ETH zürich
Departement Informatik

Algorithmik für schwere Probleme

Prof. Dr. Dennis Komm
Dr. Hans-Joachim Böckenhauer
`https://courses.ite.inf.ethz.ch/schwere_prob_23/`

Lösungsvorschläge – Blatt 4

Zürich, 21. März 2023

Lösung zu Aufgabe 4

(a) Sei α eine beliebige Belegung von Φ und sei $\overline{\alpha} = 1 - \alpha$. Dann erfüllt entweder α oder $\overline{\alpha}$ mindestens die Hälfte aller m Klauseln, also mindestens $\lceil m/2 \rceil \geq m/2 \geq k$ Klauseln. (Diese Aussage wurde in leicht anderer Formulierung bereits in Aufgabe 3 von Blatt 3 bewiesen.)

(b) Ein X saturierendes Matching ordnet jeder Variablen $x \in X$ eine andere Klausel von Φ zu, die entweder x oder $\overline{x}$ enthält. In ersterem Fall setzen wir $\alpha(x) = 1$, in letzterem $\alpha(x) = 0$. Die Belegung α erfüllt dann mindestens $|X|$ Klauseln.

(c) Wir nehmen an, dass $G_\Phi = (V, E)$ kein X saturierendes Matching hat. Gemäss dem Satz vom Aufgabenblatt finden wir in Polynomzeit eine Teilmenge $C \subseteq X$ mit $|N(C)| < |C|$ (und somit $C \neq \emptyset$), so dass für jede echte Teilmenge $T \subsetneq C$ gilt, dass $|N(T)| \geq |T|$. Sei $H = N(C)$ die offene Nachbarschaft von C, also die Menge aller Knoten von G_Φ, die zu einem Knoten von C adjazent sind, minus die Knoten von C. Sei schliesslich $B = V \setminus (C \cup H)$ die Menge der verbleibenden Knoten. Wir zeigen, dass (C, H, B) eine Kronenzerlegung ist.

Die Menge C ist als Teilmenge des Independent Sets X ebenfalls ein Independent Set. Da H die gesamte Nachbarschaft von C ist, separiert H die beiden Mengen C und $B = V \setminus (C \cup N(C))$, das heisst, es gibt keine Kanten zwischen C und B. Es bleibt zu zeigen, dass ein Matching zwischen H und C existiert, das H saturiert.

Sei $x \in C$. Sei T eine beliebige Teilmenge von $C \setminus \{x\}$. Da T eine echte Teilmenge von C ist, gilt nach der Minimalitätsbedingung an C, dass $|N(T)| \geq |T|$. Nach dem Satz vom Aufgabenblatt finden wir also in Polynomzeit ein Matching M zwischen $C \setminus \{x\}$ und H, das $C \setminus \{x\}$ saturiert. Dieses Matching M besteht also aus $|C \setminus \{x\}| = |C| - 1$ Kanten, entsprechend gilt $|H| \geq |C| - 1$. Die umgekehrte Umgleichung $|H| \leq |C| - 1$ gilt ebenfalls, da $|H| = |N(C)| < |C|$. Daher gilt $|M| = |H|$, das Matching M zwischen $C \setminus \{x\}$ und H saturiert also H. Somit ist M auch ein H saturierendes Matching zwischen C und H.

(d) Sei (C, H, B) eine Kronenzerlegung von G_Φ. Dann existiert ein H saturierendes Matching M zwischen C und H. Zum Widerspruch nehmen wir an, es existiere eine Belegung α von Φ, die einerseits die Anzahl erfüllter Klauseln maximiert und andererseits nicht alle Klauseln erfüllt, die den Knoten von H entsprechen. Sei α' die Belegung, die aus α entsteht, indem wir für jede Klausel, die einem Knoten in H entspricht, die in M benachbarte Variable so neu belegen, dass die Klausel erfüllt ist. Da zwischen C und B keine Kanten existieren, erfüllt α' eine Klausel, die einem Knoten aus B entspricht, genau dann, wenn auch α das tut. Somit erfüllt α' mindestens eine Klausel mehr als α, im Widerspruch zu unserer Annahme für α.

(e) Wir beschreiben eine Reduktionsregel. Sei (Φ, k) gegeben. Falls $m \geq 2k$, können wir nach Teilaufgabe (a) eine triviale Ja-Instanz ausgeben. Falls $n < k$, geben wir (Φ, k) unverändert als Kern aus. Nehmen wir also an, dass $n \geq k$ und $m < 2k$. Wir benützen den Polynomzeitalgorithmus aus dem Satz vom Aufgabenblatt, um herauszufinden, ob der Inzidenzgraph G_Φ ein X saturierendes Matching hat. Falls ja, können wir gemäss Teilaufgabe (b) eine triviale Ja-Instanz ausgeben. Ansonsten berechnen wir eine Kronenzerlegung (C, H, B) von G_Φ mit $C \subseteq X$ und $H \subseteq Y$. Sei Φ' die Formel, die aus Φ entsteht, wenn wir alle Klauseln löschen, die von Knoten in H repräsentiert werden. Sei zudem $k' = k - |H|$. Das Resultat der Reduktionsregel ist in diesem Fall (Φ', k').

Die beschriebene Reduktionsregel ist wegen (c) in Polynomzeit durchführbar und wegen Teilaufgabe (d) korrekt. Die Anwendung dieser Reduktionsregel verringert sowohl die Anzahl der Variablen als auch die Anzahl Klauseln strikt, denn C ist nicht leer und H ebenso, weil jede Variable in mindestens einer Klausel vorkommt. Durch wiederholtes Anwenden der Regel erhalten wir einen Kern der gewünschten Grösse.