

Inhaltsverzeichnis

Baum Wald Fluss

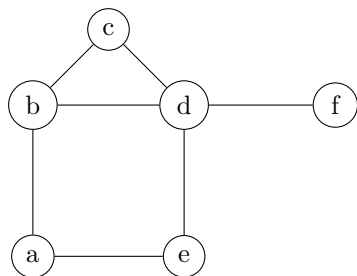
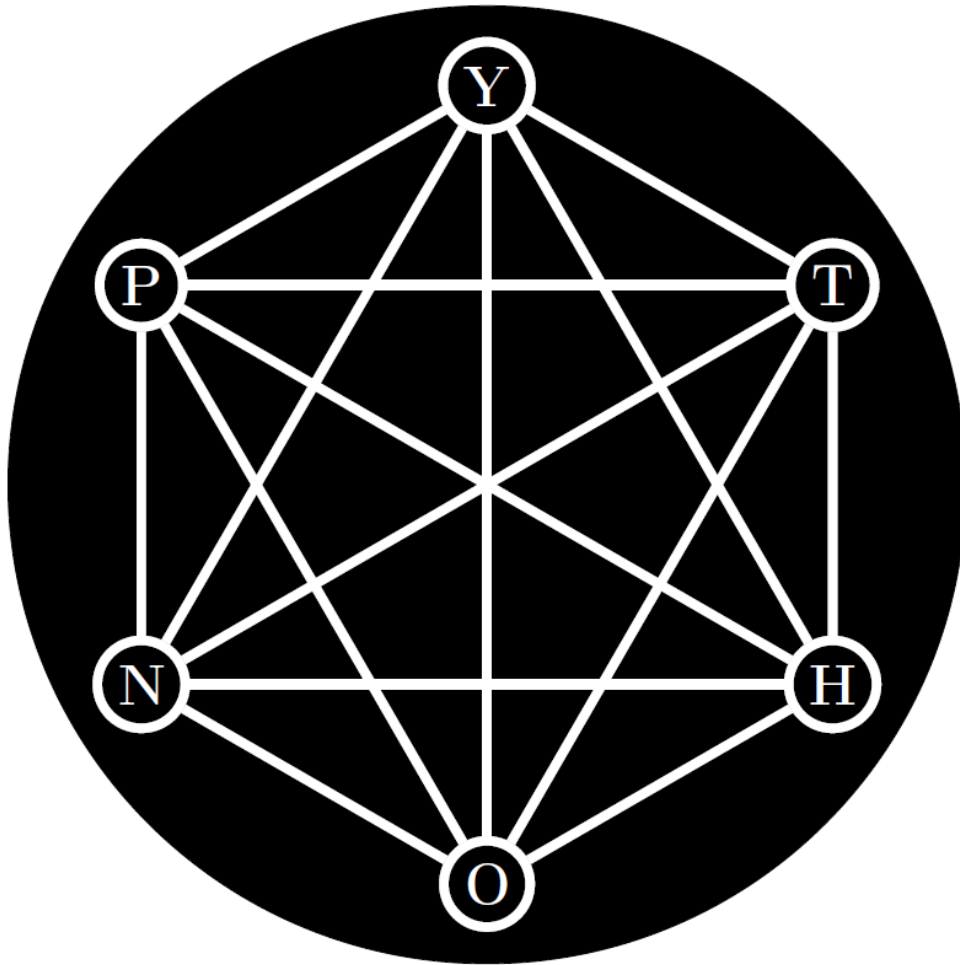


Abb. 1.1: Ein Graph

1 Aus der Kursbeschreibung

Markus Oehme und Sebastian Brandt

Bei Gelegenheit könnte hier Inhalt stehen.[?, s. Literaturverzeichnis]

2 Graphen

Jakob Gierschmann und Annika Dunkel

2.1 Was sind Graphen?

Ein **Graph** besteht aus **Knoten** und **Kanten**. Abb. 1.1 zeigt ein einfaches Beispiel für einen Graphen.

Definition 1. Ein Graph ist ein Tupel $G = (V, E)$ aus einer endlichen Knotenmenge V und einer Kantenmenge $E \subseteq \{e \in 2^V \mid |e| = 2\}$.

2.2 Geschichte der Graphentheorie

Leonhard Euler (1707–1783) wird im Allgemeinen als Begründer der Graphentheorie bezeichnet. Er befasste sich mit Problemstellungen wie dem **Königsberger Brückenproblem**. Auch wenn er bereits die ersten Konzepte der Graphentheorie auf aufstellte, entwickelte sich die Graphentheorie in den folgenden 100 Jahren kaum weiter. Erst *Gustav Robert Kirchhoff* (1824–1887), der elektrische Ströme mithilfe der Graphentheorie darstellte, ermöglichte dadurch den nächsten großen Durchbruch.

Schließlich brachte *Arthur Cayley* (1821–1895) die Strukturformeln von chemischen Molekülen in die Graphentheorie ein. Außerdem stellte er den **Vierfarbensatz** auf. Ab 1945 machte die Graphentheorie aufgrund der rasanten Computerentwicklung noch größere Fortschritte.

2.3 Teilgraphen

Ein Teilgraph G' besteht aus einem Teil der Knoten und Kanten eines übergeordneten Graphen G .

Definition 2. Ein Graph $G' = (V', E')$ ist ein Teilgraph von $G = (V, E)$, wenn $V' \subseteq V$ und $E' \subseteq E$, sodass für alle $e = \{v_1, v_2\} \in E'$ auch $v_1, v_2 \in V'$.

Definition 3. Ein Teilgraph $G' = (V', E')$ von $G = (V, E)$ heißt von V' induzierter Teilgraph, wenn jede Kante $e \in E$, die zwischen Knoten $v'_1, v'_2 \in V'$ verläuft, auch in E' vorhanden ist.

2.4 Verbindungen

Graphen haben einige wichtige Eigenschaften, die wir im Folgenden beschreiben und definieren wollen.

Definition 4. Ein Weg der Länge n ist eine Folge von Knoten $v_0, v_1, v_2, \dots, v_n \in V$, sodass Kanten $e = \{v_{i-1}, v_i\} \in E$ für $1 \leq i \leq n$ existieren.

Definition 5. Ein Pfad ist ein Weg, bei dem alle Knoten paarweise verschieden sind. v_0 heißt Startknoten und v_n heißt Zielknoten.

Definition 6. Ein Graph $G = (V, E)$ heißt zusammenhängend, wenn zu je zwei Knoten $v_1, v_2 \in V$ ein Pfad von v_1 nach v_2 existiert.

Definition 7. Die maximalen zusammenhängenden Teilgraphen heißen Zusammenhangskomponenten.

2.5 Variationen

Neben den gewöhnlichen Graphen gibt es einige **Variationen**, die in der Regel ausdrücklich als solche gekennzeichnet werden.

- **Multigraphen** Ein Multigraph lässt mehrere Kanten statt nur einer Kante zwischen zwei Knoten zu.
- **Gerichteter Graphen** Bei einem gerichteten Graphen sind die Kanten keine Mengen, sondern Tupel. Die Knoten, die sie verbinden haben also eine Reihenfolge und somit hat die Kante eine feste Richtung.
- **Graphen mit Schleifen** Wenn ein Knoten durch eine Kante mit sich selbst verbunden ist, ist diese Kante eine Schleife. In Graphen, die Schleifen zulassen, ist also $|e| = 1$ zulässig.

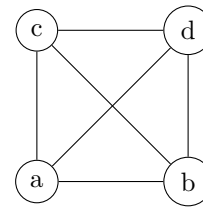


Abb. 1.2: Der Graph K_4

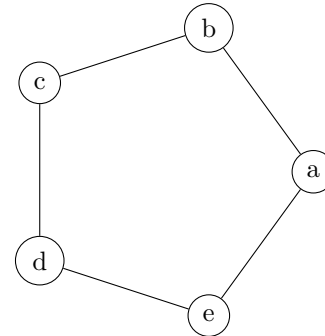


Abb. 1.3: Der Graph C_5

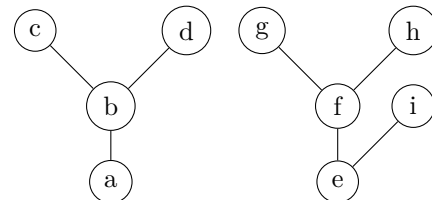


Abb. 1.4: Ein Wald aus zwei Bäumen

- **Unendliche Graphen** Wenn Graphen eine unendliche Anzahl von Knoten V und/oder Kanten E haben, werden sie unendliche Graphen genannt.

2.6 Spezielle Graphen

In der Graphentheorie gibt es spezielle Graphen, die besondere Eigenschaften haben. Einige von ihnen nennen wir kurz, um uns später auf sie beziehen zu können.

Definition 8. Ein vollständiger Graph K_n ist ein Graph $G = (V, E)$ mit n Knoten, bei dem je zwei Knoten $v, w \in V$ durch eine Kante $e = \{v, w\}$ verbunden sind.

Ein Beispiel für einen vollständigen Graphen ist der Graph K_4 in Abb. 1.2.

Definition 9. Ist v_0, v_1, \dots, v_n ein Pfad und existiert $e_0 = \{v_n, v_0\}$ so ist v_0, v_1, \dots, v_n ein Kreis. Einen Kreis mit n Knoten nennen wir C_n .

Ein Beispiel für einen Kreis ist der Graph C_5 in Abb. 1.3.

Definition 10. Ein Wald ist ein Graph ohne Kreise. Die Zusammenhangskomponenten heißen Bäume.

Abb. 1.4 zeigt ein Beispiel für einen Wald.

2.7 Wichtige Messgrößen

Es gibt in Graphen einige Meßgrößen, die wir durch Funktionen oder Variablen angeben können. Einige seien hier kurz definiert.

Der Grad eines Knotens ist ein Beispiel für eine lokale Messgröße, da wir ihn beschreiben können, ohne den ganzen Graphen zu betrachten.

Definition 11. Sei $v \in V$ ein Knoten. Der Grad von v , geschrieben $d(v)$, ist die Anzahl an Kanten, die v enthalten.

$$d(v) := |\{e \in E \mid v \in e\}|$$

Definition 12. Das Minimum aller Grade der Knoten $v \in V$ wird mit δ bezeichnet, das Maximum mit Δ .

Fakt 1. Die Summe $\sum_{v \in V} d(v)$ ist gerade und $|E| = \frac{1}{2} \sum_{v \in V} d(v)$.

Beweis. Jede Kante verbindet zwei Knoten, also erhöht jede Kante die Summe aller Grade um 2. \square

Mit den gerade definierten Eigenschaften von Graphen können wir eine weitere Funktion einführen, die wir benutzen können, um Knoten besser differenzieren zu können.

Definition 13. Der Abstand $d(v, v')$ zwischen zwei Knoten $v, v' \in V$ ist die Länge des kürzesten Pfades von v nach v' . Gibt es keinen solchen Pfad, so setzen wir $d(v, v') = \infty$.

3 Formalismus und Beweisführung

Felix Forner und Anna Siess

3.1 Formalismus

In der Mathematik dient in der Regel die **Prädikatenlogik** als Grundlage für Schlussfolgerungen und Beweise. Wir arbeiten meistens mit einer Prädikatenlogik, in der es nur die Wahrheitswerte „wahr“ und „falsch“ gibt, wobei eine Aussage immer genau einen davon hat. Prädikate sind Aussagen mit Platzhaltern.

3.1.1 Verknüpfungen

Der Zusammenhang zweier Aussagen lässt sich durch Verknüpfungen herstellen, die durch folgende Operatoren repräsentiert werden.

- **Konjunktion** $A \wedge B$: wahr, wenn sowohl A als auch B wahr sind.
- **Disjunktion** $A \vee B$: wahr, wenn mindestens eine der Aussagen wahr ist.
- **Negation** $\neg A$: wahr, wenn A falsch ist.

- **Implikation** $A \Rightarrow B$: wahr, wenn die Wahrheit von A stets die Wahrheit von B nach sich zieht. D. h., dass $A \Rightarrow B$ nur falsch ist, wenn A wahr und B falsch ist. Wenn $A \Rightarrow B$ und A wahr sind, so muss also auch B wahr sein. Hat A den Wahrheitswert falsch, so kann B sowohl wahr als auch falsch sein.

- **Äquivalenz** $A \Leftrightarrow B$: wahr, wenn A und B die gleichen Wahrheitswerte haben.

3.1.2 Quantoren

Quantoren drücken aus, für welche Objekte eine Aussage gilt. Man unterscheidet im Wesentlichen zwischen zwei Quantoren.

- **Existenzquantor** \exists : „es gibt ein“
- **Allquantor** \forall : „für alle“

Bsp.: $\forall n \in \mathbb{N} : n \in \mathbb{Z}$ (Jede natürliche Zahl ist auch eine ganze Zahl.)

3.2 Beweise

Um logische Schlussfolgerungen, sogenannte **Ableitungen**, vornehmen zu können, sind Grundlagen nötig, die **Axiome** genannt werden. Aus diesen kann man weitere Fakten folgern. Axiome sind dadurch gekennzeichnet, dass sie selbst nicht durch Schlussfolgerungen begründet werden können. Meist wird die Zermelo-Fraenkel-Mengenlehre als Basis verwendet. Strukturierte Ableitungsketten, also eine Reihe von aufeinander aufbauenden Ableitungen, sind **Beweise**. Es gibt verschiedene Arten von Beweisen, die sich durch die Strategien auszeichnen, die ihnen zu Grunde liegen.

• Direkter Beweis

Wahrheit von B wird durch Implikationen aus der Wahrheit von A gefolgert.

Bsp.: Zu beweisen ist, dass die Summe dreier aufeinanderfolgender natürlicher Zahlen stets durch drei teilbar ist.

Beweis. Sei n eine natürliche Zahl, dann sind die folgenden natürlichen Zahlen $n+1$ und $n+2$. Die Summe dieser drei Zahlen ist also $n + (n+1) + (n+2) = n + n + n + 1 + 2 = 3n + 3$. Sowohl $3n$ als auch 3 sind durch drei teilbar. Damit ist auch $3n + 3$, also die Summe dreier aufeinanderfolgender natürlicher Zahlen durch drei teilbar. \square

• Indirekter Beweis

Annahme des Gegenteils und Herleitung eines Widerspruchs.

- **(Gegen-)Beispiel**

Zeigen einer \exists -Aussage („es existiert ein“) durch Beispiel oder Widerlegen einer \forall -Aussage („für alle ... gilt“) durch Gegenbeispiel.

Bsp.: Um die Aussage „Es gibt keine Primzahlzwillinge“ zu widerlegen, genügt es, ein Gegenbeispiel anzuführen, z. B. die Zahlen 3 und 5. (Anmerkung: Primzahlzwillinge sind zwei Primzahlen mit einer Differenz von 2.)

- **Kontraposition**

Betrachtung von $\neg B \Rightarrow \neg A$ statt $A \Rightarrow B$, da diese beiden Aussagen äquivalent sind.

Bsp.: Anstatt zu zeigen, dass jedes gleichseitige Dreieck gleichschenkelig ist („D ist gleichseitig.“ \Rightarrow „D ist gleichschenkelig.“) genügt es, zu beweisen, dass jedes Dreieck, das nicht gleichschenkelig ist, auch nicht gleichseitig sein kann („D ist nicht gleichschenkelig.“ \Rightarrow „D ist nicht gleichseitig.“).

