benachbart sind. Also haben die Kanten $e_1, e_2 \in M$ einen gemeinsamen Knoten in G. Folglich ist M kein Matching, was ein Widerspruch ist.

(b) Zeigen Sie: Wenn der Graph H ein Independent Set der Größe k besitzt, dann besitzt G ein Matching der Größe k.

Es sei $S \subseteq V_H$ ein Independent Set von H der Größe k. Es wird gezeigt, dass die Menge $S' \coloneqq \{e \in E_G \mid w^e \in S\}$ der Größe k ein Matching von G ist. Angenommen, S' ist kein Matching, d. h., es gibt zwei Kanten $e_1, e_2 \in S'$, die in G einen gemeinsamen Knoten haben. Damit sind $w^{e_1}, w^{e_2} \in S$ nach Definition in H adjazent. Folglich ist S kein Independent Set, was ein Widerspruch ist.

(c) Zeigen Sie: MATCHING $\leq_p$ INDEP-SET

Reduktion f:

- Ungültige Kodierungen für MATCHING werden auf ungültige Kodierungen für INDEP-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 genau m Knoten und höchstens m^2 Kanten. Durch eine Schleife über alle m^2 Kantenpaare lässt sich H einfach aus G in polynomieller Zeit konstruieren.

Korrektheit: Für eine ungültige Kodierung gilt die Korrektheit trivialerweise. Für eine Eingabe (G, k) folgt aus (a) und (b), dass G ein Matching der Größe mindestens k besitzt gdw. H ein Independent Set der Größe mindestens k besitzt, indem man die Aussagen für alle $k' \geq k$ betrachtet. Damit gilt auch in diesem Fall die Korrektheit.