Prof. T. Nipkow
K. Kappelmann, J. Rädle, L. Stevens

KEINE ABGABE

Technische Universität München
Lehrstuhl für Logik und Verifikation

Einführung in die Theoretische Informatik

Sommersemester 2020 – Übungsblatt Lösungsskizze 11

AUFGABE 11.1. (*Wichtige Begriffe*)

Überprüfen Sie, dass Sie die Folgenden Begriffe korrekt definieren können.

Stufe A

- Polynomielle Reduktion $\leq_p$
- NP-schwer
- NP-vollständig
- SAT
- 3-KNF-SAT

AUFGABE 11.2. ($\mathsf{P} = \mathsf{NP} \overset{\mathsf{P}\neq 0}{\iff} \frac{\mathsf{P}}{\mathsf{P}} = \frac{\mathsf{NP}}{\mathsf{P}} \iff \frac{1}{1} = \frac{\mathsf{N}}{1} \iff 1 = \mathsf{N} \quad \square$)

Diskutieren Sie die folgenden Aussagen:

Stufe B

(a) Für jede Sprache, die von einer NTM in polynomieller Zeit akzeptiert werden kann, existiert eine DTM, die ebenfalls die Sprache in polynomieller Zeit akzeptiert.

(b) Wenn A NP-schwer ist, dann gilt $A \in \mathbf{P}$.

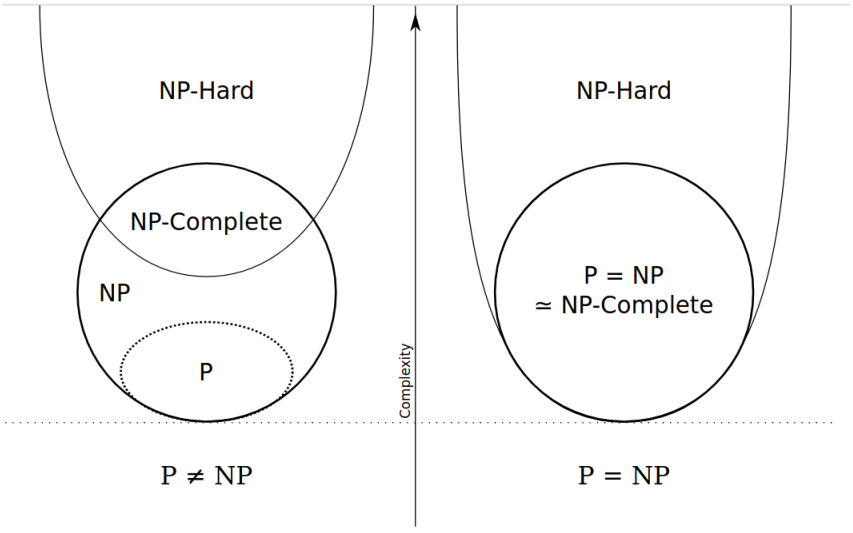
Die textuelle Lösung finden Sie vorab auf Moodle/der Vorlesungswebsite.

Lösungsskizze

(a) Die Frage lässt sich formulieren als $\mathsf{NP} \subseteq \mathsf{P}$. Falls $\mathsf{P} = \mathsf{NP}$, gilt die Aussage. Falls $\mathsf{P} \neq \mathsf{NP}$, gilt sie nicht, da offensichtlich $\mathsf{P} \subseteq \mathsf{NP}$ gilt und somit $\mathsf{NP} \not\subseteq \mathsf{P}$ gelten muss.
Die Frage ob $\mathsf{P} \overset{?}{=} \mathsf{NP}$ ist aber seit langem offen, wie auch in der Vorlesung besprochen.

(b) Im Gegensatz zur vorherigen Aussage ist diese definitiv falsch. Es gibt NP-schwere Probleme, die nicht in NP liegen. Diese liegen folglich auch definitiv nicht in P.
NP-schwer ist eine untere Schranke – das Problem ist mindestens so schwer wie jedes Problem in NP. Ist das Problem zusätzlich in NP, dann ist das Problem NP-vollständig.

Die zwei Aussagen lassen sich mit folgeden Diagramm verdeutlichen ("NP-hard" entspricht "NP-schwer"):



Quelle Wikipedia

AUFGABE 11.3. (*Ich hätt dann mal gern den Publikumsjoker*)

Nehmen Sie an, dass $\mathbf{P} \neq \mathbf{NP}$.