- **o. B. d. A.** (ohne Beschränkung der Allgemeinheit)

Betrachtung eines einfachen Falls bei symmetrischen Problemen. Demnach reicht es, falls mehrere Fälle vorliegen, die analog zueinander sind, einen zu behandeln. Unter Symmetrien fallen nicht nur geometrische, sondern auch abstrakte Symmetrien.

- **Ringschluss**

Gelten die Implikationen $(A_1 \Rightarrow A_2), (A_2 \Rightarrow A_3), \dots, (A_{k-1} \Rightarrow A_k), (A_k \Rightarrow A_1)$, so sind die Aussagen $A_1, A_2, A_3, \dots, A_{k-1}, A_k$ äquivalent.

- **Induktion**

Möchte man beweisen, dass eine Aussage A für jede natürliche Zahl $n \geq n_0$ gilt, so hilft oft ein induktives Verfahren. Eine Aussage $A(n)$ ist für jedes n wahr, wenn

- (1) $A(n_0)$ wahr ist (Induktionsanfang) und
- (2) $\forall n \geq n_0$: Ist $A(n)$ wahr, so ist auch $A(n+1)$ wahr (Induktionsschritt).

Bsp.: Zu beweisen ist, dass $n(n+1)$ für jede natürliche Zahl n gerade ist.

Beweis. Induktionsanfang: Für $n = 1$ gilt: $1(1+1) = 2$.

Induktionsschritt: Es ist zu zeigen, dass die Aussage unter der Annahme, dass sie für n gilt, auch für $n+1$ gilt. Setzt man $n+1$ in den Term ein, so erhält man: $(n+1)(n+1+1) = (n+1)(n+2)$. Durch Umformung erhält man $(n+1)(n+2) = n(n+1) + 2(n+1)$. Dabei handelt es sich um eine Summe mit den

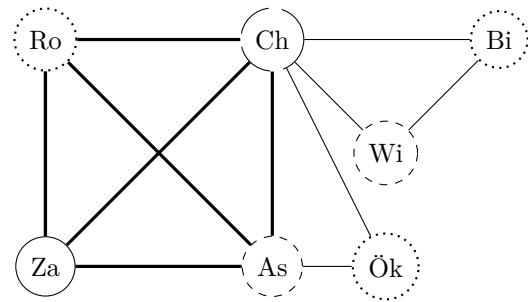


Abb. 1.5: Dies ist der Graph, der die Kurswahlen aus Tabelle 1.1 darstellt. Kurse werden als Knoten dargestellt, die Konflikte zwischen Kursen als Kanten. Die Farben werden durch verschiedene Knotenumrandungen repräsentiert.

Summanden $n(n+1)$ und $2(n+1)$. Sind beide Summanden gerade, so auch die Summe. Die Induktionsannahme besagt, dass $n(n+1)$ gerade ist, wobei es sich um den ersten Summanden handelt. Der zweite Summand $2(n+1)$ ist offensichtlich ein Vielfaches von 2 und damit gerade. Somit ist auch $n(n+1) + 2(n+1)$ gerade. Hiermit ist bewiesen, dass $n(n+1)$ für jede natürliche Zahl n gerade ist. \square

4 Färbungen

Josua Kugler und Amelie Koch

Ein Teilgebiet der Graphentheorie, nämlich die Färbung von Graphen, bietet Raum für zahlreiche Anwendungen. Im Folgenden möchten wir ein solches Anwendungsbeispiel näher betrachten.

4.1 Knotenfärbung

Angenommen, die DSA kann im Jahr 2028 allen Teilnehmern ermöglichen, alle gewählten Kurse zu besuchen. Wie viele Akademien müssen bei einer Kurswahl wie in Tabelle 1.1 mindestens stattfinden, wenn alle TeilnehmerInnen nur einen Kurs pro Akademie besuchen können?

Für die geringste Zahl an Akademien muss versucht werden, möglichst viele TeilnehmerInnen auf einer Akademie unterzubringen. Wie können wir an dieses Problem herangehen? Hier hilft ein graphentheoretischer Ansatz weiter. Hierfür wählen wir die Kurse als Knoten. Kurse, die nicht gleichzeitig stattfinden dürfen, da sie beide von der gleichen Person gewählt wurden, werden durch eine Kante verbunden. Diese Kurse stehen in Konflikt miteinander, sie müssen also auf verschiedenen Akademien stattfinden. Wir wissen also, dass benachbarte Knoten auf unterschiedlichen Akademien stattfinden müssen. Dies können wir veranschaulichen, indem benachbarte Knoten unterschiedlich eingefärbt werden. Jede Akademie wird dann durch eine andere Farbe repräsentiert. Wie in Abb. 1.5 zu erkennen, benötigt man in unserem Beispiel höchstens vier Farben bzw. Akademien. Kann diese Zahl

Zahlentheorie	Robotik	Astronomie	Ökologie	Wirtschaft	Biologie	Chemie
T_1	T_3	T_3	T_5	T_6	T_6	T_6
T_2	T_4	T_5				T_5
	T_1	T_4				T_4
	T_2	T_2				T_1

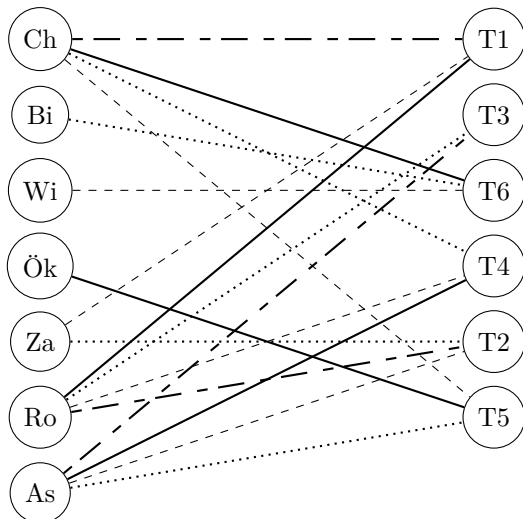
Tab. 1.1: Kurswahl von 6 TeilnehmerInnen T_1 bis T_6 

Abb. 1.6: Kurse werden durch die Knoten auf der linken Seite dargestellt, TeilnehmerInnen auf der rechten Seite, Kanten stellen die Belegungen dar, unterschiedliche Kantenfärbungen entsprechen der Art der Linie.

verringert werden? Dass dies nicht möglich ist, erkennt man an den Kanten zwischen Robotik, Chemie, Zahlentheorie und Astronomie im Graphen (fett hervorgehobene Kanten in Abb. 1.5). Diese Knoten sind vollständig miteinander verbunden, sodass sie verschieden eingefärbt sein müssen.

Definition 14. Eine **k-Färbung** eines Graphen $G = (V, E)$ ist eine Funktion $c : V \rightarrow \{1, 2, \dots, k\}$, sodass für jede Kante $e = \{v_1, v_2\}$ gilt: $c(v_1) \neq c(v_2)$.

Ein Graph heißt **k-färbbar**, wenn eine k-Färbung des Graphen existiert.

Definition 15. Die **chromatische Zahl** $\chi(G)$ eines Graphen $G = (V, E)$ ist das kleinste k , sodass G k-färbbar ist.

4.2 Kantenfärbung

Die DSA bietet im Jahr 2038 allen TeilnehmerInnen Einzelunterricht an. Auf einer Akademie wird jeder Kurs weiterhin höchstens einmal angeboten und alle TeilnehmerInnen belegen jeden gewählten Kurs genau einmal. Jeder Kurs wird auf mehreren Akademien angeboten und jeder Teilnehmer und jede Teilnehmerin besucht auf einer Akademie genau einen Kurs. Wie viele Akademien werden nun bei den Kurswahlen aus Tabelle 1.1 benötigt?

Das Problem liegt hier darin begründet, die von den TeilnehmerInnen gewählten Kurse so auf unterschiedliche Akademien zu verteilen, dass insgesamt möglichst wenige Akademien stattfinden müssen. Auch dieses Verteilungsproblem lässt sich mathematisch mittels eines Graphen beschreiben. Allerdings müssen jetzt nicht mehr nur die Kurse, sondern auch TeilnehmerInnen auf Akademien verteilt werden. Wir müssen also stets Paare (Kurs, Teilnehmer) einzelnen Akademien zuordnen. Wie kann dies im Graph dargestellt werden?

Das Tupel (Kurs, Teilnehmer) entspricht in einem Graphen der Kante zwischen dem Knoten „Kurs“ und dem Knoten „Teilnehmer“. Diese Knoten werden dann entsprechend ihrer Belegungen verbunden. Welche dieser Tupel dürfen nicht zusammen auf einer Akademie vorkommen? Stimmen TeilnehmerIn oder Kurs (also ein Knoten) zweier solcher Tupel überein, so können diese nicht beide auf einer Akademie sein. Kennzeichnet man nun wie oben unterschiedliche Akademien mit unterschiedlichen Farben, ergibt sich folgende Regel für die Färbung: Zwei Kanten, die mit dem gleichen Knoten verbunden sind, dürfen nicht die gleiche Farbe haben. So kann über eine Kantenfärbung (oder äquivalent eine Knotenfärbung des Liniengraphen) eine Lösung für das Problem gefunden werden. Wie in Abb. 1.6 zu erkennen ist, reichen auch in diesem Fall vier Farben aus. Dieser Graph kann nicht mit weniger Farben gefärbt werden, da es Knoten mit Grad vier gibt.

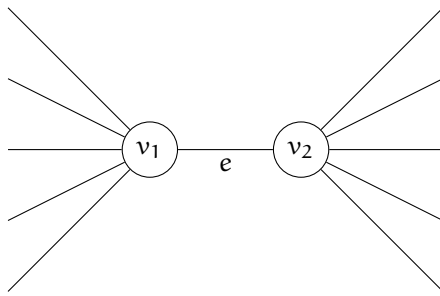
Definition 16. Eine **k-Kantenfärbung** eines Graphen $G = (V, E)$ ist eine Funktion $c : E \rightarrow \{1, \dots, k\}$, sodass für jeden Knoten v gilt: Für alle $e_1, e_2 \in E$ mit $e_1 \neq e_2$ und $v \in e_1, v \in e_2$ gilt $c(e_1) \neq c(e_2)$.

Ein Graph heißt **k-kantenfärbbar**, wenn eine k-Kantenfärbung des Graphen existiert.

Definition 17. Das kleinste k , für das ein Graph G k-kantenfärbbar ist, heißt **chromatischer Index** $\chi'(G)$.

4.3 Schranken für $\chi(G)$ und $\chi'(G)$

Es stellt sich nun die Frage, ob man die chromatische Zahl bzw. den chromatischen Index abschätzen kann, wenn man den Maximalgrad Δ eines Graphen kennt. Eine untere Schranke für $\chi'(G)$ ist bereits dadurch gegeben, dass bei einem Knoten mit

Abb. 1.7: Obere Schranke für $\chi'(G)$?

dem Maximalgrad Δ genau Δ Kanten zusammenlaufen, man benötigt also mindestens Δ Farben für eine Kantenfärbung.

Um eine Kantenfärbung zu bestimmen, kann man nach folgendem Greedy-Algorithmus vorgehen: Um die Farbe einer Kante $e = \{v_1, v_2\}$ zu bestimmen, betrachtet man die Farben aller angrenzenden Kanten e_i mit $v_1 \in e_i \vee v_2 \in e_i$ und färbt die Kante e mit einer Farbe, die nicht für eine der Kanten e_i verwendet wurde. Eine obere Schranke kann man zunächst einmal dadurch bestimmen, dass man den Worst Case betrachtet: Eine Kante e , deren Knoten v_1 und v_2 jeweils den Maximalgrad besitzen (siehe Abb. 1.7 mit $\Delta = 6$). Sollten alle Kanten unterschiedlich gefärbt sein (es sei dahingestellt, ob das tatsächlich auch möglich ist), so kommt man mithilfe des Greedy-Algorithmus auf $2\Delta - 2$ zu e „benachbarte“ Kanten und damit $2\Delta - 1$ Farben. Diese obere Schranke lässt sich aber noch weiter herabsetzen.

Satz 1 (Vizing). $\Delta \leq \chi'(G) \leq \Delta + 1$

Ähnliches lässt sich auch für die chromatische Zahl feststellen. Wenn man von $|V| \leq 1$ absieht, beträgt hier die untere Schranke 2. Das liegt daran, dass bipartite Graphen mit beliebigem Maximalgrad mit zwei Farben gefärbt werden können. Für vollständige Graphen erhält man bereits $\Delta + 1$ benötigte Farben, da jeder Knoten mit Δ anderen Knoten verbunden ist und es daher nicht möglich ist, eine Farbe zweimal zu verwenden. Zu zeigen bleibt, dass es keine Graphen gibt, die eine größere Anzahl an Farben benötigen. Folgender nicht leicht zu beweisender Satz bestätigt unsere Vermutung.

Satz 2 (Brooks). $\chi(G) \leq \Delta + 1$ und die einzigen zusammenhängenden Graphen mit $\chi(G) = \Delta + 1$ sind die vollständigen Graphen und Kreise mit ungerader Anzahl an Knoten.

Ein mögliches Anwendungsbeispiel ist die Organisation eines Schachturniers mit n TeilnehmerInnen. Wie viele Runden müssen mindestens gespielt werden, damit jeder gegen jeden spielt? Auch dieses Problem lässt sich mithilfe eines Graphen

modellieren. Die TeilnehmerInnen werden als Knoten dargestellt, ein Spiel als Kante zwischen den Gegnern. Die Lösung des Problems besteht in einer Kantenfärbung, wobei eine Farbe einer Runde entspricht. Nach dem Satz von Vizing müssen höchstens $\Delta + 1$, also in unserem Fall $(n - 1) + 1 = n$ Runden gespielt werden.

5 Planarität

Janek Paeßens und Charlotte Häußler

5.1 Definitionen und Motivation

Im Folgenden soll die Planarität von Graphen behandelt werden, die relevante Anwendungen, wie auch eine interessante Theorie mit sich bringt. Planare Graphen sind Graphen, die sich überschneidungsfrei auf ein Blatt Papier zeichnen lassen. Beispiele für die Anwendung von planaren Graphen lassen sich im Chip-Design bzw. in der Platinen-Entwicklung finden, in der die Schaltkreise nur an möglichst wenigen Stellen Überschneidungen haben sollten.

Definition 18. Ein Graph $G = (V, E)$ heißt **planar**, wenn er sich so in die Ebene einbetten lässt, dass sich keine Kanten schneiden.

Ob ein Graph planar ist oder nicht hängt dabei nicht von seiner Einbettung ab. Es ist also durchaus möglich, dass ein Graph trotz vorhandener Planarität mit Kantenüberschneidungen dargestellt wird.

Definition 19. Ein Graph heißt **maximal planar**, wenn er planar ist und keine weitere Kante hinzugefügt werden kann, ohne dass die Planarität zerstört wird.

Definition 20. Ein **triangulierter Graph** ist ein planarer Graph der in einer spezifischen Einbettung nur aus Dreiecken besteht und auch in seiner Außenform ein Dreieck darstellt.

In einem triangulierten Graphen bilden also alle Knoten mit ihren Nachbarknoten und den entsprechenden Kanten immer Dreiecke.

