

Seminarausarbeitungskonzept
On the Power of Color Refinement

Florian Lüdiger
2. Dezember 2017

im Rahmen des Seminars
Algorithm Engineering
von Prof. Dr. Petra Mutzel
Wintersemester 2017/18

Betreuer:
Christopher Morris

Basierend auf:
V. Arvind, Johannes Köbler, Gaurav Rattan und Oleg Verbitsky
On the Power of Color Refinement
https://link.springer.com/chapter/10.1007/978-3-319-22177-9_26

Fakultät für Informatik
Algorithm Engineering (Ls11)
Technische Universität Dortmund

Inhaltsverzeichnis

1	Einführung	4
2	Graph-Isomorphie und Color Refinement	4
2.1	Graph-Isomorphie	4
2.2	Color Refinement	4
2.3	Formale Definition und Amenability	6
3	Lokale Struktur zugänglicher Graphen	7
4	Globale Struktur zugänglicher Graphen	7
5	Laufzeitanalyse	7
6	Fazit	7
	Literaturverzeichnis	8

Glossar

bipartiter Graph

Ein Graph heißt bipartit, wenn sich seine Knoten in zwei Teilmengen aufteilen lassen, sodass Kanten nur zwischen den beiden Mengen aber nicht innerhalb existieren. $G[X, Y]$ ist der bipartite Graph, welcher durch die beiden disjunkten Teilmengen $X, Y \subseteq V(G)$ und allen Kanten, die Knoten aus X und Y verbinden, gebildet wird. 3

bipartites Komplement

Das bipartite Komplement eines Graphen G mit Knotenklassen X und Y stellt der bipartite Graph G' dar, welcher die selben Knotenklassen enthält, allerdings das Komplement der Kanten zwischen den beiden Knotenklassen. 3

biregulärer Graph

Ein bipartiter Graph G mit Knotenklassen X und Y ist biregulär, wenn alle Knoten in X und Y den gleichen Grad besitzen. 3

disjunkte Vereinigung

Die knotendisjunkte Vereinigung von G und H wird $G + H$ genannt. Die disjunkte Vereinigung von m Kopien des Graphen G wird als mG geschrieben. 3, 6

Hypergraph

Ein Hypergraph ist ein Graph, in dem eine Kante, auch Hyperkante genannt, mehr als zwei Knoten verbinden kann. 3

matching Graph

Ein matching Graph ist ein Graph, bei dem kein Knoten mehr als eine inzidente Kante besitzt. Somit gibt es nur Zusammenhangskomponenten mit maximal einer Kante und zwei Knoten. 3

Multimenge

Eine Multimenge unterscheidet eine Menge dadurch, dass Elemente mehrfach vorkommen können. Multimengen werden hier durch doppelte geschweifte Klammern dargestellt. 3, 5, 6

Nachbarschaft

Die Nachbarschaft $N(u)$ bildet die Menge der Knoten, die adjazent zu $u \in V(G)$ sind. 3, 4

regulärer Graph

Ein Graph ist regulär wenn alle seine Knoten den gleichen Grad besitzen. 3

Subgraph

Der Subgraph $G[X]$ ist ein Teilgraph von G , der durch die Knotenmenge $X \subseteq V(G)$ und deren inzidenten Kanten gebildet wird. 3

Unigraph

Die Isorphieeigenschaften eines Unigraphen sind durch die Sequenz der Knotengrade genau definiert. Dies bedeutet, dass allein anhand der Knotengrade zweier Unigraphen bestimmt werden kann, ob diese isomorph sind. 3

vollständiger Graph

In einem vollständigen Graphen K_n mit n Knoten, ist jeder Knoten mit jedem anderen Knoten verbunden und besitzt somit den Grad $n - 1$. 3

Zyklus

Ein geschlossener Pfad eines Graphen über n Knoten wird Zyklus C_n genannt. 3

1 Einführung

- Allgemeine Informationen zum Paper
- Was ist das Ziel der Forschung und inwiefern wurde dieses erreicht?
- Überblick über die Kapitelstruktur

2 Graph-Isomorphie und Color Refinement

2.1 Graph-Isomorphie

Zwei Graphen G und H sind isomorph, kurz $G \simeq H$, wenn es eine bijektive Abbildung ϕ der Knoten von G auf H gibt, sodass die Adjazenz aller Knoten erhalten bleibt. Es gilt also: $(u, v) \in E_G \Leftrightarrow (\phi(u), \phi(v)) \in E_H$ für alle $u, v \in V_G$. Eine so definierte Abbildung ϕ wird *Isomorphismus* genannt.

Es wurde gezeigt, dass das Graph-Isomorphie-Problem in NP liegt, die NP -Vollständigkeit ist allerdings noch unklar. Es konnte außerdem noch kein Algorithmus gefunden werden, welcher das Problem in polynomieller Zeit löst, jedoch wird davon ausgegangen, dass es nicht NP -vollständig ist. Siehe Goldreich et al. (1991).