Stufe B

(a) Wenn $\mathsf{A} \leq_p$ 2-KNF-SAT, dann ist A **NP**-vollständig.

(b) Sei $\mathsf{B} \in \mathbf{P}$ und $\mathsf{A} \leq \mathsf{B}$. Dann ist $\mathsf{A} \in \mathbf{P}$.

(c) Sei $\overline{\mathsf{A}} \in \mathbf{P}$. Dann ist $\mathsf{A} \in \mathbf{NP}$. (Hinweis: $\overline{\mathsf{A}}$ ist das Komplement von A.)

(d) Es gilt: ($\mathsf{A} \leq_p$ SAT und MÜ $\leq_p \mathsf{A}$) gdw. A **NP**-vollständig.

(e) Gegeben sei ein gewichteter Graph G. Folgendes Problem ist **NP**-schwer:
`INPUT:` Eine natürliche Zahl k. `OUTPUT:` 1, falls in G eine Rundreise der Länge $\leq k$ existiert, die alle Knoten genau einmal besucht, sonst 0.

Lösungsskizze

(a) Falsch. Sei $\mathsf{A} = \text{2-KNF-SAT}$, dann gibt es die Reduktion trivialerweise (Identität), aber 2-KNF-SAT $\in \mathbf{P}$, deshalb kann 2-KNF-SAT unter der Annahme $\mathbf{P} \neq \mathbf{NP}$ nicht **NP**-schwer sein, und folglich auch nicht **NP**-vollständig.

(b) Falsch. $\leq$ sagt nichts über die Komplexität der Reduktion aus ($\leq_p$ hingegen schon). Wähle z.B. $\mathsf{A} = \text{3-KNF-SAT}$ und $\mathsf{B} = \{1\}$. Eine korrekte Reduktion kann zuerst berechnen, ob eine gegebene Formel erfüllbar ist und dann entsprechend 0 oder 1 ausgeben.

(c) Wahr. Da $\overline{A} \in \mathbf{P}$, kann $a \in \overline{A}$ als auch $a \notin \overline{A}$ in polynomieller Zeit entschieden werden. Ersteres ist äquivalent zu $a \notin A$ und letzteres zu $a \in A$. Somit kann auch A in polynomieller Zeit entschieden werden. Da weiters $\mathbf{P} \subseteq \mathbf{NP}$ gilt, folgt $A \in \mathbf{NP}$.

(d) Wahr.
$\Longrightarrow$: Mit $A \leq_p \text{SAT}$ folgt $A \in \mathsf{NP}$. Nach Vorlesung ist MÜ NP-vollständig. Somit folgt aus MÜ $\leq_p \mathsf{A}$, dass A NP-vollständig ist (Lemma 6.20).
$\Longleftarrow$: Wir wissen, dass SAT NP-vollständig ist. Mit $\mathsf{A} \in \mathsf{NP}$ folgt damit $\mathsf{A} \leq_p \text{SAT}$. Außerdem wissen wir, dass MÜ $\in \mathsf{NP}$ nach Vorlesung. Da A NP-vollständig ist nach Annahme, folgt somit MÜ $\leq_p \mathsf{A}$.

(e) Falsch. Der Graph ist fixiert. Wir können für G nun vorab bestimmen, ob eine Rundreise existiert und falls ja, die Länge der kürztesten Rundreise bestimmen. Wir bauen dann eine Funktion, die bei Eingabe k die Ausgabe 1 erzeugt genau dann wenn k kleiner als die bestimmte Länge ist und 0 sonst Die Funktion terminiert in linearer Zeit.

AUFGABE 11.4. (*Stundenplan*) Stufe C

In dieser Aufgabe betrachen wir das Stundenplan-Problem:

Gegeben: Endliche Mengen S (Studierende), V (Vorlesungen) und T (Termine) und eine Relation $R \subseteq S \times V$. Dabei bedeutet $(s, v) \in R$, dass s die Vorlesung v besuchen möchte.