5.2 Eigenschaften planarer Graphen

Das folgende Lemma ist eine Aussage über planare Graphen, die dazu dient numerische Invarianten eines Graphen, in diesem Fall die Größe der Kanten- und Knotenmenge, miteinander in Beziehung zu setzen.

Lemma 1. Sei $G = (V, E)$ ein planarer Graph, der mindestens drei Knoten hat. Dann gilt

$$|E| \leq 3 \cdot |V| - 6.$$

Beweis. Der Beweis geschieht mit Induktion über die Knotenanzahl $|V|$. O.B.d.A. wird er nur für triangulierte Graphen durchgeführt werden. Triangulierte Graphen sind maximal planar und

damit ist die Kantenanzahl im Verhältnis zur Knotenanzahl ebenfalls maximal. Somit wird nur der worst case betrachtet.

Als Induktionsbeginn wird das obige Lemma für die kleinste betrachtete Knotenzahl $|V| = k = 3$ gezeigt. Da K_3 der Graph ist, der für einen Graph mit $|V| = 3$ die meisten Kanten, nämlich $|E| = 3$ Kanten, besitzt wird dieser betrachtet. Setzt man die Werte dieses Falles in die Ungleichung $|E| \leq 3 \cdot |V| - 6$ ein, so geht diese auf, da

$$3 \leq 3 \cdot 3 - 6.$$

Die Induktionsvoraussetzung ist, dass für jeden triangulierten und planaren Graphen $G = (V, E)$ mit $|V| = k$ gilt

$$|E| \leq 3 \cdot |V| - 6.$$

Aus dieser Voraussetzung muss nun im Induktionsschritt gefolgert werden, dass die Aussage auch für einen beliebigen planaren Graphen mit $k + 1$ Knoten gilt. Allerdings lassen sich nicht alle planaren triangulierten Graphen mit $k + 1$ Knoten aus einem triangulierten Graphen mit k Knoten konstruieren, indem ein Knoten und beliebig viele Kanten, die ihn mit den Knoten des Graphen verbinden, hinzugefügt werden. Deshalb wird von einem beliebigen Graphen $G = (V, E)$ mit $|V| = k + 1$ Knoten ausgegangen und ein beliebiger Knoten $v \in V$ entfernt, sodass ein neuer Graph $G' = (V', E')$ mit $V' = V \setminus \{v\}$ und $E' = E \setminus \{e \in E \mid v \in e\}$ entsteht. Damit die Induktionsvoraussetzung angewendet werden kann, muss G' trianguliert werden. Hierzu wird eine Fallunterscheidung vorgenommen.

1. $d(v) = 3$: In diesem Fall ist der Graph G' bereits trianguliert, da von dem Knoten v in G nur drei Kanten ausgehen und so nach Entfernen des Knotens v die drei ehemaligen Nachbarknoten von v in G' ein Dreieck bilden. Der Graph G' ist daher weiterhin trianguliert, da er immer noch nur aus Dreiecken besteht.
2. $d(v) > 3$: In diesem Fall muss man Kanten ergänzen, um den Graphen zu triangulieren. Durch das Löschen des Knotens v bilden die zu v in G adjazenten¹ Knoten $\{v_1, v_2, \dots, v_{d(v)}\}$ in G' ein $d(v)$ -Eck. Dieses wird im Folgenden so ergänzt, dass sich ein triangulierter Graph $G'' = (V'', E'')$ ergibt. Dabei muss man beachten, dass man keine bereits außerhalb des $d(v)$ -Ecks durch eine Kante $e_{\text{außen}}$ verbundenen Knoten verbindet. Falls es keine solche Kante zwischen zwei Knoten des $d(v)$ -Ecks gibt, so kann von einem beliebigen Punkt v_j aus zu

allen anderen Knoten des $d(v)$ -Ecks bis auf v_j und die zu v_j adjazenten Knoten eine Kante ergänzt werden. Falls es doch mindestens eine solche Kante $e_{\text{außen}}$ gibt, wird eine Kante $e_{\text{außen}} = \{v_k, v_l\}$ betrachtet, für die die beiden Endpunkte $v_k, v_l \in V'$ die Knoten sind, die im $d(v)$ -Eck die kürzeste Distanz zueinander haben. Durch diese spezifische Wahl wird garantiert, dass sich im Pfad innerhalb des $d(v)$ -Ecks zwischen v_k und v_l keine Knoten befinden, die außerhalb des $d(v)$ -Ecks durch eine Kante verbunden sind, da $e_{\text{außen}}$ bereits per Definition die kürzeste solcher Kanten ist. Ebenso können die Knoten, die im $d(v)$ -Eck zwischen v_k und v_l liegen, zu keinem anderen Knoten außerhalb des Pfades durch eine außerhalb des $d(v)$ -Ecks liegende Kante verbunden sein, da ein solcher Weg durch $e_{\text{außen}}$ blockiert wird. Daher kann ein beliebiger Knoten $v_{k \rightarrow l}$ von den Knoten, die im $d(v)$ -Eck zwischen v_k und v_l liegen, mit allen anderen Knoten des $d(v)$ -Ecks, bis auf die beiden Nachbarn und sich selbst, verbunden werden, um den Graphen zu triangulieren.

Für den Graphen $G'' = (V'', E'')$ mit $|V''| = k$ gilt nach Induktionsvoraussetzung $|E''| \leq 3 \cdot |V''| - 6$. Beim Triangulieren werden genau $d(v) - 3$ Kanten ergänzt, da alle Knoten bis auf $v_{k \rightarrow l}$ und seine Nachbarknoten mit $v_{k \rightarrow l}$ verbunden werden. Daher gilt die Beziehung $|E'| + d(v) - 3 = |E''|$. Weiterhin werden hierbei keine Knoten ergänzt oder entfernt, sodass $|V''| = |V'|$ gilt. Somit ergibt sich

$$|E'| + d(v) - 3 \leq |V'| - 6. \quad (1.1)$$

Des Weiteren gilt, dass $|V| = |V'| + 1$, da genau ein Knoten v entfernt wurde. Da beim Entfernen dieses einen Knotens auch alle Kanten, die zu v inzident sind, entfernt werden, gilt auch der Zusammenhang $|E| = |E'| + d(v)$. Setzt man dies in Ungleichung 1.1 ein, so ergibt sich, was zu beweisen war.

$$\begin{aligned} |E| - d(v) + d(v) - 3 &\leq 3 \cdot (|V| - 1) - 6 \\ \Leftrightarrow |E| - 3 &\leq 3 \cdot |V| - 9 \\ \Leftrightarrow |E| &\leq 3 \cdot |V| - 6 \quad \square \end{aligned}$$

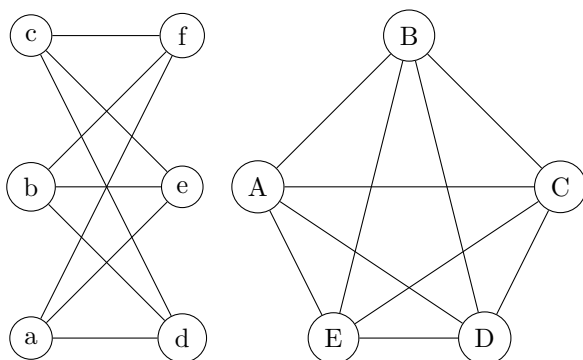
Aus dem nun bewiesenen Lemma ergibt sich eine weitere Eigenschaft planarer Graphen, die im Folgenden kurz erläutert und bewiesen werden soll.

Korollar 1. Sei $G = (V, E)$ ein planarer Graph. Dann gilt für seinen Minimalgrad $\delta(G) \leq 5$.

Beweis. Da von jedem Knoten mindestens $\delta(G)$ Kanten ausgehen und die Summe aller Knotengrade eines Graphen das Doppelte der Kantenanzahl ist, gilt die Beziehung

$$\frac{1}{2} \cdot |V| \cdot \delta(G) \leq |E|.$$

¹benachbart

Abb. 1.8: Die Graphen $K_{3,3}$ (links) und K_5 (rechts)

Zusammen mit Lemma 1 lässt sich schlussfolgern, dass

$$\begin{aligned} \frac{1}{2} \cdot |V| \cdot \delta(G) &\leq |E| \leq 3 \cdot |V| - 6 \\ \Leftrightarrow \delta(G) &\leq 6 - \frac{12}{|V|} \\ \Rightarrow \delta(G) &\leq 5 \quad \square \end{aligned}$$

5.3 Spezielle nicht planare Graphen

Fakt 2. K_5 und $K_{3,3}$ (s. Abb. 1.8) sind nicht planar.

Beweis. Dass K_5 nicht planar ist, folgt direkt aus dem oben bewiesenen Lemma, da die Ungleichung $|E| \leq 3 \cdot |V| - 6$ für die Werte von K_5 nicht gilt.

$$10 \not\leq 3 \cdot 5 - 6$$

In $K_{3,3}$ wird ein Kreis der Länge sechs gesucht und o. B. d. A. dann planar in die Ebene eingebettet. Damit verbleiben noch genau drei Kanten, die es planar einzubetten gilt. Da diese drei Kanten aber die drei langen Diagonalen des Kreises sind, kann nur jeweils eine innerhalb und außerhalb des Kreises planar eingebettet werden. Somit ist in jedem Fall die planare Einbettung mindestens einer Kante und damit des gesamten Graphen nicht möglich. \square

6 Wissenschaft

Annika Dunkel und Josua Kugler

6.1 Was ist Wissenschaft?

Die Wissenschaft kann in drei Hauptbereiche untergliedert werden: Gesellschaftswissenschaften, Naturwissenschaften und Geisteswissenschaften. Ziel der Wissenschaft sind Erkenntnis und Fortschritt, welche durch empirische Experimente o. ä. und Schlussfolgerungen vorangebracht werden. Wichtig für wissenschaftliche Thesen bzw. Theorien ist die Falsifizierbarkeit, da nicht falsifizierbare Thesen nicht zur Erkenntnis beitragen. Das

liegt daran, dass der Wahrheitsgehalt solcher Thesen nicht bestimmt werden kann und sie damit für Wissenschaft, die auf Logik beruht, keinen Nutzen haben. Die Wissenschaft ist also ein zusammenhängendes System von Aussagen, Theorien und Verfahrensweisen. Dieses System wird ständig auf dessen Wahrheitsgehalt hin überprüft und ist mit dem Anspruch objektiver, überpersönlicher Gültigkeiten verbunden.

Der Erfolg der Wissenschaft hängt von der richtigen Kommunikation ab. Beispielsweise wurde 2012 ein möglicher Beweis für eine wichtige mathematische Vermutung, die abc-Vermutung, veröffentlicht, bisher konnte dieser Beweis oder die zugehörige Theorie allerdings nicht nachvollzogen werden.

6.2 Was ist Mathematik?

Es ist schwierig, den Begriff der Mathematik zu definieren, da jeder etwas anderes darunter versteht. Für manche zählen Hypothesen zur Mathematik, für wieder andere besteht die Mathematik nur aus Beweisen und Axiomen. Eine Definition der Mathematik stammt aus dem Jahr 1870 von Charles Sanders Peirce.

Definition 21. *Mathematik ist die Wissenschaft, die notwendige Schlüsse zieht. [?, s. Literaturverzeichnis]*

Saunders MacLane beschreibt Mathematik wie folgt.

Mathematik bezieht sich auf Beweise und Beweise sind ewig. [?, s. Literaturverzeichnis]

Ein weiterer Aspekt der Mathematik ist, dass man Zahlen, Punkte und Funktionen weder hören, schmecken, sehen noch riechen kann.

Auch wenn es Symbole dafür gibt, bleiben die Dinge an sich den menschlichen Sinnen nicht zugänglich. Das einzige, mit dem die Menschen die Mathematik erfassen können, ist der Geist. Die Mathematik beschäftigt sich also mit Objekten, die nicht in der Natur vorkommen, und ist damit eine Geisteswissenschaft.

Die Mathematik fällt Urteile allein durch logische Deduktion (siehe Definition 21). Während z. B. physikalische Aussagen von den äußeren Umständen abhängig sind, gilt eine mathematische Aussage unabhängig davon, wer sie wann trifft. Eine bewiesene Aussage ist in der Mathematik unabhängig vom Beweisansatz immer gültig. Sicherlich dient die Mathematik auch anderen Wissenschaften, jedoch ist sie auch heute noch, wie bereits bei den Griechen, eine eigenständige Wissenschaft, welche unabhängig von ihren Anwendungen betrieben wird.

7 Algorithmen

Philipp Davydov und Petula Diemke

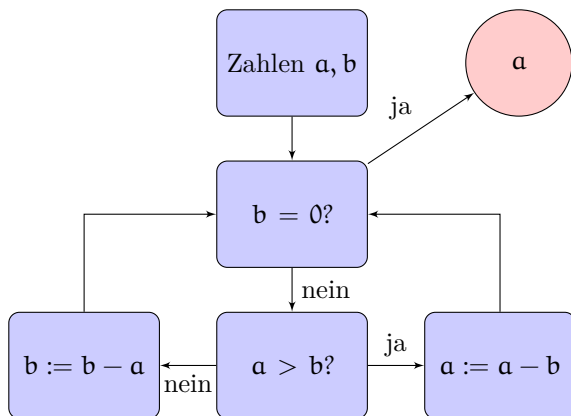


Abb. 1.9: Flowchart für Euklidischen Algorithmus

7.1 Was ist ein Algorithmus?

Ein Algorithmus ist ein Verfahren zur Lösung eines bestimmten Problems. Der Begriff ist in verschiedenen Quellen unterschiedlich definiert worden, aber es existieren dennoch Kriterien, die einen typischen Algorithmus ausmachen. Der Algorithmus selbst ist eine Sequenz von Operationen, die auf einen vorher eingegebenen Input angewendet werden. Diese Sequenz hat das Ziel, ein bestimmtes Ergebnis auszugeben, welches von dem Input abhängt; z. B. ermittelt der Euklidische Algorithmus den größten gemeinsamen Teiler zweier natürlicher Zahlen (siehe Abb. 1.9). Weitere Merkmale sind eine endliche Beschreibung und eine endliche Anzahl der ausgeführten Schritte.

7.2 Wie werden Algorithmen notiert?

Ein Algorithmus kann auf verschiedene Arten schriftlich festgehalten werden. Informelle, aber häufig benutzte Notationen sind die Beschreibung der durchzuführenden Operationen in Worten, als Flowchart (siehe Abb. 1.9) oder mithilfe von anderen bildlichen Darstellungsweisen. Eine formellere Möglichkeit ist der Pseudo-Code (also eine „Programmiersprache in Worten“ die sich nicht auf eine bestimmte Syntax festlegt, siehe Abb. 1.10).

Input: Zwei natürliche Zahlen a und b
 Ist $b = 0$?:
 Wenn ja, gib a aus
 Wenn nein:
 Prüfe, ob $a > b$:
 Wenn ja, setze a auf $(a - b)$
 Wenn nein, setze b auf $(b - a)$
 Kehre zu "Ist $b = 0$?" zurück

Abb. 1.10: Pseudo-Code für Euklidischen Algorithmus

```

def euklid (a,b):
    while True:
        if b == 0:
            return a
        else:
            if a > b:
                a = a - b
            else:
                b = b - a
  
```

Abb. 1.11: Python-Code für Euklidischen Algorithmus

Die formellste und zugleich meistgenutzte Notationsweise ist eine tatsächliche Programmiersprache, z. B. Python (siehe Abb. 1.11), was den Vorteil der (sofortigen) Anwendbarkeit nach sich zieht.

