Lösungsstrategien für NP-schwere Probleme
Aufgabenblatt 01

Jakob Rieck 6423721

Konstantin Kobs 6414943

Thomas Maier 6319878

Tom Petersen 6359640

Abgabe zum 18.04.16

Aufgabe 1

a) Zunächst formulieren wir das 4D-Matching-Problem:
Eingabe: Disjunkte Mengen A, B, C und D mit $|A| = |B| = |C| = |D| = n$ sowie eine Menge $T \subseteq A \times B \times C \times D$ von Tripeln.
Frage: Gibt es eine Menge von n Tripeln in T, sodass jedes Element aus $A \cup B \cup C \cup D$ in genau einem dieser Tripel vorkommt?

Nun zeigen wir, dass 4D-Matching in NP liegt. Dazu muss gezeigt werden, dass ein perfektes Matching in polynomieller Zeit verifiziert werden kann. Dies ist möglich, indem geprüft wird, ob jedes Element genau einmal in der Tupelmenge auftaucht.

Nun zeigen wir, dass das 4D-Matching-Problem NP-vollständig ist, also dass 3D-Matching $\leq_p$ 4D-Matching gilt. Wir reduzieren dazu das 3D-Matching-Problem auf das 4D-Matching-Problem. Wir zeigen nun eine Konstruktion, welche ein korrektes Matching in 3D auf ein korrektes Matching in 4D, sowie ein falsches Matching in 3D auf ein falsches Matching in 4D abbildet. Dazu verbinden wir die Mengen C und D mithilfe einer bijektiven Abbildung. Man könnte zum Beispiel die Elemente der beiden Mengen jeweils nach einem bestimmten Kriterium sortieren und dann die jeweils i-ten Elemente in der Abbildung verbinden (für $i \in \{1, \dots, n\}$). Gegeben sei nun ein perfektes 3D-Matching für die Mengen A, B und C. Alle Elemente der genannten Mengen sind also in genau einem Tupel enthalten. Nach unserer Konstruktion eines 4D-Matchings ist wegen der Bijektivität in dem Matching genau einmal jedes Element der Menge D vorhanden. Ein falsches Matching in 3D trifft entweder nicht alle Elemente der Mengen A oder B, oder nicht alle Elemente aus der Menge C und damit auch der Menge D. Somit ist gezeigt, dass sich das 3D-Matching-Problem auf das 4D-Matching-Problem reduzieren lässt, was bedeutet, dass 4D-Matching NP-vollständig ist.

b) k-CLIQUE ist für jedes $2 < k < |V|$ NP-vollständig.

Beweis: Zunächst zeigen wir wieder, dass das Problem in NP liegt. Ist eine k-elementige Menge von Knoten als Lösung gegeben, so kann einfach überprüft werden, ob zwischen jedem der gegebenen Knoten eine Kante existiert. Somit kann eine Lösung in polynomieller Zeit verifiziert werden. Nun wird eine Reduzierung vorgenommen. Hierzu kann k-Clique wie folgt auf Independent-Set reduziert werden. Zu jedem Graphen $G = (V, E)$ wird der komplementäre Graph $G^* = (V^*, E^*)$ erzeugt, wobei $V^* = V$ und $e \in E^*$ genau dann, wenn $e \notin E$ gilt. Anschließend wird auf dem Graphen G^* Independent Set ausgeführt. Wenn nun eine unabhängige Menge der Größe k in G^* gefunden wird, muss es auch einen vollständigen Graphen der Größe k in G geben. Für $k = |V|$ wird geprüft, ob der Graph G vollständig ist. Dies ist jedoch in P möglich. Da nicht bekannt ist ob $P = NP$ gilt kann keine Aussage darüber getroffen werden, ob k-Clique auch für $k = |V|$ NP-vollständig ist. Ähnliches gilt für $k = 2$. In diesem Fall wird geprüft, ob der Graph G eine Kante enthält. Für $k = 1$ ist die Antwort trivialerweise immer ja, denn jeder einzelne Knoten ist ein vollständiger Subgraph von G.

Aufgabe 2

a) Um eine Menge H in polynomieller Zeit als Hitting Set zu verifizieren, wird H als Zertifikat verwendet. Zunächst wird geprüft ob $|H| \leq k$ gilt. Als nächstes wird geprüft, ob $H \subseteq A$ gilt und zum Schluss wird geprüft, ob $H \cap B_i \neq \emptyset$ für jedes $i \in 1, \ldots, m$ gilt. Diese drei Prüfungen liegen in polynomieller Laufzeit. Somit liegt Hitting Set in NP.

Im Folgenden wird nun die Reduktion 3-SAT $\leq_p$ HITTING SET beschrieben. Sei Φ eine Instanz von 3-SAT mit den Klauseln $C_1, \ldots, C_m$ bestehend aus den Literalen $x_1, \ldots, x_n$ bzw. deren negierten Formen.