Problem: Gibt es eine Abbildung $f : V \to T$, so dass alle Studierenden einen überschneidungsfreien Stundenplan haben, also

$$(s, v_1) \in R \wedge (s, v_2) \in R \wedge v_1 \neq v_2 \implies f(v_1) \neq f(v_2) \,.$$

(a) Zeigen Sie, dass Stundenplan in NP liegt.

(b) Zeigen Sie, dass Stundenplan NP-schwer ist, indem sie eine polynomielle Reduktion von 3col auf Stundenplan angeben.

Lösungsskizze

(a) Um zu zeigen, dass Stundenplan in NP liegt, genügt es zu zeigen, dass es effizient überprüfbare Zertifikate gibt (siehe Satz 6.9.). Eine solches Zertifikat ist offensichtlich die gesuchte Abbildung f, die man z.B. als geordnete Liste von Terminen (einen pro Vorlesung) kodieren kann. Wir müssen uns lediglich klarmachen, dass dieses Zertifikat in Polynomialzeit überprüft werden kann. Das ist der Fall, denn man kann z.B. zur Überprüfung die Stundenpläne aller Studierenden ($|S|$ Stundenpläne mit maximal $|V|$ Einträgen) aufstellen und auf Überschneidungen prüfen.

(b) Für einen gegebenen Graph $G = (V, E)$ konstruieren wir ein Stundenplanproblem (S, V', T, R) wie folgt:

$$\begin{aligned} S &= E \\ V' &= V \\ T &= \{1, 2, 3\} \\ R &= \{(\{v_1, v_2\}, v) \in S \times V \mid v \in \{v_1, v_2\}\} \end{aligned}$$

Die Knoten des Graphen werden also zu Vorlesungen und die drei Farben aus dem Färbbarkeitsproblem zu Terminen, die man den Vorlesungen zuordnet. Für jede Kante zwischen zwei Knoten erzeugt man nun einen Studenten, der genau die beiden entprechenden Vorlesungen hören will und somit verhindert, dass sie auf denselben Termin gelegt werden.
Die oben beschriebene Transformation ist in Polynomialzeit berechenbar (es handelt sich im Wesentlichen um eine "Umbenennung" der Konzepte). Fehlt noch der Korrektheitsbeweis:
$\Longrightarrow$: Sei $G \in$ 3col. Dann gibt es eine Färbung $c : V \to \{1, 2, 3\}$ mit $c(v) \neq c(w)$ für alle $\{v, w\} \in E$ mit $v \neq w$. Falls $(s, v_1) \in R \wedge (s, v_2) \in R \wedge v_1 \neq v_2$ gilt, folgt nach Definition von R, dass $s = \{v_1, v_2\} \in E$ und $v_1 \neq v_2$. Somit gilt $c(v_1) \neq c(v_2)$, d.h. c ist eine valide Zeitzuordnung für die Stundenplan-Instanz.
$\Longleftarrow$: Angenommen, es gibt eine valide Zeitzuordnung f für die Stundenplan-Instanz korrespondierend zu den Graphen G. Sei $\{v, w\} \in E$ mit $v \neq w$. Dann gilt per Definition von R, dass $(\{v, w\}, v) \in R \wedge (\{v, w\}, w) \in R \wedge v \neq w$. Da f eine valide Zeitzuordnung ist, folgt dann $f(v) \neq f(w)$ und weiters $f(v), f(w) \in \{1, 2, 3\}$ per Definition von T. Somit ist f eine valide 3-Färbung von G.
Somit ist das Graphenfärbungsproblem auf das Stundenplanproblem polynomiell reduzierbar:

$$\text{3col} \leq_p \text{Stundenplan}$$

Da von 3COL bekannt ist, dass es NP-schwer ist (wurde in der Vorlesung aber nicht bewiesen), haben wir nun gezeigt, dass STUNDENPLAN NP-schwer ist, und somit (mit (a)) NP-vollständig.

AUFGABE 11.5. (*Wizard of ZOLP*) Stufe D

Zeigen Sie, dass das Entscheidungsproblem *Zero-One-Linear-Program (ZOLP)* NP-schwer ist. Geben Sie hierzu eine geeignete Reduktion von 3-KNF-SAT auf ZOLP an und beschreiben Sie **zusätzlich** das Vorgehen anhand folgender Formel:

$$(\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_3 \vee x_4)$$

Den Korrektheitsbeweis müssen Sie nicht ausformulieren.