2.2 Color Refinement

Der Color Refinement Algorithmus stellt eine Heuristik dar, mit der in polynomieller Zeit festgestellt werden kann, dass zwei Graphen nicht isomorph sind. Der Algorithmus geht wie folgt vor.

1. Sämtliche Knoten des Graphen werden in der selben Farbe gefärbt.

2. Die Knotenfärbungen werden verfeinert, indem überprüft wird, ob zwei Knoten gleicher Farbe unterschiedliche *Nachbarschaften* mit Berücksichtigung der Farbe besitzen. Ist dies der Fall, werden die Knoten in unterschiedlichen Farben gefärbt und das Verfeinern wird fortgeführt.
3. Ist die Bedingung für kein Knotenpaar mehr erfüllt, terminiert der Algorithmus.
4. Sind die *Multimengen* der Farben beider Graphen unterschiedlich, sind diese nicht isomorph.

Es kann bei diesem Vorgehen vorkommen, dass zwei nicht isomorphe Graphen G und H nicht voneinander unterschieden werden können. Ein Beispiel hierfür wird in Abbildung 1 dargestellt. Zu erkennen ist, dass alle Knoten beider Graphen initial in einer Farbe gefärbt wurden, danach allerdings kein Verfeinerungsschritt mehr nötig ist, da die Nachbarschaften aller Knoten identisch gefärbt sind. Obwohl offensichtlich erkennbar ist, dass die beiden Graphen nicht isomorph sind, gelingt es dem Algorithmus also nicht diese zu unterscheiden. Auf Basis dieser Erkenntnisse wird die folgende Klasse von Graphen definiert.

Definition 1 Ein Graph G wird *amenable* genannt, wenn der Color Refinement Algorithmus diesen von jedem beliebigen, nicht zu G isomorphen Graphen H unterscheiden kann.

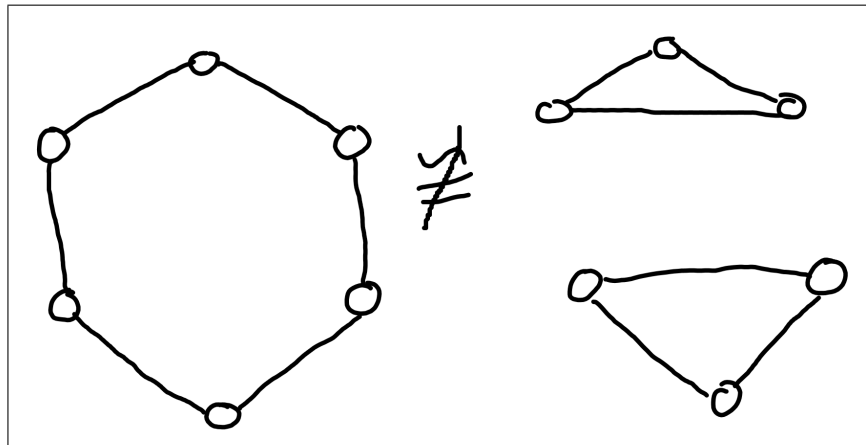


Abbildung 1: Zwei nicht isomorphe Graphen, welche vom Color Refinement nicht unterschieden werden können

Neben dem klassischen Isomorphie-Test ist der Color Refinement Algorithmus auch für andere Anwendungsfelder geeignet. Als Beispiel ist hier die Verkleinerung von linearen Programmen durch Reduktion der Dimensionen der Gleichungssysteme und somit eine Erhöhung der Effizienz zu nennen. Siehe Grohe et al. (2014). Weitere Anwendungsfelder finden sich in Shervashidze et al. (2011) und Kersting et al. (2014).

2.3 Formale Definition und Amenability

Der Color Refinement Algorithmus berechnet iterativ eine Sequenz von Knotenfärbungen C^i für einen Graphen G . Die initiale Färbung C^i weist jedem Knoten die selbe Farbe zu. Nachfolgend wird in jeder Iteration nach folgender Regel eine neue Färbung gebildet.

$$C^{i+1}(u) = (C^i(u), \{\{C^i(a) : a \in N(u)\}\}) \quad (1)$$

Die doppelten geschweiften Klammern $\{\{\}$ markieren hierbei eine *Multimenge*. Die Knoten von G werden mithilfe einer Partitionierung \mathcal{P} in die entsprechenden Farbklassen unterteilt, wodurch also zu jeder Färbung C^i eine Partition \mathcal{P}^i gehört.

Bei der Ausführung des Algorithmus wird irgendwann eine Partitionierung erreicht, die sich bei weiteren Verfeinerungsschritten nicht mehr verändert, da keine neue Färbung mehr generiert wird. Für diese gilt das Folgende.

Definition 2 Wenn sich eine Partitionierung bei weiteren Verfeinerungsschritten nicht mehr verändert, stellt dies die stabile Partitionierung \mathcal{P}^s dar. Für sie gilt $\mathcal{P}^s = \mathcal{P}^i$ für alle $i \geq s$.