7.3 Was zeichnet einen guten Algorithmus aus?

Ein guter Algorithmus strebt mehrere Ziele an. Eines der wichtigsten Kriterien ist eine möglichst kurze Laufzeit, also eine möglichst schnelle Ermittlung der Lösung. Der Speicherverbrauch wird dabei idealerweise möglichst klein gehalten, da der Speicherplatz in der Praxis immer begrenzt ist. Weiterhin sollte ein guter Algorithmus für andere Entwickler nachvollziehbar sein, damit sie diesen für ihre Zwecke abwandeln oder erweitern können.

7.4 Wie ist die Laufzeit definiert?

Die Laufzeit ist ein wichtiger Parameter eines Algorithmus. Sie hängt von der Größe des Inputs, z. B. in der Graphentheorie von der Anzahl n der Knoten eines Graphen, ab. Dabei kann die Laufzeit anhand ihres Verhaltens bei wachsendem n klassifiziert und in sogenannte Laufzeitklassen einteilt werden. Einige wichtige davon sind:

- $\Theta(n)$ → Die Laufzeit hängt linear vom Input ab.
- $\Theta(n^2)$ → Die Laufzeit hängt quadratisch vom Input ab.
- $\Theta(2^n)$ → Die Laufzeit hängt exponentiell vom Input ab.

Generell gelten Laufzeitklassen, die polynomiell in n sind, als effizient (n ; $2n$; n^2 ; etc.), während exponentielle Laufzeitklassen (2^n) sehr ineffizient sind. Abhängig von der Anwendung wird bei der Klassifizierung entweder vom „average case“ oder, häufiger, vom „worst case“ ausgegangen.

8 Python

Sebastian Leon Schmidt und Florian Weiser

Python² ist eine im Jahr 1991 [?, s. Literaturverzeichnis] veröffentlichte Programmiersprache. Sie gilt als einsteigerfreundlich, unter anderem wegen der dynamischen Typisierung, wodurch Datentypen nicht schon im Code festgelegt werden müssen. Quellcode in Python wird erst während der Laufzeit des Programms vom Interpreter in ausführbaren Computercode verwandelt. Dadurch benötigt Python, verglichen mit kompilierten Programmiersprachen, oft eine längere Laufzeit. Die Entwicklung von Python hat Teile anderer Programmiersprachen wie JavaScript und Swift³ beeinflusst.

8.1 Vorteile

Pythonquellcode hat durch Schlüsselwörter wie **if**, **while** oder **for**, die einfaches Englisch sind, eine gut lesbare Struktur. Durch den einfach lesbaren Quellcode ist Python für Anfänger geeignet, erfahreneren Entwicklenden hilft die Ähnlichkeit der Syntax zu anderen Programmiersprachen. Eine Besonderheit von Python ist, dass Blöcke nicht durch Klammern, sondern durch Einrückungen voneinander getrennt werden.

8.2 Zen of Python

Die Philosophie von Python besagt, dass es genau einen offensichtlichen Weg geben sollte, ein Problem zu lösen. Zudem gibt es unterschiedliche Grundprinzipien wie: „Fehler sollen niemals unbemerkt auftreten“. Dementsprechend sollten unerwartet auftretende Fehler bei der Ausführung des Programms nicht abgefangen, sondern ausgegeben werden, damit die jeweiligen Entwicklenden das Problem einfacher erkennen können.

8.3 Mathematik in Python

Python bietet Befehlsstrukturen für Mathematiker, um mathematische Probleme zu modellieren und zu berechnen. Teile von diesen werden im Folgenden vorgestellt.

8.3.1 Mengen

Das Rechnen mit Mengenoperatoren ist ohne Importe, d. h. man benötigt keine zusätzlichen Module von Python, möglich. Im folgenden Beispiel werden zwei Mengen definiert und deren Vereinigungsmenge gebildet.

```
m = {3, 5, 6, 76}
n = {1, 2, 3, 4, 5}
v = m.union(n)
```

²Der Name basiert auf der britischen Komikergruppe Monty Python.

³Dies ist die Programmiersprache, in welcher Apps für den Apple Appstore programmiert werden

8.3.2 Tupel

Ein Tupel fasst eine bestimmte Anzahl an Elementen zusammen, kann nicht verändert werden und besitzt eine feste Reihenfolge.

Dieses Beispiel `d = (3, 4)` zeigt ein 2 Tupel `d` mit den natürlichen Zahlen 3 und 4 als Elemente. In Python können Tupel folgendermaßen definiert werden.

```
tuple(itertools.combinations(
    '12345', 3))
```

In diesem Beispiel werden alle Kombinationen aus drei Zeichen von den fünf Ziffern als Tupel zurückgegeben.

8.3.3 Generatoren

Ein Generator ist ein Objekt, das eine Reihe von einzelnen Werten oder Objekten hintereinander berechnet und ausgibt. Die ausgegebenen Werte oder Objekte erfüllen vom Entwickelnden definierte Bedingungen. Zum Beispiel kann ein Generator unter der Bedingung erstellt werden, dass die einzelnen Elemente nicht durch eine gewisse Primzahl teilbar sind.

```
gen = (i*i for i in range(20)
        if i % 2 == 0)
```

Die obere Zeile zeigt einen Quellcodeausschnitt, in welchem ein Generator erstellt wird, der mit Hilfe der geraden, unquadrierten Zahlen im Bereich 0 bis 19 die Quadratzahlen errechnet. Anschließend lässt sich mit `next(gen)` der nächste Wert, beginnend mit dem Intervallanfang, zurückgeben.

8.3.4 Dictionaries (Dicts)

Ein Dictionary besteht aus Schlüssel-Wert-Paaren. Dies bedeutet, dass jedes Schlüsselwort mit genau einem Wert oder Objekt verknüpft ist. Durch die eindeutige Zuordnung von Schlüsselwort und Objekt ist ein Dictionary vergleichbar mit einer mathematischen Abbildung. In Python werden Dictionaries wie folgt definiert.

```
d = {"Hans": 76, "Peter": 69}
print(d["Hans"])
```

Die Schlüsselwörter aus diesem Beispiel sind „Hans“ und „Peter“, die auf die Werte 76 und 69 referenzieren (zugehöriges Alter). In der zweiten Zeile wird der mit dem Schlüsselwort „Hans“ verknüpfte Wert 76 ausgegeben.

9 Euler und Hamilton

Theresa Schwarz und Anja Voigt

Im frühen 18. Jahrhundert wurde in der Stadt Königsberg folgendes Problem entwickelt: Die Stadt wird durch einen Fluss in zwei Stadtteile S_1 und S_2 unterteilt. In diesem Fluss gibt es zwei Inseln, die über Brücken miteinander und mit den beiden

Stadtteilen verbunden sind. Wenn wir die beiden Inseln mit A und B bezeichnen, ist A über jeweils zwei Brücken mit beiden Stadtteilen und über eine Brücke mit B verbunden, während B über besagte Brücke mit A und über jeweils eine Brücke mit den beiden Stadtteilen verbunden ist (siehe Abb. 1.12).

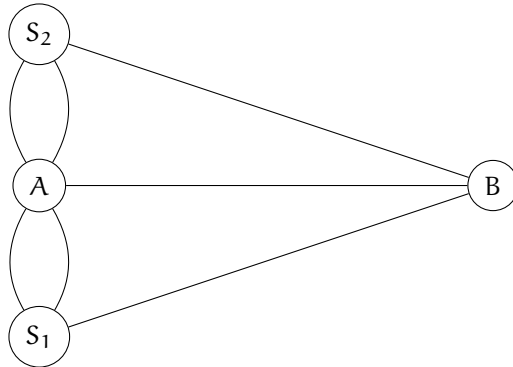


Abb. 1.12: Königsberger Brückenproblem

Die Aufgabe war nun, einen Weg über die Brücken zu finden, sodass jede Brücke genau einmal überquert wurde. Da niemand eine Lösung fand, schien diese Aufgabe nicht lösbar zu sein. Im Jahre 1736 schließlich gelang es Leonhard Euler (1707–1783) zu beweisen, dass es einen solchen Weg nicht gibt. Nach ihm wurde diese Art von Wegen in der Graphentheorie „Eulerweg“ genannt.

Ein ähnliches Problem ist Mitte des neunzehnten Jahrhunderts durch ein von Sir William Rowan Hamilton (1805–1865) erfundenes Spiel namens „Icosian Game“ zum Thema Reiserouten entstanden, wodurch diese Art von Problemen als „Hamiltonweg“ bezeichnet wird.

9.1 Eulerweg und -kreis

Ein **Eulerweg** ist ein Weg, der **alle Kanten** eines Graphen genau einmal abläuft. Eulerwege sind zum Beispiel bei einer Stadtbesichtigung interessant, wenn wir uns nicht für bestimmte Kreuzungen oder Stationen interessieren, sondern für die verschiedenen Straßen. Kurz gesagt besuchen wir jede Straße genau einmal.

Ein **Eulerkreis** ist ein spezieller Eulerweg. Dieser endet im gleichen Knoten, in dem er auch beginnt.

Besitzt ein Graph G einen Eulerkreis, wird er **eulersch** genannt.

Fakt 3. Ein Graph G besitzt genau dann einen Eulerweg, wenn maximal zwei Knoten einen ungeraden Grad haben.

Beweis. Wenn wir über eine Kante an einem Knoten ankommen, müssen wir ihn über eine andere Kante wieder verlassen. Dadurch fallen bei jedem Besuch eines Knotens exakt zwei mögliche Kanten

weg, über die wir den Knoten erreichen oder verlassen können. Wenn am Ende des Durchlaufens aller Knoten alle Kanten abgelaufen sein sollen, müssen alle Knoten eine gerade Anzahl von Kanten besitzen, die von ihm ausgehen – mit Ausnahme des Anfangs- und Endknotens. Der Knoten, an dem wir den Eulerweg beginnen, muss einmal mehr verlassen als betreten werden, während der Endknoten einmal mehr angelaufen als verlassen wird. Daher dürfen diese beiden Knoten einen ungeraden Grad besitzen, während die restlichen Knoten einen geraden Grad haben. \square

Fakt 4. Ein Graph G besitzt genau dann einen Eulerkreis, wenn **alle** Knoten einen geraden Grad haben.

Beweis. Um die Äquivalenz zu beweisen, erklären wir erst die eine Schlussfolgerungsrichtung und anschließend die andere.

Genau wie beim Eulerweg folgt aus der Euler-Eigenschaft, dass wir gerade Grade brauchen. Da wir in diesem Fall allerdings am Startknoten wieder ankommen müssen, müssen diesmal **alle** Knoten einen geraden Grad haben. Sonst landen wir an einem Knoten, den wir nicht wieder verlassen können, und können keinen Kreis schließen.

Anders herum soll aus einem Graphen, welcher nur gerade Grade besitzt, folgen, dass ein Eulerkreis existiert. Wenn wir am Startknoten wieder ankommen, ohne vorher alle Kanten abgelaufen zu haben, markieren wir die abgelaufenen Kanten und suchen uns einen Knoten, welcher sowohl abgelaufene als auch noch nicht besuchte Kanten besitzt. Von dort laufen wir wieder über alle noch nicht markierten Kanten, bis wir an demselben Knoten wieder ankommen und wiederholen diesen Vorgang so lange, bis der ganze Graph abgelaufen ist. Die jeweiligen Startknoten sind die Verbindungsstellen der kleineren Eulerkreise, über welche wir diese zu einem großen Eulerkreis verbinden können. \square

9.2 Hamiltonweg und -kreis

Bei **Hamiltonwegen** läuft man nicht wie in Abschnitt 9.1 alle **Kanten** ab, sondern alle **Knoten**. Jeder Knoten wird genau einmal besucht, aber es muss nicht jede Kante abgelaufen werden.

Analog zum Eulerkreis ist der **Hamiltonkreis** ein spezieller Hamiltonweg, bei welchem der Startgleich dem Endknoten ist.

Besitzt ein Graph G einen Hamiltonkreis, wird er **hamiltonsch** genannt.

9.3 Der Petersen-Graph

Ein besonderer Graph ist der Petersen-Graph, den wir im Folgenden im Zusammenhang mit Euler- sowie Hamiltonkreisen untersuchen werden. Er besitzt 10 Knoten und 15 Kanten, wobei alle Knoten Grad drei besitzen. Weiterhin können wir ihn in

einen äußeren (Fünfeck) und einen inneren Bereich (Stern) unterteilen, welche durch fünf Kanten verbunden sind.

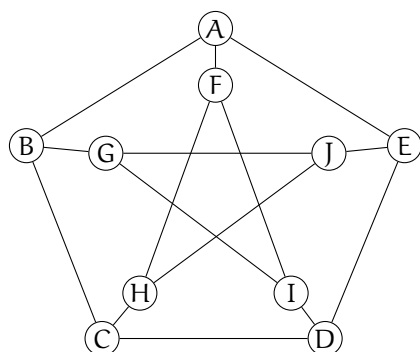


Abb. 1.13: Der Petersen-Graph.

Wenn wir die Grade der Knoten betrachten, können wir schnell feststellen, dass im Petersengraph kein Eulerweg existiert, da mehr als zwei Knoten, nämlich alle zehn, einen ungeraden Grad besitzen.

Existiert ein Hamiltonkreis? Graphentheoretisch sind der innere und der äußere Teil des Graphen gleich, miteinander verbunden besitzen sie jedoch gegensätzliche Arten von Symmetrie: den äußeren Teil könnte man einfach im oder gegen den Uhrzeigersinn ablaufen, im Inneren erreicht man jedoch immer jeden zweiten Knoten, wenn man die Knoten entsprechend der Darstellung im Uhrzeigersinn abläuft. Beim Hamiltonkreis muss der Weg mit demselben Knoten enden, mit dem er auch begonnen hat, und deshalb entweder zweimal oder viermal zwischen dem Inneren und dem Äußeren wechseln.

Im Falle, dass man einmal vom äußeren zum inneren und einmal vom inneren zum äußeren Kreis wechselt, müsste man den jeweiligen Kreis vor dem Wechsel komplett ablaufen. Im inneren Kreis würde man beim Ablaufen des inneren Teilweges an einem anderen Knoten ankommen, als wenn man den Außenkreis abläuft. So können wir den Innenkreis nicht an dem Knoten beenden, der mit dem Anfangsknoten des Außenkreises verbunden ist. Es kann also kein Hamiltonkreis geschlossen werden.

Im Fall von vierfachem Kreiswechsel wird genau eine der Verbindungen zwischen Innen- und Außenkreis nicht abgelaufen. Dies sei o. B. d. A. $\{A, F\}$. Wenn $\{A, F\}$ nicht abgelaufen wird, muss A mit B und E verbunden werden. Da außer $\{A, F\}$ alle Verbindungen zwischen Innen- und Außenkreis abgelaufen werden müssen, ist ein Teilstück des Weges nun $G-B-A-E-J$. Von J und G gibt es nun abgesehen von $\{G, J\}$, die an dieser Stelle eingebaut den Kreis beenden würde, nur noch die Verbindungen zu I und H. Unser bisheriger Weg wäre also $I-G-B-A-E-J-H$. Von I und H aus müssen wir nun wegen des vierfachen Kreiswechsels wieder in den Außen-

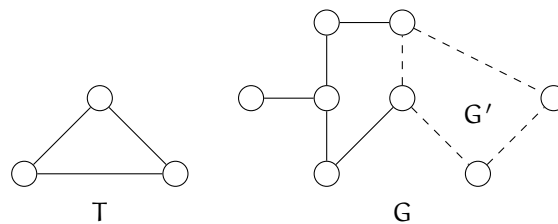


Abb. 1.14: T ist ein topologischer Minor von G

kreis zu C bzw. D wechseln, wodurch der Kreis zwar geschlossen wird, F jedoch kein Teil dieses Kreises ist.