Definition (ZOLP-Entscheidungsproblem)

Eingabe: Ein System von linearen Ungleichungen

$$\begin{aligned} b_1 &\leq a_{1,1}y_1 + \ldots + a_{1,n}y_n \\ b_2 &\leq a_{2,1}y_1 + \ldots + a_{2,n}y_n \\ &\vdots \\ b_m &\leq a_{m,1}y_1 + \ldots + a_{m,n}y_n \end{aligned}$$

mit $a_{i,j}, b_i \in \mathbb{Z}$ und $m, n > 0$.
Frage: Gibt es für die Variablen $y_1, \ldots, y_n$ Werte aus $\{0, 1\}$, sodass alle Ungleichungen gleichzeitig erfüllt sind?

Lösungsskizze
Möglichkeit 1: Für jede Variable im Ausgangsproblem zwei Variablen im Zielproblem. Sei $F := \bigwedge_{i=1}^{k} C_i$ und $C_i := \bigvee_{j=1}^{3} L_{i,j}$ mit z Variablen die Eingabe für das 3KNF-SAT Problem. Die Reduktion f modelliert jede Variable des SAT-Problems mit zwei Variablen des ZOLP-Problems. Sei x_i eine Variable dann encodiert y_{2i}, ob das Literal x_i wahr ist, und y_{2i-1}, ob das Literal $\neg x_i$ wahr ist. Somit haben wir

$$f(x_i) \mapsto y_{2i} \qquad f(\neg x_i) \mapsto y_{2i-1}$$

Für das ZOLP wählen wir $m = k + 2z$ und $n = 2z$. Die Zeile $i \in [1, k]$ ist dann definiert in Abhängigkeit von C_i als

$$f(L_{i,1} \vee L_{i,2} \vee L_{i,3}) \qquad \mapsto \qquad 1 \leq f(L_{i,1}) + f(L_{i,2}) + f(L_{i,3})$$

Zusätzlich fügen wir folgende Konsistenzgleichungen ein, damit y_{2i} und $y_{2i} - 1$ unterschiedliche Werte erhalten für alle $i \in [1, z]$

$$\begin{aligned} 1 &\leq y_{2i} + y_{2i-1} \\ -1 &\leq -y_{2i} - y_{2i-1} \end{aligned}$$

Die Reduktion ist polynomiell in der Eingabegröße, da für jede Klausel eine Ungleichung und für jede Variable 2 Ungleichungen erstellt werden.
Anwendung auf Beispiel:

$$\begin{aligned} 1 &\leq y_1 + y_4 + y_6 \\ 1 &\leq y_2 + y_3 + y_8 \\ 1 &\leq y_1 + y_2 \\ -1 &\leq -y_1 - y_2 \\ 1 &\leq y_3 + y_4 \\ -1 &\leq -y_3 - y_4 \\ 1 &\leq y_5 + y_6 \\ -1 &\leq -y_5 - y_6 \\ 1 &\leq y_7 + y_8 \\ -1 &\leq -y_7 - y_8 \end{aligned}$$

Möglichkeit 2: Für jede Variable im Ausgangsproblem eine Variable im Zielproblem. Skizze: Wir kodieren jede Klausel als eine Ungleichung im Zielproblem. Disjunktionen ersetzen wir mit Addition. Literale x_i werden in einen Summanden y_i übersetzt und Literale $\neg x_i$ in einen Summanden $1 - y_i$. Damit ist ein Literal l erfüllt genau dann wenn der entsprechende Summand den Wert 1 annimmt. Um sicherzustellen, dass jede Klausel erfüllt ist, steht auf der linken Seite jeder Ungleichung 1. Offensichtlich kann eine jede Ungleichung dieser Form in die von ZOLP geforderte Form gebracht werden.

$$(\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_3 \vee x_4)$$

Anwendung auf Beispiel:

$$\begin{aligned} 1 &\leq (1 - y_1) + y_2 + y_3 \\ 1 &\leq y_1 + (1 - y_3) + y_4 \end{aligned}$$

Umformen:

$$\begin{aligned} 0 &\leq -y_1 + y_2 + y_3 \\ 0 &\leq y_1 - y_3 + y_4 \end{aligned}$$