Schwere Probleme und die Kunst der Reduktion

Institut für Theoretische Informatik

Sommersemester 20??

Inhaltsverzeichnis

Kapitel 1

List Coloring

Thomas Bläsius

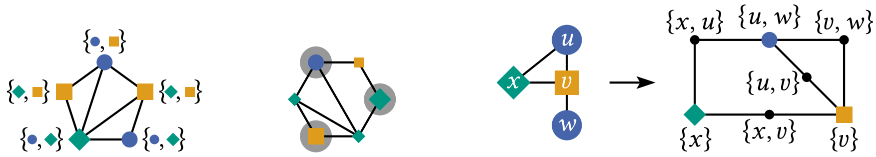
Bei dem Problem List Coloring besteht die Eingabe aus einem Graphen, sowie einer Liste von möglichen Farben für jeden Knoten. Ziel ist es, für jeden Knoten eine Farbe aus seiner Liste zu wählen, sodass je zwei benachbarte Knoten unterschiedliche Farben haben; siehe Abbildung 1.1a für eine Instanz mit Lösung. Das Problem ist offensichtlich NP-schwer, denn erlaubt man für jeden Knoten dieselben drei Farben (z.B. rot, blau und grün), so erhält man das bekanntermaßen NP-schwere Problem 3-Coloring [3].

Eine naheliegende Frage ist nun, ob man das Problem eventuell effizient lösen kann, wenn man nur Graphen betrachtet, die eine einfache Struktur haben. Wie einfach die Struktur eines Graphen ist, wollen wir in dieser Seminararbeit darüber messen, wie viele Knoten man löschen muss, bis keine Kanten mehr übrig sind. Sei k diese Anzahl zu löschender Knoten und sei außerdem n die Anzahl der Knoten des Graphen. Man könnte jetzt hoffen einen Algorithmus zu finden, der beispielsweise die Laufzeit $2^k \cdot n^{O(1)}$ hat. Das wäre zwar exponentiell in dem Parameter k aber polynomiell in n; was ggf. akzeptabel ist, wenn k sehr klein ist.

In dieser Arbeit werden wir sehen, dass diese Hoffnung vermutlich unbegründet ist, was wir mittels einer Reduktion von dem Problem Multicolored Independent Set zeigen. Die Reduktion ist der Arbeit von Fellows, Fomin, Lokshtanov, Rosamond, Saurabh, Szeider und Thomassen [2] entnommen. In Abschnitt 1.1 führen wir zunächst einige grundlegende Definitionen ein, bevor wir in Abschnitt 1.2, dem Hauptteil dieser Arbeit, die Reduktion zeigen. In Abschnitt 1.3 werden wir ein Fazit ziehen und insbesondere kurz diskutieren, inwiefern diese Reduktion einen Algorithmus mit Laufzeit $2^k \cdot n^{O(1)}$ ausschließt (unter gängigen komplexitätstheoretischen Annahmen).

1.1 Grundlegende Definitionen

Sei $G = (V, E)$ ein Graph mit Knotenmenge V und Kantenmenge $E \subseteq \binom{V}{2}$. Für eine Kante $\{u, v\} \in E$ schreiben wir auch knapper uv und nennen u und v *benachbart*. Für eine Menge C an *Farben*, nennen wir eine Abbildung $\mathrm{col}\colon V \to C$, die jedem Knoten eine Farbe zuordnet, eine *Färbung* des Graphen G mit $|C|$ Farben. Eine Färbung heißt *gültig*, wenn je zwei benachbarte Knoten unterschiedliche Farben haben, also wenn für alle $uv \in E$ gilt, dass $\mathrm{col}(u) \neq \mathrm{col}(v)$. Sei nun für jeden Knoten $v \in V$ eine Menge $C_v \subseteq C$ an *erlaubten Farben* gegeben. Wir nennen die Färbung col eine *Listenfärbung*, wenn sie jedem Knoten eine für ihn erlaubte Farbe zuweist, also wenn $\mathrm{col}(v) \in C_v$ für alle $v \in V$.



(a) LC **(b)** MIS **(c)** MIS → LC

Abbildung 1.1: Illustration der Probleme LIST COLORING (a), MULTICOLORED INDEPENDENT SET (b), sowie der Reduktion von Zweiterm auf Ersteres (c).

Das Problem LIST COLORING hat als Eingabe einen Graphen, sowie für jeden Knoten eine Menge erlaubter Farben und man muss entscheiden, ob eine gültige Listenfärbung existiert. Abbildung 1.1a zeigt eine gültige Listenfärbung.