Die stabile Partitionierung ist bei jedem Graphen nach maximal $n - 1$ Verfeinerungsschritten erreicht, da spätestens dann jeder Knoten eine unterschiedliche Farbe besitzt und somit keine weitere Verfeinerung mehr möglich ist.

Eine Partitionierung kann außerdem die Eigenschaft *equitable* haben. Dafür müssen die im Folgenden definierten Eigenschaften erfüllt sein. Die Elemente der Partitionierung werden hier Zellen genannt.

Definition 3 Eine Partitionierung wird *equitable* genannt, wenn die folgenden Eigenschaften erfüllt sind:

1. Jede Zelle $X \in \mathcal{P}$ ist einfarbig und enthält somit nur Knoten einer einzigen Farbe.
2. Für jede Zelle $X \in \mathcal{P}$ ist der Graph $G[X]$ ein regulärer Graph.
3. Für beliebige Zellen $X, Y \in \mathcal{P}$ ist der bipartite Graph $G[X, Y]$ ein biregulärer Graph.

Es ist somit leicht zu erkennen, dass die durch das Color Refinement erstellte stabile Partitionierung \mathcal{P}^s equitable ist.

Eine weitere Eigenschaft der Graphfärbungen C^i ist der Erhalt der Färbung über Isomorphismen hinweg.

Lemma 1 Für die Färbungen zweier isomorpher Graphen G und H und ihren Isomorphismus ϕ muss gelten: $C^i(u) = C^i(\phi(u))$ für alle $u \in V_G$.

Aus dieser Aussage leitet sich die folgende Gleichung her.

$$\forall i \geq 0 : C^i(u) : u \in V_G = \{\{C^i(v) : v \in V_H\}\} \quad (2)$$

Das Color Refinement akzeptiert somit zwei Graphen G und H bei der Eingabe der *disjunkten Vereinigung* $G + H$ genau dann, wenn die Gleichung (2) erfüllt ist.

Das Überprüfen dieser Bedingung ist in endlicher Zeit berechenbar. Für einen Zeugen i dafür, dass die Gleichung nicht erfüllt ist, gilt $i < 2n$, wobei n die Anzahl der Knoten jedes Graphens ist. Dies beruht darauf, dass \mathcal{P}^{2n-1} in jedem Fall die stabile Partitionierung von $G + H$ ist und weitere Verfeinerungsschritte keine neue Partitionierung erstellen würden. Es genügt sogar die Gleichung für $i = n$ zu verifizieren, da die Existenz einer Partitionierung $\mathcal{P}^{i+1} \neq \mathcal{P}^i$ bedeuten würde, dass es mehr als n Partitionen, da in jedem Partitionierungsschritt mindestens eine Partition hinzukommt. Mehr als n Partitionen sind ein Indikator dafür, dass die Graphen G und H nicht isomorph sind, da die n Knoten jedes Graphen unmöglich in mehr als n Partitionen unterteilt werden können und Gleichheit somit ausgeschlossen ist.

Aus den gewonnenen Erkenntnissen ergibt sich folgende Erweiterung zu Definition 1.

Definition 4 Ein Graph G wird *amenable* genannt, wenn für jeden beliebigen, nicht isomorphen Graphen H die Gleichung (2) für $i = n$ nicht erfüllt ist.

3 Lokale Struktur zugänglicher Graphen

- Erklärung der lokalen Struktur zugänglicher Graphen (A,B)
- jeweils gegebenenfalls mit ausführlicher Erklärung der Bedeutung und Beispielen
- Fokussierung auf die Beweisidee für die jeweiligen Lemmata
- Ausführlichere und leichter verständliche Aufbereitung einiger oder aller Beweise aus dem Paper

4 Globale Struktur zugänglicher Graphen

- Erklärung der globalen Struktur zugänglicher Graphen (C,D,E,F,G,H)
- Ansonsten wie Kapitel 3

5 Laufzeitanalyse

- Beweis der Laufzeit von $O((n + m) \log n)$

6 Fazit

- Zusammenfassung der Ergebnisse
- Bezugnehmen auf die Einleitung und ob die Ziele erreicht wurden
- Abschluss der Arbeit

Literatur

- O. Goldreich, S. Micali, und A. Wigderson. Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *J. ACM*, 38(3):690–728, Jull 1991. ISSN 0004-5411. doi:10.1145/116825.116852. URL <http://doi.acm.org/10.1145/116825.116852>.
- M. Grohe, K. Kersting, M. Mladenov, und E. Selman. *Dimension Reduction via Colour Refinement*, Seiten 505–516. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. ISBN 978-3-662-44777-2. doi:10.1007/978-3-662-44777-2_42. URL https://doi.org/10.1007/978-3-662-44777-2_42.
- K. Kersting, M. Mladenov, R. Garnett, und M. Grohe. Power iterated color refinement. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, Seiten 1904–1910. AAAI Press, 2014.
- N. Shervashidze, P. Schweitzer, E. J. v. Leeuwen, K. Mehlhorn, und K. M. Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(Sep):2539–2561, 2011.