Somit existiert im Petersengraph kein Hamiltonkreis.

10 Planarität

Amelie Koch und Jakob Gierschmann

10.1 Charakterisierung von Planarität

Im Folgenden möchten wir uns genauer damit auseinandersetzen, was genau einen planaren Graphen ausmacht und wie sich diese Klassifikation auf andere räumliche Gebilde übertragen verhält.

Bezüglich der Planarität aus Lemma 1 ist das notwendige Kriterium $|E| \leq 3|V| - 6$ für einen Graphen G bekannt. Darüber hinaus lassen sich jedoch weitere Beobachtungen bezüglich der Planarität aufstellen.

- Jeder Teilgraph eines planaren Graphen ist planar.
- Die Unterteilung von Kanten hat keinen Einfluss auf die Planarität eines Graphen G.

Der letztere Aspekt führt zur folgenden Definition.

Definition 22. Ein Graph T heißt **topologischer Minor** eines Graphen G, falls G einen Teilgraphen G' besitzt, sodass G' durch Kantenunterteilung aus T hervorgeht.

Abbildung 1.14 ist zu entnehmen, dass T ein topologischer Minor von G ist, da sich aus G' so Knoten entfernen und je zwei Kanten zu einer fusionieren lassen, sodass sich G' in T umwandeln ließe. T ist in diesem Beispiel ein C_3 , was in dem Sinne ein besonderer Fall ist, dass jeder Graph G, für den T ein topologischer Minor ist, einen Kreis enthält und umgekehrt (T ist topologischer Minor von $G \Leftrightarrow G$ enthält einen Kreis).

Faktisch geht die Klassifikation planarer Graphen daraus hervor, dass die Graphen K_5 und $K_{3,3}$ selbst nicht planar sind, sowie jegliche andere, die einen davon als topologischen Minor enthalten.

Satz 3 (Kuratowski). Ein Graph ist genau dann planar, wenn er weder K_5 noch $K_{3,3}$ als topologischen Minor enthält.

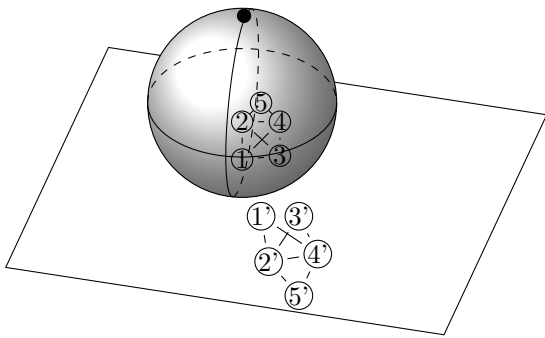


Abb. 1.15: Stereographische Projektion

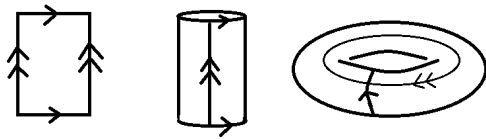
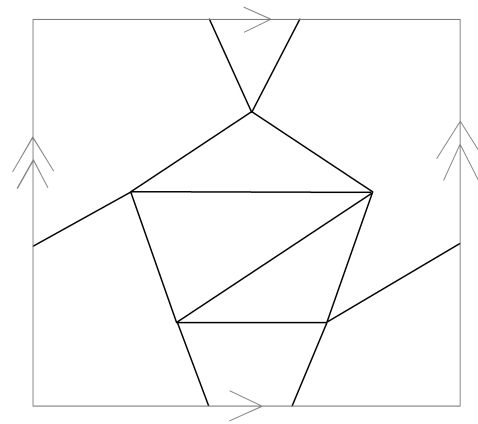


Abb. 1.16: Vom Rechteck zum Torus

10.2 Topologische Erweiterung der überschneidungsfreien Einbettbarkeit

Bisher haben wir Graphen noch nicht im dreidimensionalen Raum betrachtet. Könnte die räumliche Anordnung einen Unterschied in der überschneidungsfreien Einbettbarkeit von Graphen bewirken? Betrachten wir zunächst einen K_4 : Es wird deutlich, dass die Einbettung in eine Sphäre (z. B. die Erde) überschneidungsfrei möglich ist. Unterscheidet sich dieser dann aber von einem K_4 , der in eine Ebene eingebettet wurde? Geometrisch gesehen wäre das eindeutig der Fall – in dem von uns betrachteten Gebiet, der Graphentheorie, jedoch nicht, da über seine **stereographische Projektion** auf eine Ebene der Graph weiterhin planar abgebildet würde. Diese Projektion, wie sie in Abbildung 1.15 zu sehen ist, lässt sich wie folgt veranschaulichen: Eine Lampe, die sich im „Nordpol“ der Sphäre befindet, wirft von innen heraus einen Schatten des Graphen auf die Ebene. Ein typisches Anwendungsbeispiel für diese Projektion ist die Kartografie.

Da die Einbettung in die Sphäre keinen Unterschied zur Ebene bezüglich der überschneidungsfreien Einbettbarkeit eines Graphen macht, betrachten wir nun ein weiteres topologisches Gebilde: einen **Torus** (vergleichbar mit einem Donut). Sowohl Torus als auch Sphäre bezeichnet man als sogenannte **kompakte, orientierbare Mannigfaltigkeiten**, die sich durch die Anzahl ihrer Löcher charakterisieren. Bei einer Sphäre sind dies null, beim Torus eins, beim Doppeltorus zwei, usw. Demnach sind mathematisch betrachtet also auch

Abb. 1.17: K_5

ein Donut und eine Kaffetasse gleiche räumliche Gebilde. Um die Darstellung eines Graphen z. B. auf einem Torus oder einer Sphäre zu vereinfachen, verwendet man **Verklebungen**, eine zweidimensionale Abbildungsmöglichkeit, bei der Schnittkanten den dreidimensionalen Körper zu einer Fläche aufklappbar machen. Das schrittweise Verkleben dieser Schnittkanten von Rechteck zu Torus wird in Abbildung 1.16 veranschaulicht.

Aus dem Satz von Kuratowski leitet sich nun folgende Fragestellung ab: Welche K_n sind auf dem Torus überschneidungsfrei einbettbar? Im Folgenden betrachten wir insbesondere den K_5 als nicht-planaren, vollständigen Graphen auf dem Torus noch einmal genauer. Hierfür bilden wir zunächst den Graphen zweidimensional ab und springen dabei für die Kanten – falls notwendig – an den Schnittkanten der Verklebung zur gegenüberliegenden Seite, wie in Abbildung 1.17 zu sehen ist. Der K_5 ist nach dieser Vorgehensweise auf dem Torus überschneidungsfrei einbettbar und es wird somit deutlich, dass für die überschneidungsfreie Einbettung auf einem Torus der Satz von Kuratowski nicht gilt.

Allgemein ist bekannt, dass sich bei $\lceil \frac{(n-3)(n-4)}{12} \rceil$ Löchern einer Mannigfaltigkeit entsprechend der K_n überschneidungsfrei einbetten lässt. Von $n = 5$ bis einschließlich $n = 7$ liefert diese Formel 1 als Lösung, weshalb für den Torus der K_7 der maximale vollständige Graph ist, der sich überschneidungsfrei einbetten lässt.

11 Beweis des Fünffarbensatzes

Anna Siess und Vincent Meise

Satz 4. Jeder planare Graph ist mit fünf Farben färbbar, sodass keine zwei benachbarte Knoten mit der gleichen Farbe gefärbt sind.

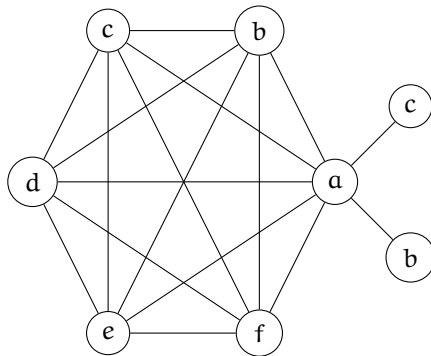


Abb. 1.18: Gegenbeispiel zum Ansatz

Das Problem, einen Graphen mit möglichst wenig Farben zu färben, sodass jeder Knoten anders als all seinen Nachbarknoten gefärbt ist, beschäftigt die Menschen schon lange. Bei planaren Graphen gibt es tatsächlich eine obere Schranke: Nur vier Farben genügen. Dies ist allerdings sehr schwer zu beweisen. Im Folgenden soll daher bewiesen werden, warum man einen planaren Graphen mit nur fünf Farben färben kann.

Ansätze zur Beweisführung:

Für den Beweis wird das Prinzip der Induktion, sowie folgendes Korollar 1 benutzt: $\delta \leq 5$.

11.1 Ansatz: Schluss vom Allgemeinen auf Konkretes

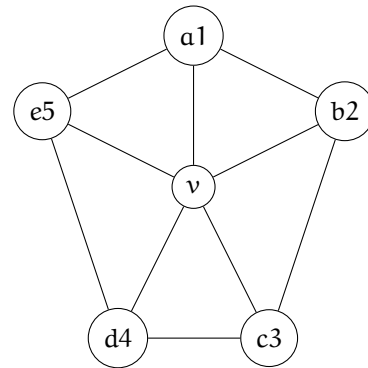
Im ersten Ansatz wird versucht den Satz zu beweisen, indem zuerst bewiesen wird, dass der Satz 4 für alle Graphen gilt, für die Lemma 1 $|E| \leq 3|V| - 6$ gilt, und anschließend auf planare Graphen geschlossen wird. Dafür gibt es allerdings ein Gegenbeispiel, welches diesen Ansatz widerlegt.

Wie man an dem Beispiel in Abbildung 1.18 sieht, gibt es auch nicht-planare Graphen, die Lemma 1 gilt, aber es nicht ausreicht, nur fünf Farben zum Färben zu benutzen. Das liegt daran, dass sechs der acht Knoten untereinander allesamt verbunden sind. Daher muss dieser Ansatz zur Findung eines Beweises aufgegeben werden.

11.2 Ansatz: Umfärben der Knoten

Beweis. Die Grundidee ist, mit einem Worst-Case-Szenario durch Induktion zu zeigen, dass bei einem bereits korrekt gefärbten Graphen ein Knoten entfernt wird und ein neuer Knoten v eingebunden werden kann, sodass der neue Knoten v ebenfalls korrekt gefärbt werden kann. Dies soll durch Umfärben der Knoten möglich gemacht werden. Es werden unterschiedliche Fälle betrachtet, die durch das Lemma 1 begrenzt werden.

Es kann der Fall auftreten, dass der Knoten einen Grad kleiner fünf hat und somit das Einfärben kein Problem ist, da es immer eine Farbe gibt, in der v gefärbt werden kann, ohne dass eine Farbe

Abb. 1.19: Färbungsproblem für Knoten v

einer seiner Nachbarknoten benutzt wird.

Ein weiterer Fall ist, dass Knoten v einen Grad fünf hat, aber die Nachbarknoten so eingefärbt sind, dass für diese nicht alle Farben verwendet werden. Knoten v kann also in einer Farbe gefärbt werden, die nicht für die Nachbarknoten benutzt worden ist.

Der letzte Fall ist, dass Knoten v Grad fünf hat und alle fünf Farben für die Nachbarknoten benutzt worden sind. Dies ist der letzte Fall, der für den Beweis dieses Falles wird das oben erläuterte Lemma 1 benutzen, das es uns ermöglichen soll, ein Worst-Case-Szenario zu erstellen, da fünf die höchste Zahl ist, die laut Lemma 1 in Frage kommt. Hierfür wird der erstmal farblose Knoten v mit der Kantenanzahl 5 untersucht, der mit fünf weiteren Knoten verbunden ist, die paarweise unterschiedlich gefärbt sind – also in insgesamt fünf unterschiedlichen Farben.

Wegen dieser Situation ist es nicht möglich, Knoten v direkt eine Farbe zuzuteilen. Mit diesem durch Korollar 1 erstellte Worst-Case-Szenario soll gezeigt werden, warum es immer möglich ist, einen Knoten v korrekt zu färben, sodass mit keinem benachbartem Knoten die gleiche Farbe hat und auch alle andere benachbarte Knoten nicht die gleiche Farbe haben.

Da alle fünf Farben zum Färben der Nachbarknoten bereits verwendet worden sind, wie man in Abbildung 1.19 sehen kann, kann Knoten v nicht sofort gefärbt werden. Daher wird der Ansatz verwendet, einige Knoten umzufärben, um so für Knoten v eine Farbe zu erhalten, mit der es möglich ist, diesen problemlos einzufärben.

Wird nun a von 1 auf 3, eine Farbe eines Knotens, der mit v verbunden ist aber nicht mit a , umgefärbt, kann Knoten v die Farbe 1 annehmen. Das wird in Abbildung 1.20 dargestellt. Ist Knoten a mit einem Knoten weiteren der Farbe 3 verbunden, so muss dieser andere Knoten von 3 auf 1 umgefärbt werden. Folgt auf a eine ganze Kette aus Knoten, die alle ausschließlich abwechselnd 1 und 3 gefärbt sind und mit Knoten a benachbart sind, so müssen die Knoten mit Farbe 1 auf Farbe

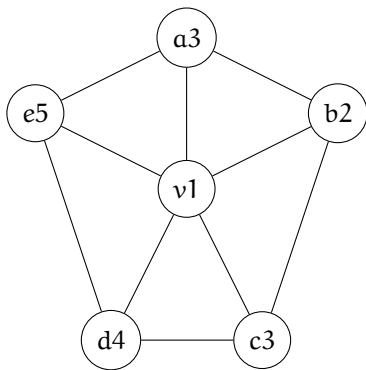


Abb. 1.20: Gewünschte Lösung des Problems

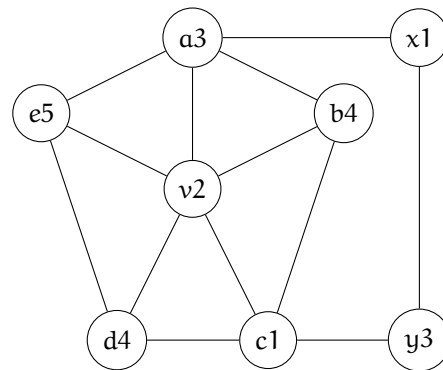


Abb. 1.22: Neues Umfärben ohne Probleme

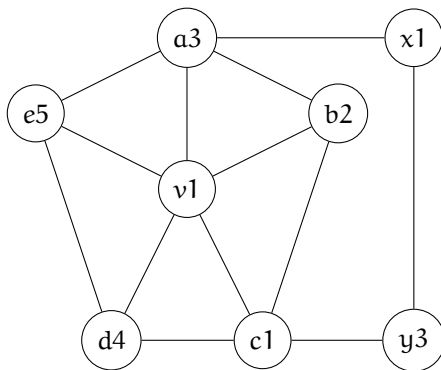


Abb. 1.21: Problem durch Umfärbung

3 umgefärbt werden und umgekehrt. So wird erreicht, dass trotz des Umfärbens alle bereits vorhandenen und gefärbten Knoten korrekt gefärbt sind. Im Folgenden wird ein solcher Kantenzug Umfärbungskantenzug genannt werden.