Sei $G = (V, E)$ ein Graph. Wir nennen eine Knotenmenge $S \subseteq V$ eine *unabhängige Menge*, falls es zwischen Knoten aus S keine Kanten gibt, also falls $|e \cap S| \leq 1$ für alle $e \in E$. Sei nun zu dem Graphen $G = (V, E)$ auch noch eine (nicht notwendigerweise gültige) Färbung gegeben. Wir nennen eine Knotenmenge $S \subseteq V$ *bunt*, wenn alle Knoten in S unterschiedliche Farben haben, also wenn für alle $u, v \in S$ mit $u \neq v$ gilt, dass $\mathrm{col}(u) \neq \mathrm{col}(v)$. Das Problem MULTICOLORED INDEPENDENT SET fragt für einen mit k Farben gefärbten Graphen, ob es eine bunte unabhängige Menge der Größe k gibt; siehe Abbildung 1.1b für eine Instanz mit markierter Lösung. Beachte, dass diese unabhängige Menge dann für jede Farbe genau einen Knoten mit dieser Farbe enthält.

1.2 Reduktion

Im Folgenden geben wir eine Reduktion von MULTICOLORED INDEPENDENT SET auf LIST COLORING an. Die Kernbeobachtung ist dabei die Folgende. MULTICOLORED INDEPENDENT SET kann man so auffassen, dass man pro Farbe einen Knoten auswählen muss, sodass das Resultat eine unabhängige Menge ist. Bei LIST COLORING muss man für jeden Knoten eine erlaubte Farbe auswählen, sodass das Resultat eine gültige Färbung ist. Damit klingt es naheliegend, dass wir von MULTICOLORED INDEPENDENT SET zu LIST COLORING übersetzen, indem wir Farben in Knoten und Knoten in Farben umwandeln. Anschließend sorgen wir dafür, dass die Bedingungen zueinander passen, also sich unabhängige Mengen in gültige Färbungen übersetzen und umgekehrt. Beachte, dass die erste Aussage des folgenden Theorems schon ausreicht, um NP-Schwere zu zeigen. Im zweite Teil misst k jeweils, wie einfach die Struktur der Graphen ist, und lässt uns zusätzlich folgern, dass es auch für strukturell relativ einfache Graphen schwer bleibt.

Theorem 1 ([2]). *Eine gegebene Instanz von MULTICOLORED INDEPENDENT SET kann in polynomieller Zeit in eine äquivalente Instanz von LIST COLORING übersetzt werden. Wenn die MULTICOLORED INDPENEDENT SET-Instanz k Farben verwendet, dann gibt es in der resultierenden LIST COLORING-Instanz k Knoten, deren Löschung alle Kanten entfernt.*

Beweis. Gegeben sei eine Instanz von MULTICOLORED INDEPENDENT SET, bestehend aus einem Graphen $G = (V, E)$ zusammen mit einer Färbung $\mathrm{col}\colon V \to C$ mit $k = |C|$ Farben. Für eine Farbe $c \in C$, sei $V_c \subseteq V$ die Menge der Knoten mit dieser Farbe, also $V_c = \{v \in V \mid \mathrm{col}(v) = c\}$.

Wir definieren nun eine Instanz von LIST COLORING wie folgt; siehe auch Abbildung 1.1c. Zunächst erstellen wir für jede Farbe in C einen Knoten. Als die erlaubten

Farben für einen Knoten $c \in C$ wählen wir V_c. Beachte, dass damit die Farbenmenge der List Coloring-Instanz gerade der Knotenmenge V der Multicolored Independent Set-Instanz entspricht. Damit haben wir die oben angedeutete Übersetzung von Farben in Knoten und Knoten in Farben gemacht.

Als nächstes übersetzen wir noch die Bedingungen. Beachte dazu, dass eine Kante $e = uv \in E$ beim Multicolored Independent Set fordert, dass höchstens einer der beiden Knoten u und v ausgewählt wird. Anders formuliert, dürfen wir nicht gleichzeitig für die Farbe $\mathrm{col}(u)$ den Knoten u und für die Farbe $\mathrm{col}(v)$ den Knoten v auswählen. Übersetzt in List Coloring bedeutet das, dass wir nicht gleichzeitig für den Knoten $\mathrm{col}(u)$ die Farbe u und für den Knoten $\mathrm{col}(v)$ die Farbe v auswählen dürfen. Das können wir erzwingen, indem wir in die List Coloring Instanz einen zusätzlichen Knoten für die Kante e einfügen, den wir ebenfalls e nennen. Als erlaubte Farben für e wählen wir die Menge $\{u, v\}$ und wir verbinden e mit $\mathrm{col}(u)$ und $\mathrm{col}(v)$. Das hat den Effekt, dass man für e eine der Farben u oder v auswählen muss. Damit kann man nicht gleichzeitig für den Nachbarn $\mathrm{col}(u)$ die Farbe u und für den Nachbarn $\mathrm{col}(v)$ die Farbe v auswählen, was genau der gewünschte Effekt war.