Diese Instanz lässt sich nun folgendermaßen auf HITTING SET reduzieren: Die Grundmenge $A = \{x_1, \overline{x_1}, \ldots, x_n, \overline{x_n}\}$ besteht aus allen möglichen Literalen in positiver und negativer Form. Für jede Klausel $C_i = (x_{i1} \vee x_{i2} \vee$

x_{i3}) wird eine Menge $B_i = \{x_{i1}, x_{i2}, x_{i3}\}$ hinzugefügt (mit x_i seien hier sowohl positive als auch negative Literale beschrieben). Zusätzlich wird für jedes Literal x_j eine Menge $B_j = \{x_j, \overline{x_j}\}$ hinzugefügt. Für $k = \frac{|A|}{2}$ ergibt sich so eine Instanz von HITTING SET, für die gilt: Die konstruierte Instanz besitzt ein HITTING SET genau dann, wenn Φ erfüllbar ist.

Beweis "$\rightarrow$": Besteht ein HITTING SET für $k = \frac{|A|}{2}$, dann ist mindestens ein Element jeder Menge $B_i = \{x_{i1}, x_{i2}, x_{i3}\}$ in der Ergebnismenge H enthalten. Außerdem gilt dies ebenso für ein Element jeder Menge $B_j = \{x_j, \overline{x_j}\}$ (in diesem Fall sogar genau ein Element, da es k dieser Mengen gibt). Wird nun Φ so belegt, dass jedes $x_i \in H$ mit wahr belegt wird, alle anderen mit falsch, so handelt es sich um eine erfüllende Belegung für Φ. Nach der Konstruktion ist jede Klausel erfüllt, da mindestens eine ihrer Variablen zu wahr ausgewertet wird. Weiterhin kann keine Variable uneindeutig belegt sein, da - wie bereits erwähnt - genau ein Literal aus jedem $B_j = \{x_j, \overline{x_j}\}$ mit wahr belegt ist und sein Komplement entsprechend mit falsch.

Beweis "$\leftarrow$": Wenn Φ erfüllbar ist, so gibt es eine Belegung, die mindestens ein x_i jeder Klausel zu wahr auswertet. Außerdem ist jedes x_i entweder mit wahr oder falsch belegt. Wählt man nun in der entsprechend konstruierten HITTING SET-Instanz H als die Menge aller x_i, die zu wahr ausgewertet werden (wiederum seien hier auch negative Literale gemeint), so erhält man mit H ein HITTING SET, denn jede Menge B_j, die aus einer Klausel gebildet wurde, enthält mindestens ein x_i aus H, da Φ erfüllbar ist. Außerdem wird auch jede Menge $B_j = \{x_j, \overline{x_j}\}$ von genau einem Element aus H "getroffen", da entweder x_j oder $\overline{x_j}$ wahr ist. Da Φ eine vollständige Belegung für alle Literale (positive und negative) ist und H alle wahren Literale enthält, gilt außerdem $|H| = \frac{|A|}{2}$.

b) Wir zeigen: `VERTEX COVER` $\leq_p$ `HITTING SET` Gegeben sei eine Eingabe für `VERTEX COVER`, also $G = (V, E)$ und $k \geq 1$.

Wir konstruieren Mengen $B_1, ..., B_m$ als $B_i = \{u, v\} \forall e \in E$, d.h. für jede Kante des ursprünglichen Graphen gibt es genau eine Menge ($m = |E|$).

Zu zeigen ist nun: Lösung für `VERTEX COVER` $\Rightarrow$ Lösung für `HITTING SET`. Es liege eine Knotenüberdeckung S vor, d.h. $\forall (u, v) \in E$ gilt: $u \in V$ oder $v \in V$ und $|S| \leq k$. Da entweder u oder v in V vorkommt, gilt auch $S \cap B_i \neq \emptyset$, den für jede Kante ($\rightarrow$ jede Menge B_i) wird ein Element getroffen. S ist also auch gleichzeitig das gesuchte `HITTING SET` H mit $|S| = |H| \leq k$ Elementen.

Außerdem zu zeigen ist: Lösung für `HITTING SET` $\Rightarrow$ Lösung für `VERTEX COVER` Eine Lösung H mit $|H| \leq k$ ist eine Menge, für die gilt: $H \cap B_i \neq \emptyset$. Also gilt für jede Kante $(u, v) \in E : (u \in H)$ oder $(v \in H)$ (Schließlich waren die Mengen B_i nichts weiter als die 2 Knoten jeder Kante) Damit wird jede Kante aus E "getroffen", H ist also auch eine Knotenüberdeckung $\blacksquare$.