Karlsruher Institut für Technologie
Institut für Theoretische Informatik

Prof. Dr. Peter Sanders
L. Hübschle-Schneider, T. Maier

11. Übungsblatt zu Theoretische Grundlagen der Informatik im WS 2015/16

http://algo2.iti.kit.edu/TGI2015.php
{sanders,huebschle,t.maier}@kit.edu

Musterlösungen

Aufgabe 1 (*Reduktionen,* $2+4+4$ *Punkte*)

INDEPENDENT SET: Gegeben ein Graph $G = (V, E)$ und $k \in \mathbb{N}$, gibt es ein Independent Set der Größe k, also eine Menge $I \subseteq V$ mit $|I| \geq k$, sodass $(I \times I) \cap E = \emptyset$, d.h. für keine Kante des Graphen beide Endpunkte in I liegen?

CLIQUE: Gegeben ein Graph $G = (V, E)$ und $k \in \mathbb{N}$, gibt es eine Clique der Größe k, d.h. eine Menge $C \subseteq V$ mit $|C| \geq k$, sodass $C \times C \subseteq E$, d.h. jeder Knoten der Clique mit jedem anderen verbunden ist?

VERTEX COVER: Gegeben ein Graph $G = (V, E)$ und $k \in \mathbb{N}$, gibt es ein Vertex Cover der Größe k, d.h. eine Menge $C \subseteq V$ mit $|C| \leq k$, sodass für jede Kante mindestens einer der Endpunkte in C liegt?

EXACT COVER: Gegeben eine Grundmenge M und ein Teilmengensystem $T \subseteq \mathcal{P}(M)$, existiert eine *disjunkte* Überdeckung von M mit Hilfe von Teilmengen in T, d.h. $\exists_{U \subseteq T} : \dot{\bigcup}_{u \in U} = M$?

SUBSET SUM: Gegeben eine Zahl $W \in \mathbb{N}$ und eine Menge $M = \{v_1, \ldots, v_n\}$ von Gegenständen, habe Gegenstand v_i das Gewicht $c(v_i) \in \mathbb{N}$. Gibt es eine Teilmenge $U \subseteq M$ sodass $\sum_{v \in U} c(v) = W$?

Zeigen Sie unter Verwendung, dass INDEPENDENT SET und EXACT COVER NP-vollständig sind:

a) CLIQUE ist NP-vollständig ($\geq_p$ INDEPENDENT SET)

b) VERTEX COVER ist NP-vollständig ($\geq_p$ INDEPENDENT SET)

c) SUBSET SUM ist NP-vollständig ($\geq_p$ EXACT COVER)

Musterlösung:

a) CLIQUE liegt in NP, da für eine nichtdeterministisch gewählte Knotenmenge leicht geprüft werden kann, ob es sich um eine Clique handelt.

   Zwischen INDEPENDENT SET und CLIQUE existiert eine Dualität: Eine Menge M ist genau dann ein Independent Set, wenn es *keine* Kante zwischen ihren Knoten gibt, während sie genau dann eine Clique ist, wenn es *alle* Kanten zwischen ihren Knoten gibt. Wenn wir also $E' = (V \times V) \setminus E$ und $G' = (V, E')$ betrachten, ist M ein Independent Set in G genau dann wenn es eine Clique in G' ist!

   Für die Reduktion bilden wir also die INDEPENDENT SET-Instanz (G, k) auf die CLIQUE-Instanz (G', k) ab und sind fertig. Diese Reduktion ist offensichtlich polynomiell, genauer: $\mathcal{O}(|V|^2)$.

b) VERTEX COVER liegt in NP, da für eine nichtdeterministisch gewählte Knotenmenge leicht geprüft werden kann, ob es sich um ein Vertex Cover handelt.

   Wir beobachten: Wenn in einem Graphen $G = (V, E)$ die Menge I ein Independent Set ist, dann ist $V \setminus I$ ein Vertex Cover. Diese Äquivalenz ist leicht zu beweisen: Für jede Kante $e = (u, v) \in E$ gilt $\neg(u \in I \wedge v \in I) \iff (u \notin I) \vee (v \notin I) \iff (u \in V \setminus I) \vee (v \in V \setminus I)$

Daher ist I genau dann ein Independent Set in G wenn $V \setminus I$ ein Vertex Cover ist. Die Reduktion muss für $G = (V, E)$ also nur die INDEPENDENT SET-Instanz (G, k) auf die VERTEX COVER-Instanz $(G, |V| - k)$ abbilden. Dies ist offensichtlich in $\mathcal{O}(1)$ möglich.

c) SUBSET SUM liegt in NP, da die Summe einer nichtdeterministisch gewählten Teilmenge leicht berechnet werden kann (es genügt diese mit W zu vergleichen.