Zusammenfassend erhalten wir damit für die List Coloring-Instanz den Graphen $H = (V_H, E_H)$ mit $V_H = C \cup E$ und $E_H = \{\{e, c\} \mid e \in E, c \in C, e \cap V_c \neq \emptyset\}$. Die erlaubten Farben für $c \in C$ ist die Menge V_c und für $e \in E$ die Menge e selbst. Siehe auch nochmal Abbildung 1.1c. Beachte, dass das Löschen der k Knoten C aus H auch alle Kanten entfernt. Es bleibt also nur noch zu zeigen, dass G genau dann eine bunte unabhängige Menge der Größe k hat, wenn für H eine gültige Listenfärbung existiert.

Sei $S \subseteq V$ eine bunte unabhängige Menge der Größe $k = |S|$. Wir zeigen nun, dass H eine gültige Listenfärbung hat. Zunächst wählen wir Farben für die Knoten aus $C \subseteq V_H$. Sei $c \in C$ ein solcher Knoten. Da $|S| = k$, muss S einen Knoten v mit Farbe $\mathrm{col}(v) = c$ enthalten. Nach Konstruktion ist $v \in V_c$ eine erlaubte Farbe für c und wir wählen diese aus. Da es in H keine Kanten zwischen Knoten aus C gibt, haben bislang keine Nachbarn dieselbe Farbe. Wir müssen nun nur noch Knoten aus E Farben zuweisen. Sei $e \in E$ ein solcher Knoten mit erlaubten Farben $\{u, v\}$. Dann ist e in H mit $\mathrm{col}(u)$ und $\mathrm{col}(v)$ verbunden und außerdem war e in G die Kante uv. Da S eine unabhängige Menge ist, enthält S höchstens einen der Knoten u oder v. Wir nehmen ohne Einschränkung der Allgemeinheit an, dass $v \notin S$. Wir weisen dann e in H die Farbe v zu. Damit hat e eine andere Farbe als seine Nachbarn, denn $\mathrm{col}(u)$ hat v gar nicht als erlaubte Farbe und für $\mathrm{col}(v)$ haben wir nicht die Farbe v gewählt ($v \notin S$). Damit erhält jeder Knoten in H eine für ihn erlaubte Farbe und keine zwei Nachbarn haben dieselbe Farbe.

Für die andere Richtung sei $\mathrm{col}_H : V_H \to V$ eine gültige Listenfärbung von H. Wir müssen zeigen, dass es in G eine bunte unabhängige Menge der Größe k gibt. Das machen wir hier aber nicht, weil diese Seminararbeit sonst zu lang wird. □

1.3 Diskussion und Fazit

In Theorem 1 haben wir gesehen, wie man das Problem Multicolored Independent Set auf List Coloring reduzieren kann. Das Interessante daran ist, dass Multicolored Indpenedent Set selbst dann schwer ist, wenn die Anzahl k an Farben klein ist. Konkret gehen wir davon aus, dass es für keine berechenbare Funktion f einen Algorithmus mit Laufzeit $f(k) \cdot n^{O(1)}$ gibt, da sonst FPT = W[1]; für mehr Details zu diesen Komplexitätsklassen, siehe [1, Kapitel 13]. Damit zeigt Theorem 1 über die NP-Schwere von List Coloring hinaus, dass es (unter derselben Komplexitäts-Annahme) keinen Algorithmus mit Laufzeit $f(k) \cdot n^{O(1)}$ gibt, wobei k die Anzahl Knoten ist, die man löschen muss, um alle Kanten zu entfernen. Diesen Parameter k bezeichnet man auch als die *Vertex-Cover-Zahl* des Graphen. In der Sprache der parametrisierten Komplexität haben wir damit gezeigt, dass das mit der Vertex-Cover-Zahl parametrisierte Problem List Coloring $W[1]$-schwer ist.

Literatur

[1] Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk und Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi: 10.1007/978-3-319-21275-3.

[2] Michael R. Fellows, Fedor V. Fomin, Daniel Lokshtanov, Frances Rosamond, Saket Saurabh, Stefan Szeider und Carsten Thomassen. "On the Complexity of Some Colorful Problems Parameterized by Treewidth". In: *Information and Computation* 209.2 (2011), S. 143–153. doi: 10.1016/j.ic.2010.11.026.

[3] Larry Stockmeyer. "Planar 3-Colorability is Polynomial Complete". In: *SIGACT News* 5.3 (1973), S. 19–25. doi: 10.1145/1008293.1008294.

Kapitel 2

Ein interessantes Thema

Dein Name