Nun wird eine Fallunterscheidung vorgenommen. Im ersten Fall ist Knoten a durch einen oben genannten Umfärbungskantenzug mit Knoten c verbunden und somit wird Knoten c von Farbe 3 auf Farbe 1 umgefärbt. Dadurch hat Knoten v allerdings dieselbe Farbe wie Knoten c und die Bedingung, dass zwei miteinander verbundene Knoten nicht mit der gleichen Farbe gefärbt sein dürfen, wäre nicht erfüllt. Dies wird in Abbildung 1.21 gezeigt. Es muss also eine andere Möglichkeit gefunden werden, Knoten v korrekt zu färben. Hierfür wird erst einmal die durch die Kette aus mit den Farben 1 und 3 gefärbten Knoten von Knoten a zu Knoten c gegebene Situation untersucht.

Durch den beliebig langen Umfärbungskantenzug aus nur Knoten mit Farben 1 und 3 wird der Knoten b mit der Farbe 2, sowie die eventuell mit ihm verbundenen Knoten isoliert.

Knoten b wird mit Farbe 4 umgefärbt. Wieder gilt, dass alle Knoten, die von Knoten b aus über einen ausschließlich abwechselnd aus Knoten mit Farbe 2 und aus Farbe 4 bestehende Kantenzug erreichbar sind, von Farbe 2 auf Farbe 4 umgefärbt werden und umgekehrt. Jeder solch Kantenzug von Knoten

b aus kann aufgrund des Umfärbungskantenzuges von a nach c nicht mit d oder e verbunden sein. Somit kann für Knoten v problemlos die Farbe 2 benutzt werden, wie man in Abbildung 1.22 sehen kann.

Im zweiten Fall ist Knoten a nicht durch einen Umfärbungskantenzug mit Knoten c verbunden und somit hat Knoten v die Farbe 1 a , ohne dass weitere Umfärbungen notwendig sind. Wieder gilt, dass alle Knoten, die über einen Kantenzug aus abwechselnd Knoten mit Farbe 1 oder 3 mit a verbunden sind, von 1 auf 3 umgefärbt werden und umgekehrt. Knoten v hat somit eine Möglichkeit der Färbung, nämlich Farbe 1.

Durch das Beweisen dieses Worst-Case-Szenario kann bewiesen werden, dass der Satz 4 für alle planare Graphen gilt, da jedes mal nicht nur Knoten v betrachtet wird, sondern automatisch der gesamte Graph, der gegebenenfalls durch Umgefärbungen angepasst wird. Zudem wird durch die Umfärbungskantenzügen sichergestellt, dass keine neue problematische Knoten entstehen.

□

12 Isomorphismus

Florian Weiser und Sebastian Leon Schmidt

12.1 Eigenschaften von Funktionen

Oftmals ist das Umkehren einer Abbildung⁴ nur mit Einschränkungen der Definitions- oder Wertemenge möglich. Im Folgenden sollen Eigenschaften definiert werden, welche bereits Rückschlüsse auf die Umkehrbarkeit von Abbildungen erlauben.

12.1.1 Injektive Abbildungen

Definition 23. Eine *injektive Abbildung* (Injektion) ist eine Abbildung $f: A \rightarrow B$, durch die zu jedem Element $b \in B$ höchstens ein Element $a \in A$ mit $b = f(a)$ existiert.

⁴Abbildungen sind eindeutige Zuordnungen zwischen zwei Mengen D und Z . Jedem Element $x \in D$ wird durch die Abbildung f genau ein Element $f(x) \in Z$ zugeordnet. Der Begriff Funktion wird meist synonym zu Abbildung benutzt.

Keine zwei verschiedenen Elemente der Definitionsmenge dürfen auf das selbe Element der Zielmenge abbilden. In Abbildung 1.23 ist eine injektive Abbildung zwischen den Mengen A und B visualisiert. Die Umkehrung einer injektiven Funktion ist im Bereich der tatsächlich angenommenen Werte möglich.

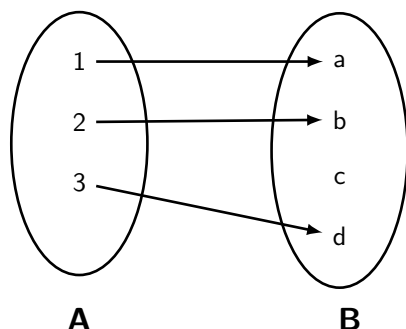


Abb. 1.23: Illustration einer injektiven Abbildung

12.1.2 Surjektive Abbildungen

Definition 24. Eine **surjektive Abbildung** (Surjektion) ist eine Abbildung $f : A \rightarrow B$, durch die jedes $b \in B$ mindestens ein Urbild in der Menge A besitzt.

Abbildung 1.24 veranschaulicht eine surjektive Abbildung von A auf B. Surjektive Abbildungen (Surjektionen) sind nicht notwendigerweise eindeutig umkehrbar, weil einige Elemente aus B mehrere Urbilder besitzen können.

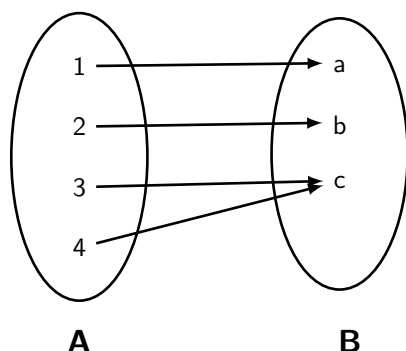


Abb. 1.24: Illustration einer surjektiven Abbildung

12.1.3 Bijektive Abbildungen

Definition 25. Eine **bijektive Abbildung** (Bijektion) ist eine Abbildung $f : A \rightarrow B$, die sowohl injektiv als auch surjektiv ist.

Zwischen der Definitionsmenge und Bildmenge einer bijektiven Funktion besteht eine Eins-zu-Eins-Beziehung. Jedem Element aus der Menge A wird dabei exakt ein Element aus der Menge B

zugeordnet und alle Elemente aus B besitzen genau ein Urbild in A. Dieses Verhältnis ist in Abbildung 1.25 veranschaulicht. Man spricht von einer eindeutigen Zuordnung. Beispielsweise ist die Funktion $g(x) = x$ bijektiv im Bereich der reellen Zahlen. Die Umkehrabbildung $f^{-1} : Y \rightarrow X$ einer bijektiven Funktion f mit $y \mapsto f^{-1}(y)$ kann ohne Einschränkungen gebildet werden.

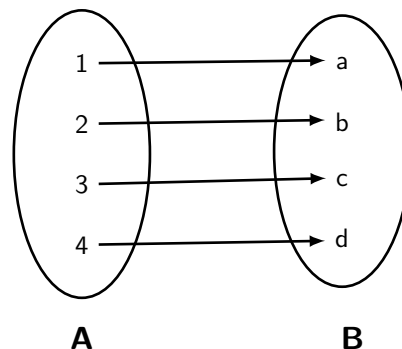


Abb. 1.25: Illustration einer bijektiven Abbildung

12.2 Isomorphismus

Definition 26. Eine Abbildung $f : A \rightarrow B$ ist ein **Isomorphismus**, wenn sie bijektiv und strukturerhaltend ist.

So ist etwa die Abbildung h von \mathbb{N}_0 auf \mathbb{Z} mit $h : n \mapsto (-1)^n \lceil \frac{n}{2} \rceil$ bijektiv, aber kein Isomorphismus, da die Struktur der Addition von Elementen nicht erhalten bleibt und zum Beispiel $h(2 + 3) \neq h(2) + h(3)$.

12.3 Exkurs: Homotope Abbildungen

Definition 27. Sei I das Einheitsintervall $[0, 1]$ und seien $f, g : X \rightarrow Y$ zwei stetige Abbildungen zwischen den topologischen Räumen X und Y . Eine **Homotopie** ist eine stetige Abbildung $H : X \times I \rightarrow Y$, welche die Eigenschaften

$$H(x, 0) = f(x) \quad (1.2)$$

und

$$H(x, 1) = g(x) \quad (1.3)$$

erfüllt.

Die Homotopie H kann sich als Verformung einer Abbildung in eine andere vorgestellt werden. Zwei Abbildungen sind homotop zueinander, wenn eine Homotopie für sie existiert.

Beispiel: Sei S^1 eine 1-dimensionale Sphäre⁵ und T^2 ein 2-dimensionaler Torus. In Abbildung 1.26 sind die Abbildungen $a, b, c : S^1 \rightarrow T^2$ im topologischen Raum des Torus eingezeichnet. Die Abbildungen a und c sind homotop zueinander, weil sich

⁵Eine 1-dimensionale Sphäre wird auch als Kreislinie bezeichnet

die eine Abbildung durch Verformung der anderen auf der Oberfläche des Torus erreichen lässt. Da die eingezeichneten Abbildungen außer b homotop zu einem Punkt sind, gibt es keine Homotopie zwischen b und einer der beiden anderen Abbildung.

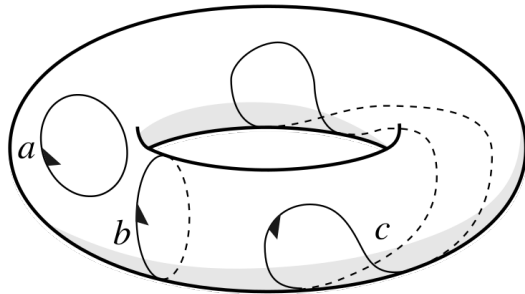


Abb. 1.26: Kreislinien a , b und c auf der Oberfläche eines Torus (Bildnachweis: <https://en.wikipedia.org/wiki/File:Toruscycles1.svg> von Steelpillow; Lizenz: Creative Commons Attribution-Share Alike 4.0i; Kreislinie a neu platziert)

12.4 Elementlose Isomorphismusdefinition

Der Exkurs zu homotopen Abbildungen soll als Motivation für eine elementlose Definition von Isomorphismus dienen.

Definition 28. Eine Abbildung $f : A \rightarrow B$ ist ein Isomorphismus, wenn es eine Abbildung $g : B \rightarrow A$ gibt, sodass $g \circ f = \text{id}_A$ und $f \circ g = \text{id}_B$.

12.5 Isomorphe Graphen

Die elementlose Definition von Isomorphismus kann auch auf Graphen angewandt werden.

Definition 29. Sei p eine bijektive Abbildung $p : V \rightarrow V'$ mit $G = (V, E)$ und $G' = (V', E')$. Die Abbildung p ist ein **Isomorphismus**, falls $(u, v) \in E \Leftrightarrow (p(u), p(v)) \in E'$ gilt.

Zwei isomorphe Graphen müssen die gleiche Knoten- und Kantenmenge besitzen. Sie können so gezeichnet werden, dass sie identisch aussehen. Für den Nachweis eines Isomorphismus ist es oftmals nützlich, den Knoten Nummern zuzuordnen und zu prüfen, ob die von einem Knoten ausgehenden Kanten in beiden Graphen übereinstimmen. Abbildung 1.27 zeigt zwei isomorphe Graphen mit entsprechend nummerierten Knoten.

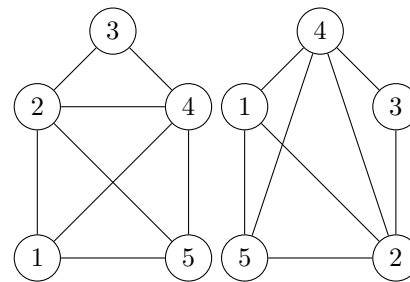


Abb. 1.27: Isomorphe Graphen

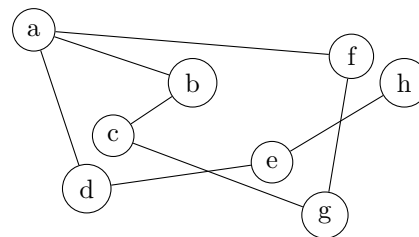


Abb. 1.28: Graph H

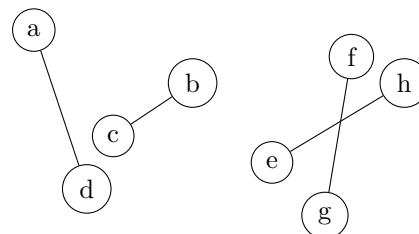


Abb. 1.29: Matching M_1 in Graph H

Graphen vorhandene Kante miteinander gematcht werden.

Definition 30. Ein **Matching** ist eine Untermenge M der Kantenmenge E , sodass für alle $e, e' \in M$ mit $e \neq e'$ gilt: $e \cap e' = \emptyset$.

Das heißt, dass jeder Knoten maximal in einer Kante der Kantenmenge M vorhanden ist. Somit hat jeder Knoten im Matching M höchstens einen Partner.

13.2 Eigenschaften von Matchings

Es gibt verschiedene Möglichkeiten ein Matching M in einem Graphen G zu bilden. Gibt es keine Möglichkeit M noch Kanten hinzuzufügen, wird das Matching als maximal bezeichnet. Es werden zwei Arten von maximalen Matchings unterschieden.

Definition 31. Ein Matching M heißt **kardinalitätsmaximal**, wenn es kein Matching M' gibt, sodass $|M'| > |M|$.

Diese Definition besagt, dass im Graph G existiert anderes Matching M' existiert, welches mehr Knoten verbindet und somit mehr Kanten als ein kardinalitätsmaximales Matching M besitzt. Das

13 Matchings

Petula Diemke und Theresa Schwarz

13.1 Was ist ein Matching?

Ein Matching oder eine Paarung liegt vor, wenn Knoten eines Graphen jeweils paarweise anderen Knoten des Graphen zugeordnet werden. Es können jeweils nur zwei Knoten über eine schon im

Matching M_1 (Abb. 1.29) ist zum Beispiel ein kardinalitätsmaximales Matching in Graph H.

Die zweite Möglichkeit eines maximalen Matchings ist ein inklusionsmaximales Matching M . Liegt dieses vor, kann einem Matching M in Graph G keine Kante mehr hinzugefügt werden, allerdings muss dieses nicht insgesamt in G das Matching mit der größten Kantenanzahl sein.

Definition 32. Ein Matching M heißt **inklusionsmaximal**, wenn kein Matching M' existiert, sodass $M \subsetneq M'$.

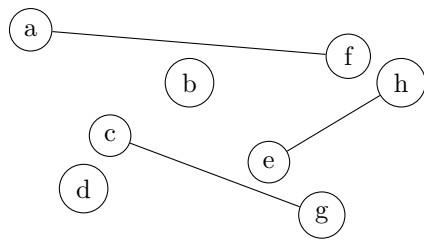


Abb. 1.30: Matching M_2 in Graph H

Das Matching M_2 (Abb. 1.30) ist ein Beispiel für ein inklusionsmaximales Matching in Graph H. Bei dieser Konstellation sind keine weitere Verbindungen zwischen Knoten möglich, es gibt in H aber ein anderes Matching M_1 , welches mehr Kanten enthält (Abb. 1.29). Somit ist M_2 nicht kardinalitätsmaximal.

Ist jeder Knoten in M enthalten, wird das Matching als perfekt bezeichnet. Da jeweils zwei Knoten durch eine Kante aus M verbunden sind, ist die Anzahl der Kanten in M genau die Hälfte der Anzahl der Knoten. Das Matching M_1 aus Abb. 1.29 ist ein Beispiel für ein perfektes Matching.

Definition 33. Ein Matching M heißt **perfekt**, wenn $|M| = \frac{1}{2}|V|$.

Wenn ein Graph G eine ungerade Anzahl von Knoten besitzt, dann gibt es kein perfektes Matching, da immer ein Knoten existiert, der nicht mit einem anderen Knoten gematcht ist. Sind aber ansonsten alle Knoten gepaart, dann wird dieses Matching als fast-perfekt bezeichnet. Somit entspricht die Anzahl der Kanten in M der Hälfte der um eins reduzierten Knotenanzahl.

Definition 34. Ein Matching M heißt **fast-perfekt**, wenn $|M| = \frac{1}{2}(|V| - 1)$.

13.3 Stabiles Matching

Für den weiteren Verlauf des Kapitels wird angenommen, dass jeder Graph G ein bipartiter Graph ist und jeder Knoten in diesem Graphen eine Präferenzliste hat.

$X = \{1, 2, 3, 4\}$	$Y = \{A, B, C, D\}$
1 — A B C D	A — 4 2 1 3
2 — D B A C	B — 3 1 4 2
3 — B A C D	C — 1 3 4 2
4 — D A B C	D — 1 4 2 3

Tab. 1.2: Knoten in Menge X und Y mit Präferenzliste

In einem **bipartiten Graphen** ist eine Einteilung der Knoten in zwei Mengen X und Y möglich, sodass innerhalb einer jeden Menge die Knoten nicht miteinander verbunden sind. Die einzigen möglichen Kanten existieren zwischen zwei Knoten unterschiedlicher Mengen.

Eine **Präferenzliste** ist eine Liste eines Knoten einer Menge, in der alle Knoten der anderen Menge vorhanden und nach Präferenz geordnet sind.

In Tab. 1.2 ist beispielsweise eine konkrete Instanz für einen bipartiten Graphen G mit den Knotenmengen $X = \{1, 2, 3, 4\}$ und $Y = \{A, B, C, D\}$ mit den jeweiligen Präferenzlisten abgebildet.

Ein Matching M in einem bipartiten Graphen G besteht immer aus Kanten zwischen jeweils einem Knoten der einen und einem Knoten der anderen Menge (welche in dieser Betrachtung beide aus der gleichen Anzahl Knoten bestehen). In einem bipartiten Graphen G mit Präferenzliste der Knoten wird meist versucht, ein stabiles Matching M zu erstellen.

Definition 35. Ein Matching M heißt **stabil**, wenn für jedes Paar (x, y) mit $x \in X$, $y \in Y$ und $\{x, y\} \notin M$ gilt, dass x seinen Partner gegenüber y präferiert oder y seinen Partner gegenüber x präferiert.

x und y sind Knoten unterschiedlicher Mengen, die nicht miteinander gematcht sind. Bevorzugt x seinen derzeitigen Partner gegenüber y oder bevorzugt y seinen derzeitigen Partner gegenüber x , dann ist das Matching stabil, da mindestens einer von x oder y nicht von seinem derzeitigen Partner getrennt und miteinander gematcht werden wollen würde. Wären sie gegenseitig weiter vorn auf ihren Präferenzlisten, würden x und y präferieren, miteinander gematcht zu werden und das momentane Matching M wäre nicht stabil.

13.4 Ermitteln eines stabilen Matchings

Für das Ermitteln eines stabilen Matchings M eines bipartiten Graphen G gibt es verschiedene Algorithmen. Der derzeit Beste ist der **Gale-Shapely-Algorithmus**.

1. Alle $x \in X$ stellen Antrag an präferiertes $y \in Y$.
2. Jedes $y \in Y$, das Antrag erhalten hat, akzeptiert den von sich selbst bevorzugten.

3. Nicht-gematchte $x \in X$ stellen Antrag an nächstbestes $y \in Y$.
Kehre zu Schritt 2 zurück.
4. Wenn alle $x \in X$ gematcht ist der Algorithmus zu Ende.

Ein bekanntes Beispiel für ein stabiles-Matching-Problem ist das Heiratsproblem. Es gibt eine bestimmte Anzahl Herren und genauso viele Damen, die sich verloben möchten. Nun hat jede der Damen eine Präferenzliste mit der Reihenfolge, in der sie die Herren bevorzugen. Die Herren haben auch jeweils eine Präferenzliste, welche wiederum angibt, welche Damen von welchem Herren präferiert werden. Jetzt wird ein stabiles Matching M gesucht.

Als Beispiel wird Tab. 1.2 betrachtet. 1, 2, 3 und 4 sind die Knoten, die die Herren darstellen; A, B, C und D sind die Knoten, die die Damen repräsentieren. Die Herren machen Anträge an die Damen in der Reihenfolge ihrer Präferenzliste, welche in der Abbildung als Aufzählung (der Knoten der anderen Menge) hinter den Knoten dargestellt ist. Wird zu diesem Beispiel der Gale-Shapely-Algorithmus ausgeführt, dann geschieht folgendes

1 macht A einen Antrag
 2 macht D einen Antrag
 3 macht B einen Antrag
 4 macht D einen Antrag
 A nimmt Antrag von 1 an
 B nimmt Antrag von 3 an
 D nimmt Antrag von 4 an, da 4 präferierter
 2 macht B einen Antrag
 B lehnt 2 ab, da 3 präferierter
 2 macht A einen Antrag
 A nimmt Antrag von 2 an, da 2 präferierter
 1 macht B einen Antrag
 B lehnt Antrag von 1 ab, da 3 präferierter
 1 macht C einen Antrag
 C nimmt Antrag von 1 an
 → stabiles Matching

X						Y					
1	—	A	B	<u>C</u>	D	A	—	4	<u>2</u>	1	3
2	—	D	B	<u>A</u>	C	B	—	<u>3</u>	1	4	2
3	—	<u>B</u>	A	C	D	C	—	<u>1</u>	3	4	2
4	—	<u>D</u>	A	B	C	D	—	1	<u>4</u>	2	3

Tab. 1.3: Knoten mit stabilem Matching M

Das vom Gale-Shapely-Algorithmus ausgegebene Matching M ist immer stabil: Angenommen Herr x präferiert Dame y gegenüber seiner aktuellen Partnerin, dann hätte er y irgendwann einen Antrag gemacht. Da sich im Laufe des Algorithmus die Damen dauerhaft verbessern, muss der Herr x weiter hinten auf ihrer Präferenzliste stehen. Ansonsten hätte sie den Antrag angenommen und es käme

gar nicht erst zu einem instabilen Matching. Auch bei den Damen besteht nicht die Chance, dass eine von ihnen einen anderen Herren bevorzugt, welcher lieber ihr den Antrag gestellt hätte als seiner derzeitigen Partnerin.

14 Approximationsalgorithmen

Anja Voigt und Janek Paßens

Die verschiedenen Problemstellungen, die man mit Hilfe von Algorithmen lösen kann, lassen sich im Wesentlichen in die folgenden drei Kategorien einordnen.

1. **Entscheidungsprobleme:** Ausgabe „ja“ oder „nein“
2. **Funktionsprobleme:** Ausgabe ist oft ein Objekt, bspw. ein Graph mit bestimmten Eigenschaften
3. **Optimierungsprobleme:** Aus einer Menge von zulässigen Lösungen soll die beste ausgewählt werden, das heißt, sie soll je nach Kontext möglichst groß (Maximierungsproblem) oder möglichst klein (Minimierungsproblem) sein.

Approximationsalgorithmen lösen Probleme der dritten Kategorie. Sie versuchen möglichst gute Lösungen für ein Problem zu finden, die nicht immer optimal sind, aber möglichst nah an die optimale Lösung gelangen. Je nachdem, ob der Approximationsalgorithmus ein Maximierungs- oder Minimierungsproblem lösen soll, gibt es verschiedene untere oder obere Schranken, für die Ausgabe des Approximationsalgorithmus.

Bei Maximierungsproblemen die durch einen Approximationsalgorithmus bearbeitet werden, sollen die Ausgaben mindestens so groß sein, wie die optimale Lösung geteilt durch den Approximationsfaktor, der angibt, wie nah man an der optimalen Lösung ist. Je kleiner dieser Faktor k ist, desto besser ist die Lösung. Die Ausgabe x soll also folgender Ungleichung genügen $x \geq \frac{\text{OPT}}{k}$, wobei OPT die Größe der optimalen Lösung bezeichnet. Für k gilt $k \geq 1$.

Bei Minimierungsproblemen genügen alle möglichen Ausgaben x der Ungleichung $x \leq k \cdot \text{OPT}$, wobei OPT und k die gleichen Bedingungen erfüllen wie bei den Maximierungsproblemen.

14.1 Ein Greedy-Algorithmus zur Bestimmung eines inklusionsmaximalen Matchings

Folgend soll ein Algorithmus vorgestellt werden, der ein inklusionsmaximales Matching in einem Graphen findet. Dieser Algorithmus gehört zur Klasse der Greedy-Algorithmen. Dabei wird jeweils eine Kante e dem Matching hinzugefügt. Anschließend werden alle zu e benachbarten Kanten aus der

Menge aller möglichen Kanten entfernt. Dies wird nun für die Menge aller verbleibenden Kanten so oft wiederholt, bis keine mögliche Kante mehr übrig ist. So wird ein inklusionsmaximales Matching erzeugt, da am Schluss dem Matching keine Kante mehr hinzugefügt werden kann.

Definition 36. *Der minimale Spannbaum $G' = (V', E')$ eines gewichteten, zusammenhängenden Graphen $G = (V, E)$, ist ein Baum, für den $V = V'$ und $E' \subset E$ gilt und in dem die Summe aller Kantengewichte minimal ist.*

14.2 Das Travelling Salesman Problem

Das **Travelling Salesman Problem**, kurz TSP, bezeichnet ein Minimierungsproblem, bei dem für einen Handelsreisenden, der n Städte besuchen und am Ende wieder an seinem Ausgangspunkt ankommen will, die kürzeste Route gesucht wird. Einfacher ausgedrückt wird in einem vollständigen, mit der Funktion $w : E \rightarrow \mathbb{R}_{>0}$ kanten-gewichteten, Graphen $G = (V, E)$ der Hamiltonkreis mit geringstem Gesamtkantengewicht gesucht. Für unsere Überlegungen betrachten wir einen Spezialfall des TSP, das sogenannte metrische Travelling Salesman Problem. Hierfür nehmen wir den Graphen als metrisch an, das heißt, die Gewichte der Kanten jedes Dreiecks $\{v_1, v_2, v_3\}$ mit $v_1, v_2, v_3 \in V$ genügen der Dreiecksungleichungen $w(\{v_1, v_2\}) \leq w(\{v_1, v_3\}) + w(\{v_2, v_3\})$. Zunächst berechnet der Algorithmus den minimalen Spannbaum des Graphen G . Um in diesem nun eine Route zu finden, in der alle Knoten enthalten sind, muss man alle Kanten des minimalen Spannbaums doppelt ablaufen. Diese Route ist die Ausgabe des Algorithmus. Dieser löst das metrische Travelling Salesman Problem mit Approximationsfaktor $k = 2$. Hierzu sei H folgend der optimale Hamiltonkreis des Graphen G . Ausgehend von diesem wird genau eine Kante entfernt, sodass ein Baum $B = (V_B, E_B)$ entsteht. Dieser Baum besitzt mindestens die gleiche Gesamtkantengewicht wie der minimale Spannbaum $S = (V_S, E_S)$ des Graphen G , sodass gilt

$$\sum_{e \in E_B} w(e) \geq \sum_{e \in E_S} w(e).$$

Da weiterhin die optimale Lösung H den Spannbaum B enthält, gilt

$$\sum_{e \in H} w(e) \geq \sum_{e \in E_S} w(e).$$

Um auf S eine Route T zu laufen, muss man hier jede Kante genau zweimal ablaufen, um zum Anfangsknoten zurückzukehren. Daher gilt

$$\sum_{e \in T} w(e) = 2 \sum_{e \in E_S} w(e).$$

Wenn die beiden Ungleichungen kombiniert werden, ergibt sich

$$\sum_{e \in T} w(e) \leq 2 \sum_{e \in H} w(e).$$

Mit diesem Algorithmus erreicht man also den Approximationsfaktor von $k = 2$.

14.3 Algorithmus von Christofides

Der Algorithmus nach Christofides liefert für einen vollständigen metrischen Graphen $G_0 = (V, E)$, der mit der Funktion $w : E \rightarrow \mathbb{R}_{>0}$ kanten-gewichtet ist, eine noch bessere Annäherung an das Optimum. Dieser soll folgend erläutert werden.

1. Finde einen minimalen Spannbaum T von G_0 .
2. Finde ein kostenminimales perfektes Matching M des von den Knoten mit ungeradem Grad induzierten Teilgraphen U . Dies ist möglich, weil die Summe aller Grade der Knoten eines Graphen gerade ist, wodurch es eine gerade Anzahl von Knoten ungeraden Grades gibt.
3. Kombiniere M und T zu einem gemeinsamen Graphen G . Dieser ist dann eulersch, weil durch das in Schritt 2 gefundene matching alle Grade gerade gemacht werden.
4. Finde einen Eulerkreis C in G .
5. Transformiere C in einen Hamiltonkreis H , indem man von einem beliebigen Knoten ausgehend den Eulerweg abläuft. Die bereits abgelaufenen Knoten werden dann jedoch übersprungen, indem man die Kante vom Vorgänger des jeweiligen Knotens zu dem Knoten abläuft, der darauf folgen würde. Da der Graph metrisch ist, ist diese Kante maximal so groß wie die Summe der Gewichte der übersprungenen Kanten.

Folgend sei $w(P)$ für eine Kantenmenge P die Summe der Kantengewichte der Kanten in P . Sei nun H' der optimale Hamiltonweg. Es gilt $w(H) \leq w(C)$, da das Gesamtkantengewicht des Hamiltonkreises stets maximal so groß ist wie das Gesamtkantengewicht des Eulerkreises. Weiterhin ist $w(C) = w(T) + w(M)$, da C die Kombination von M und T ist. Des Weiteren ist $w(T) \leq w(H')$, weil H' stets einen Spannbaum enthält und daher mindestens so groß ist wie der minimale Spannbaum.

Seien nun M' und M'' beliebige, disjunkt perfekte Matchings des von den Knoten mit ungeradem Grad induzierten Teilgraphen U von T , dann gilt $w(M') \geq w(M)$ und $w(M'') \geq w(M)$, weil M ein perfektes, kostenminimales Matching ist. A sei im Folgenden der Kreis, der alle Knoten aus $M' \cup M''$ in der Reihenfolge abläuft, in der sie auf dem perfekten Hamiltonweg H' liegen. Wegen der

Dreiecksungleichung gilt nun $w(A) \leq w(H')$. Da die Anzahl aller Knoten mit ungeradem Grad gerade ist, kann sich aus den Matchings M' und M'' immer A ergeben. Daher gilt

$$2w(M) \leq w(M') + w(M'').$$

Da $w(M') + w(M'') = w(A)$ ist, gilt weiterhin

$$2w(M) \leq w(A).$$

Daher gilt

$$w(M) \leq \frac{1}{2}w(A) \leq \frac{1}{2}w(H').$$

Deshalb ergibt sich

$$w(H) \leq w(T) + w(M) \leq \frac{3}{2}w(H').$$

Somit hat der Algorithmus von Christofides den Approximationsfaktor $\frac{3}{2}$.

15 Programmierprojekt

Vincent Meise und Felix Forner

Einen großen Teil der Kurszeit verbrachten wir mit einer Projektarbeit, die das Schreiben von Programmen in Python zur Lösung graphentheoretischer Probleme umfasste. Dabei gaben verschiedene Texte Denkanstöße und Ideen für die Themenfindung. Besonders war dabei aber, dass keine konkreten Aufgabenstellungen vorgegeben wurden. Stattdessen generierte jede Gruppe selbstständig eine angemessene Aufgabe zu der Thematik des gewünschten Textes.

Bei den Problemen war es zuerst erforderlich, sich zu überlegen, wie sie graphentheoretisch dargestellt werden können. Um den Erfolg des Projekts zu sichern, sollten einige Überlegungen in einer Vorbereitungsphase vor der Implementierung getätigt werden. In diesem Rahmen haben wir zuerst die Ein- und Ausgabe spezifiziert. Außerdem brauchten wir eine Erfolgsmetrik, um zu prüfen, ob das Programm wie gewünscht funktioniert. Um Chaos während der Entwicklung zu vermeiden, entwarfen wir einen Entwicklungsplan, der das Problem in einzelne Komponenten unterteilt. Neben dem folgenden Beispielprojekt wurden in anderen Gruppen verschiedene Anwendungen des Dijkstra-Algorithmus⁶, eine graphentheoretische Variante der Kryptographie und weitere Themen behandelt.

15.1 Beispielprojekt

15.1.1 Aufgabe

Es soll ein Algorithmus programmiert werden, der prüft, ob zwei eingegebene Graphen isomorph sind oder nicht und entsprechend einen der Wahrheitswerte „True“ oder „False“ ausgibt.

⁶Der Dijkstra-Algorithmus bestimmt den kürzesten Weg zu jedem Knoten in einem Graphen ausgehend von einem Startknoten.

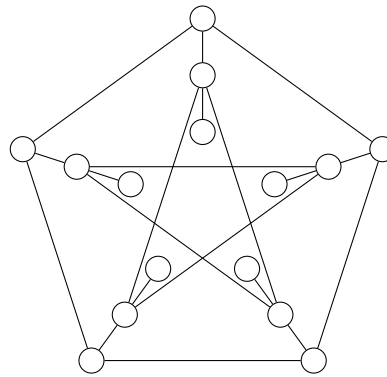


Abb. 1.31: Graph G_1

15.1.2 Umsetzung

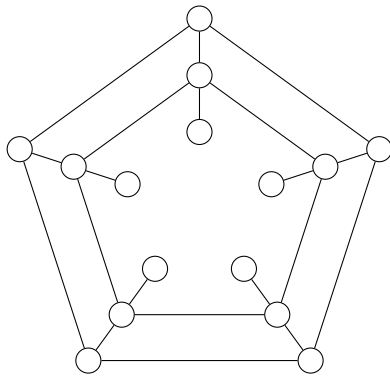
Der erste Ansatz war, ein Programm zu schreiben, das das Problem in polynomieller Laufzeit löst. Währenddessen haben wir herausgefunden, dass bisher nur Algorithmen bekannt sind, die das Problem in nicht-polynomieller Zeit lösen. Deshalb verfolgten wir die Strategie, die eingegebenen Graphen auf bestimmte Eigenschaften zu prüfen und zu vergleichen. Zu diesen Eigenschaften zählen die Anzahl der Knoten und Kanten, der Grad der Knoten, Grade von Nachbarknoten, Kreise und Längen von kürzesten Wegen im Graphen.

Wenn die Graphen isomorph sind, kann man mit Hilfe der zuvor genannten Eigenschaften jedem Knoten des einen Graphen einen des anderen Graphen zuordnen. Diese Zuordnung ist nicht zwingend eindeutig, weshalb es nötig sein kann, mehrere Zuordnungen zu testen, bis eine geeignete gefunden ist. Bei isomorphen Graphen gibt es eine solche Zuordnung, bei anderen Graphen nicht. Wenn die Werte nicht zueinander passen, sind die Graphen nicht isomorph. Da der Algorithmus nicht alle möglichen Zuordnungen testet, können einige Eingabegraphen nicht absolut sicher als isomorph identifiziert werden. Der Algorithmus findet lediglich kein Kriterium, das einen Isomorphismus ausschließt. Ein Beispiel solcher zweier Graphen sind G_1 aus Abb. 1.31 und G_2 aus Abb. ??.

G_1 ist nicht zu G_2 isomorph. Es ist aber möglich, die Knoten des einen Graphen bijektiv denen des anderen Graphen zuzuordnen, sodass die Grade der Knoten, Nachbarknoten, Nachbarnachbarknoten und Nachbar...nachbarknoten für jeden Knoten von G_1 mit dem zugeordneten Knoten von G_2 übereinstimmen.

15.1.3 Fazit

Das Problem hat sich während des Entwicklungsprozesses als schwerer als erwartet herausgestellt, weshalb der Entwicklungsplan weiterentwickelt und abgeändert werden musste. Trotzdem

Abb. 1.32: Graph G_2

ist der finale Algorithmus zufriedenstellend, da die Laufzeit für die meisten Eingabegraphen kurz ist.

16 Skalenfreie Graphen und Feldexperiment

Charlotte Häußler und Philipp Davydov

16.1 Einführung und Idee

Bereits lange vor Beginn der Akademie Grovesmühle wurden die TN über das bevorstehende Feldexperiment des Kurses 3.1 informiert. Es handelt sich um ein akademieweites Projekt zur Untersuchung sozialer Graphen, im Zuge dessen alle Freiwilligen mit Bluetooth-Beacons ausgestattet wurden. Diese Beacons kommunizieren auf einer Distanz von bis zu zwei Metern. Dabei werden jedesmal Zeitpunkt und Dauer der Verbindung zweier Beacons registriert. Die so anonym erhobenen Daten wurden von den KL aufbereitet, erneut anonymisiert und schließlich an die TN des Kurses zur Auswertung übergeben.

16.2 Skalenfreie Graphen

Das Ziel des Feldexperiments war die Untersuchung sozialer Graphen, also Graphen, die soziale Netzwerke darstellen, auf Skalenfreiheit. Da es sich bei dieser Form von Graphen um ein recht neues Gebiet der Mathematik handelt, gibt es noch keine vollständig einheitliche Definition. Zur Betrachtung der Ergebnisse wird die allgemein gängigste Definition verwendet.

Definition 37. Sei $G = (V, E)$ ein zusammenhängender Graph und $P(k)$ der Anteil der Knoten mit Grad k . Dann heißt G skalenfrei, falls die Knotengrade einem Potenzgesetz folgen, das folgendermaßen beschrieben werden kann

$$P(k) \propto k^{-\alpha}$$

mit Skalierungsfaktor α . Dabei gilt normalerweise $1 \leq \alpha \leq 3$.

Anders ausgedrückt verhält sich die Zahl der Knoten mit einem bestimmten Grad in G ähnlich wie die Funktion $f(x) = \frac{1}{x^{|\alpha|}}$.

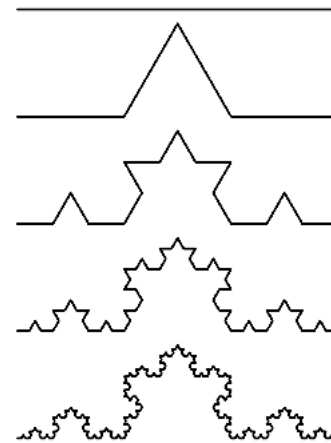


Abb. 1.33: Die selbstähnliche Struktur der Kochkurve (Quelle: https://upload.wikimedia.org/wikipedia/commons/0/0f/Von_Koch-kurvans_utveckling.png)

16.3 Soziale Graphen

Skalenfreie Graphen entstehen durch den Zufallsprozess des *preferential attachment*. Dabei existieren zu Beginn einige initiale Knoten. Danach werden iterativ Knoten ergänzt, die zufällig Kanten bilden. Neu hinzukommende Knoten verknüpfen sich hierbei bevorzugt mit Knoten die bereits einen hohen Grad haben. Je mehr Nachbarn ein Knoten also bereits hat, desto wahrscheinlicher ist das Hinzukommen weiterer Nachbarknoten. Somit können auch soziale Graphen als skalenfrei angesehen werden. Neue Menschen, die einer sozialen Gruppe beitreten, wahrscheinlich zuerst mit denjenigen Kontakte knüpfen werden, die momentan die meisten Kontakte haben bzw. beliebt sind. Die Beliebten werden dann von den Knoten mit hohem Grad repräsentiert.

16.4 Vergleich mit Fraktalen

Fraktale sind selbstähnliche Strukturen, das heißt, dass ihre Struktur in größeren Maßstäben ähnlich der des Anfangszustandes ist. Ein gutes Beispiel ist die Kochkurve in Abb. ??.

Eine Ansicht, die durch die Analogie zwischen der Skalenfreiheit des Graphen und der Maßstabsfreiheit von Fraktalen motiviert ist, ist, dass auch skalenfreie Graphen selbstähnlich sind. Dabei ergibt sich allerdings das Problem, dass aus dem Potenzgesetz nicht unmittelbar die Selbstähnlichkeit folgt.

16.5 Das Feldexperiment

Am Mittwoch den 11.07.2018 wurden die Beacons an die TN des Experiments verteilt und in die Namensschilder integriert. So wurde erreicht, dass die TN ihre Beacons etwa 17 Stunden pro Tag bei sich trugen, wodurch sehr gute Ergebnisse entstanden. Etwa eine Woche später wurde das Experiment beendet, die Beacons eingesammelt und die Rohdaten von den KL in eine leichter zu verarbeitende Form

FEHLËNDES BILD

Abb. 1.34: Auswertung des Experiments, Graph zur Kurszeit

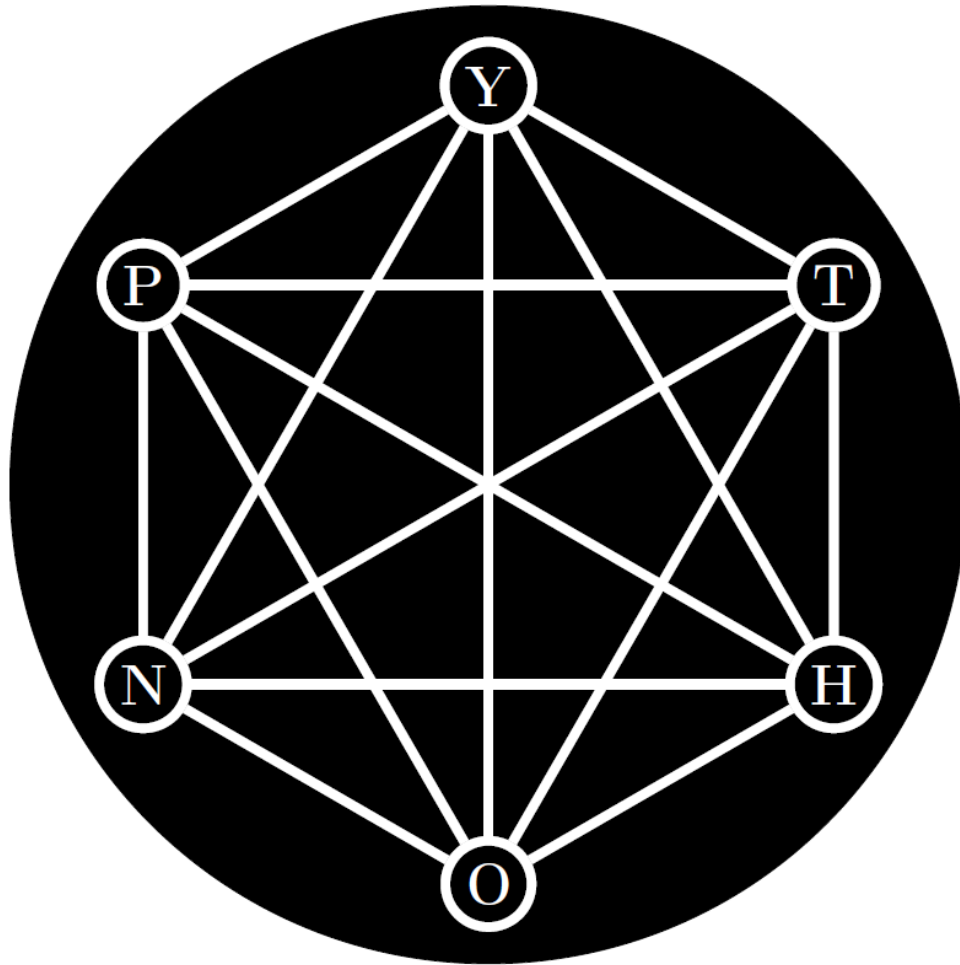
gebracht. Am Ende dieses Prozesses stand eine Liste mit etwa 66000 Zeilen. Diese begannen mit einem Buchstaben zur Kategorisierung in die zwei Bestandteile (sozialer Graph und Seuchensimulation) des Experiments. In den Zeilen für den sozialen Graphen folgten dann die Nummern der beiden kommunizierenden Beacons, der Zeitpunkt der Kontaktaufnahme in Unixzeit (Verstrichene Zeit in Sekunden seit 00:00 Uhr des 1. Januar 1970) und die Dauer der Kopplung in Sekunden. Ein Beispiel wäre die folgende (fiktive) Zeile.

N,9,124,1531654335,1783

Sie beinhaltet die Information, dass während des Experiments die Chips der (anonymisierten) Kennnummern 9 und 124 am 15.07.2018 ab etwa 13:32 Uhr (Unixzeit 1531654335) für eine knappe halbe Stunde (1783 Sekunden) miteinander gekoppelt, also weniger als zwei Meter voneinander entfernt waren.

Zur Auswertung wurde ein Python-Programm entwickelt, das aus den vorhandenen Daten verschiedene Graphen erstellt. Einer von ihnen ist in Abb. ?? dargestellt. Die Beacons werden durch Knoten und die Kopplungen durch Kanten repräsentiert. Es handelt sich um eine Momentaufnahme der bestehenden Verbindungen während der Kurszeit. Verknüpfte Knoten befanden sich demnach zu dieser Zeit in Kopplungsbereichweite, also im selben Raum und somit im selben Kurs. Die Zusammenhangskomponenten stellen also die verschiedenen Kurse dar.

Literaturverzeichnis



- [1] halo i bims 1 quelle
- [2] Davis/Hersh: Erfahrung Mathematik, Birkhäuser 1985, S. 109
- [3] Spektrum der Wissenschaft 1 / 2001, Seite 105; Spektrum der Wissenschaft Verlagsgesellschaft
- [4] Nachweis der Jahreszahl: <https://docs.python.org/3/license.html>, Zugriff: 16.7.18