Reduktion von EXACT COVER auf SUBSET SUM:

Gegeben: $M_{EC} = \{v_1, \dots, v_n\}$ und $T_{EC} = \{t_1, \dots, t_m\} \quad t_i \subseteq M$

Jede Teilmenge kann als Bitvektor der Länge n ($v_j \in t_i \Leftrightarrow \texttt{Bit}(t_i)[j] = 1$) dargestellt werden. Fasst man diesen Bitvektor als Binärzahl auf, so korrespondiert eine vollständige disjunkte Abdeckung der Grundmenge M_{EC} mit einer Summe vom Wert $2^n - 1$. Da jedes Element in genau einer Menge enthalten ist, ist jedes Bit der Summe gesetzt und es kann keinen Überlauf geben, weil die Vereinigung aus disjunkten Teilmengen besteht. Leider ist die Rückrichtung im Allgemeinen nicht korrekt.

Betrachtet man zum Beispiel die Bitvektoren `1001 + 0101 + 0001 = 1111` so ergibt sich das korrekte Ergebnis durch einen Überlauf. D.h. wir müssen verhindern, dass ein Überlauf zu der gewünschten Summe führt.

Hierzu gibt es zwei einfache Optionen:

1. Füge Nullbits als Überlaufspuffer zwischen den Bits der Bitvektoren ein. Jede einzelne Stelle des Bitvektors kann maximal m mal überlaufen. Dementsprechend genügen $\lceil \log m \rceil = k$ Stellen Puffer zwischen je 2 Bits.
2. Betrachten wir die Bitvektoren als Zahlen zur Basis m, so kann kein "echter" Überlauf entstehen.

Endgültige Reduktion: Aus der EXACT COVER-Instanz M_{EC}, T_{EC} konstruieren wir die SUBSET SUM-Instanz $M_{SUB} = \{u_1, \dots, u_{|T_{EX}|}\}$ mit $c(u_i) = \texttt{Bit*}(t_i)$ und der zugehörigen Schranke $k = \texttt{Bit*}(M_{EC})$ (dabei ist Bit* die oben beschriebene Bitvektorfunktion mit einer der Überlaufsanpassungen).

Diese Reduktion hat eine Komplexität von $\mathcal{O}(n \cdot m)$ und ist damit offensichtlich polynomiell in der Eingabegröße.

Aufgabe 2 (*SAT-Formulierungen,* $3 + 4$ *Punkte*)

Geben Sie für die folgenden Probleme an, wie deren Instanzen als Instanzen des Erfüllbarkeitsproblems SAT ausgedrückt werden können. Geben Sie dazu eine Vorschrift an, wie die Eingabeinstanz in eine Menge von Klauseln überführt werden kann und wie diese Klauseln miteinander verknüpft werden (Konjunktion oder Disjunktion, Schachtelungen sind erlaubt). Beschreiben Sie die Bedeutung der Variablen und Klauseln. Die Anzahl der Variablen und Klauseln muss polynomiell in den Parametern der Eingabeinstanz sein. *Terminologie:* Eine Klausel ist eine Konjunktion oder Disjunktion von Literalen; ein Literal ist eine Variable (z.B. a) oder deren Negation ($\neg a$).

a) 3COLOUR: Gegeben ein ungerichteter Graph $G = (V, E)$, lassen sich die Knoten des Graphen so einfärben, dass keine zwei benachbarten Knoten die selbe Farbe haben? Formell:
$\exists_{c\colon V \to \{R,G,B\}}\colon c(u) = c(v) \to (u, v) \notin E$

b) EXACT COVER, Definition siehe Aufgabe 1

Musterlösung:

a) Zur Modellierung verwenden wir für jede Kombination aus einem Knoten und einer Farbe eine aussagenlogische Variable.

$\forall_{(v,C) \in V \times \{R,G,B\}}\colon x_{v,C}$ *kodiert, ob Knoten* v *die Farbe* C *hat*

Es gibt drei Eigenschaften, die erhalten werden müssen:

- Jeder Knoten hat eine Farbe:

$$\bigwedge_{v \in V} (x_{v,R} \vee x_{v,G} \vee x_{v,B})$$

- Kein Knoten hat mehr als eine Farbe:

$$\bigwedge_{v \in V} \begin{pmatrix} (x_{v,R} \to (\neg x_{v,G} \wedge \neg x_{v,B})) \wedge \\ (x_{v,G} \to (\neg x_{v,R} \wedge \neg x_{v,B})) \wedge \\ (x_{v,B} \to (\neg x_{v,R} \wedge \neg x_{v,G})) \end{pmatrix} = \bigwedge_{v \in V} \begin{pmatrix} (\neg x_{v,R} \vee (\neg x_{v,G} \wedge \neg x_{v,B})) \wedge \\ (\neg x_{v,G} \vee (\neg x_{v,R} \wedge \neg x_{v,B})) \wedge \\ (\neg x_{v,B} \vee (\neg x_{v,R} \wedge \neg x_{v,G})) \end{pmatrix}$$

- Keine zwei benachbarten Knoten haben die selbe Farbe:

$$\bigwedge_{e=\{v,u\} \in E} \begin{pmatrix} \neg(x_{v,R} \wedge x_{u,R}) \wedge \\ \neg(x_{v,R} \wedge x_{u,R}) \wedge \\ \neg(x_{v,R} \wedge x_{u,R}) \end{pmatrix}$$

Sowohl die Größe als auch die Konstruktion der entstandenen Instanz liegt in $\mathcal{O}(|V| + |E|)$ und ist damit polynomiell in der Eingabegröße beschränkt.

b) Sei $A = \{a_1, a_2, \ldots, a_n\}$ die Menge und $M = \{M_1, M_2, \ldots, M_m\}$ das Mengensystem. Wir definieren für jede Kombination aus einem Element $a_i \in A$ und $M_j \in M$ eine Variable $v_{i,j}$, die besagt, dass Element a_i durch M_j in die Lösung gelangt ist. Ferner definieren wir für jedes $M_j \in M$ eine Variable m_j, die besagt, ob M_j im Exact Cover enthalten ist. Damit formulieren wir nun die folgenden (Kombinationen von) Klauseln.

Erstens kann eine Menge nur ganz oder garnicht zur Lösung gehören, und eine Menge kann nur ihre Elemente zur Lösung beitragen:

$$\bigwedge_{j=1,\ldots,m} \left(\left(m_j \wedge \bigwedge_{i \in M_j} v_{i,j} \wedge \bigwedge_{i \notin M_j} \neg v_{i,j} \right) \vee \left(\neg m_j \wedge \bigwedge_{i=1,\ldots,n} \neg v_{i,j} \right) \right) \tag{1}$$

Zweitens darf jedes a_i nur zu einer Menge der Lösung gehören:

$$\bigwedge_{i=1,\ldots,n} \bigwedge_{j \neq k \in \{1,\ldots,m\}} \neg v_{i,j} \vee \neg v_{i,k} \tag{2}$$

Umgekehrt muss jedes a_i aber auch zu einer Menge der Lösung gehören:

$$\bigwedge_{i=1,\ldots,n} \bigvee_{j=1,\ldots,m} v_{i,j} \tag{3}$$

Diese drei Konstrukte müssen gleichzeitig erfüllt sein, werden also mit einer Konjunktion verbunden: (1) $\wedge$ (2) $\wedge$ (3). Es bleibt noch zu zeigen, dass die Instanz nur polynomiell groß ist. Die Anzahl Variablen ist $(n+1) \cdot m$, was offensichtlich polynomiell ist. Teil (1) trägt $\mathcal{O}(m)$ Klauseln der Größe $\mathcal{O}(n)$ bei. Teil (2) trägt $\mathcal{O}(n \cdot m^2)$ Klauseln mit je zwei Literalen bei. Teil (3) letztlich trägt n Klauseln der Größe m bei. Alle Komponenten sind polynomiell, daher also auch die SAT-Instanz.

Aufgabe 3 (*3COLOUR, 3 Punkte*)

Konstruieren Sie mit dem aus der Übung bekannten Verfahren eine 3COLOUR-Instanz, die äquivalent ist zu der folgenden 3SAT-Instanz. Färben Sie außerdem den entstandenen Graphen in drei Farben und lesen sie eine korrekte Belegung ab.

$(a \vee \neg b \vee c) \wedge (\neg a \vee b \vee c)$

Musterlösung: