

# Algorithmen und Datenstrukturen\*

## Inhaltsverzeichnis

<b>Graphentheorie</b>	<b>2</b>
Universelles Phänomen der Netzwerk . . . . .	2
Jenseits von Netzwerken . . . . .	3
Problemstellung: Kartenfärbung . . . . .	3
Problemstellung: Prüfungsplanung . . . . .	4
Definition: Graph und verwandte Begriffe . . . . .	4
Eigenschaft: Summe der Knotengrade . . . . .	5
Isomorphie . . . . .	5
Besondere Graphen . . . . .	6
Problemstellung: Knotenfärbung . . . . .	8
Diskussion: Kombinatorische Explosion . . . . .	9
Eigenschaft: Knotenfärbung und maximaler Knotengrad . . . . .	10
Diskussion: Beweise und Algorithmen . . . . .	10
Einschub: Vollständige Induktion . . . . .	11
Beispiel 1: Summenformel . . . . .	12
Beispiel 2: Färbung und maximaler Knotengrad . . . . .	12
Beispiel 3: Summe der Knotengrade . . . . .	13
Beispiel 4: Nicht-Bipartite Graphen . . . . .	13
Gerichtete Graphen . . . . .	16
Beispiel: Flussüberquerung und Zustandsübergänge . . . . .	17
Definition: Wege und verwandte Begriffe . . . . .	18
Definition: Erreichbarkeit und Distanz . . . . .	19
Eigenschaft: Minimale Wege sind Pfade . . . . .	19
Problemstellung: Topologische Sortierung . . . . .	20
Definition: Zusammenhang . . . . .	21
Eigenschaft: Zusammenhangskomponenten in ungerichteten Graphen . . . . .	21
Definition: Baum . . . . .	22
Eigenschaften: Bäume und Wälder . . . . .	23
Problemstellung: Eulertour . . . . .	24

---

\*Herbstsemester 2018, ETH Zürich

# Graphentheorie

Graphen<sup>1</sup> sind grundlegende mathematische Objekte, mit deren Hilfe viele interessante Zusammenhänge aus unterschiedlichen Anwendungsbereichen elegant beschrieben werden können.

## Universelles Phänomen der Netzwerk

Ein Ziel der Graphentheorie ist Einsichten in das universelle Phänomen von Netzwerken<sup>2</sup> zu liefern.

Netzwerke beschreiben Verknüpfungen, die zwischen verschiedene Objekten bestehen. Sie sind von grosser Bedeutung in zahlreichen Anwendungsbereichen, wie die folgenden konkreten Beispiele aufzeigen.

- **Soziale Netzwerke:** Personen sind durch ihre sozialen Kontakte miteinander verknüpft.
- **Strassennetzwerke:** Orte sind durch Strassenverbindungen miteinander verknüpft.
- **Stromnetzwerke:** Erzeuger und Verbraucher von Elektrizität sind durch elektrische Leitungen miteinander verknüpft.
- **Computernetzwerke** (wie das Internet): Computer sind durch Datenverbindungen miteinander verknüpft.
- **(Künstliche und natürliche) neuronale Netzwerke:** Neuronen sind über Synapsen miteinander verknüpft.

Eine herausragende Eigenschaft viele dieser Netzwerke ist ihre Grösse. Zum Beispiel hat das soziale Netzwerk auf Facebook mehrere Milliarden aktive Nutzern und viele hundert Milliarden Verknüpfungen zwischen Nutzern. Computernetzwerke wie das Internet können diese Grösse sogar um mehrere Ordnungen überschreiten.

Mit Netzwerken dieser Grösse umzugehen ist nur mit Hilfe von Computern möglich. Dabei ist es wichtig, möglichst effiziente Algorithmen zu verwenden, auf die wir in dieser Vorlesung hinarbeiten werden.

Um solche Netzwerke zunächst besser zu verstehen, werden wir vereinfachte mathematische Modelle betrachten, die *Graphen* genannt werden. Dieses Modell erfasst nur welche Objekte eines Netzwerks miteinander verknüpft sind. Wir ignorieren dabei was die Objekte repräsentiere (z.B., Personen, Computer, oder Neuronen) und wie die Verknüpfungen zwischen den Objekten zustande

---

<sup>1</sup>Graphen, wie sie in der Graphentheorie vorkommen, und Funktionsgraphen—wie aus dem Schulunterricht bekannt—haben nichts miteinander zu tun und sollten daher nicht miteinander verwechselt werden.

<sup>2</sup>Hier beziehen wir uns auf den Begriff von Netzwerken, wie er umgangssprachlich verwendet wird. Es gibt auch einen mathematischen Begriff von Netzwerken, der im Zusammenhang mit Netzwerkflüssen vorkommt und auf den wir uns hier nicht beziehen.

kommen. Trotz dieser Vereinfachungen erlauben uns Graphen oft interessante Rückschlüsse, auf die zugrunde liegenden Netzwerke zu treffen.

In einem Graphen, werden die miteinander verknüpften Objekte *Knoten*<sup>3</sup> genannt und die Verknüpfungen zwischen Knoten werden *Kanten*<sup>4</sup> genannt. Knoten  $u$  und  $v$  heisst *adjazent*<sup>5</sup> falls zwischen ihnen eine Kante verläuft. Die Knoten, die eine Kante  $e$  miteinander verknüpft, heissen *Endpunkte*<sup>6</sup> der Kante  $e$ . Ein Knoten  $u$  und eine Kante  $e$  heissen *inzident*,<sup>7</sup> falls  $u$  ein Endknoten der Kante  $e$  ist.

## Jenseits von Netzwerken

Mit Hilfe von Graphen können auch Problemstellungen dargestellt werden, die zunächst nichts mit Netzwerken zu tun haben. Ein interessanter Effekt dabei ist, dass zunächst unterschiedliche Problemstellungen auf dasselbe Graphenproblem zurückgeführt werden können. In diesem Fall kann dann eine Lösungsidee für das zugrundeliegende Graphenproblem auf unterschiedliche Problemstellungen angewendet werden.

Hierzu betrachten wir zwei Beispiele solcher Problemstellungen, nämlich Kartenfärbung und Prüfungsplanung. Beide Problemstellungen lassen sich auf dasselbe Graphenproblem zurückführen, nämlich **Knotenfärbung**.

### Problemstellung: Kartenfärbung

Wir stellen uns eine Landkarte vor, auf der verschiedene Gebiete eingezeichnet sind z.B. Länder oder Kantone. Das Ziel ist es die verschiedenen Gebiete so einzufärben das benachbarte Gebiete unterschiedliche Farben haben. Auf diese Weise ist leichter zu erkennen, welche Teile der Landkarte zu demselben Gebiet gehört. Andererseits möchten wir vermeiden Farben zu verwenden, die nur schwer zu unterscheiden sind. Deshalb ist es auch wichtig, dass wir insgesamt möglichst wenig verschiedene Farben verwenden.

Wir können diese Problemstellung durch Graphen darstellen. Dazu betrachten wir den Graph, dessen Knoten den Gebieten der Landkarte entsprechen. Ferner fügen wir Kanten zwischen allen Paaren von benachbarten Gebieten hinzu. Für diesen Graphen wollen wir die Knoten mit möglichst wenig verschiedenen Farben so einfärben, dass alle Paare von adjazenten Knoten unterschiedliche Farben bekommen.

---

<sup>3</sup>Englisch: vertex

<sup>4</sup>Englisch: edge

<sup>5</sup>Englisch: adjacent

<sup>6</sup>Englisch: end point

<sup>7</sup>Englisch: incident

## Problemstellung: Prüfungsplanung

Wir stellen uns vor, dass an einer Hochschule Termine für eine Zahl von Prüfungen  $P_1, \dots, P_n$  festgelegt werden sollen. Es stehen nur wenige verschiedene Termine  $T_1, \dots, T_k$  zu Verfügung. Falls ein Student sich für mehrere Prüfungen angemeldet hat, müssen diese alle an unterschiedlichen Terminen statt finden.

Wir können diese Problemstellung ebenfalls durch Graphen darstellen. Wir betrachten dazu einen Graphen, dessen Knoten den Prüfungen  $P_1, \dots, P_n$  entsprechen. Für jedes Paar von Prüfungen, für die mindestens ein gemeinsamer Student angemeldet ist, fügen wir eine Kante zwischen den entsprechenden Knoten hinzu. Das Ziel ist die Termine  $T_1, \dots, T_k$  so an die Knoten zuzuweisen, dass alle Paare von adjazenten Knoten unterschiedliche Termine bekommen.

Wir können jetzt sehen das Kartenfärbung und Prüfungsplanung sich auf dasselbe Graphenproblem zurückführen lassen—der Unterschied, dass wir den Knoten in einem Fall Farben und im anderen Fall Termine zuweisen ist unwesentlich.

## Definition: Graph und verwandte Begriffe

Ein *Graph*  $G = (V, E)$  besteht aus einer Menge von *Knoten*  $V$  und einer Menge von *Kanten*  $E$ , wobei jede Kante  $e \in E$  ein ungeordnetes Paar<sup>8</sup>  $e = \{u, v\}$  von zwei Knoten  $u, v \in V$  mit  $u \neq v$  ist.<sup>9</sup> Hier nehmen wir üblicherweise an, dass  $V$  nicht leer ist.

Zwei Knoten  $u, v \in V$  heißen *adjazent* oder *benachbart*<sup>10</sup>, falls  $\{u, v\} \in E$ . Ein Knoten  $u$  und eine Kante  $e$  heißen *inzident*<sup>11</sup>, falls  $u \in e$ . Die *Nachbarschaft*<sup>12</sup>  $N_G(u)$  eines Knoten  $u$  ist die Menge aller Knoten, die adjazent zu  $u$  sind. Der *Grad*<sup>13</sup>  $\deg_G(u) := |N_G(u)|$  eines Knoten ist die Anzahl der benachbarten Knoten von  $u$ . Ein Knoten mit Grad 1 heisst *Blatt*<sup>14</sup>. Ein Knoten mit Grad 0 heisst *isoliert*.

Zum Beispiel im folgenden Graph gilt: "G" ist isoliert, "F" ist ein Blatt, "A" und "E" haben Grad 2, "B" hat Grad 3, "C" und "D" haben Grad 4.

<sup>8</sup>Hier, stellen  $\{u, v\}$  und  $\{v, u\}$  dasselbe ungeordnete Paar da.

<sup>9</sup>Andere Definitionen erlauben manchmal *Schleifen*  $e = \{u\}$  (Englisch: loop) als Kanten, die einen Knoten mit sich selbst verbinden.

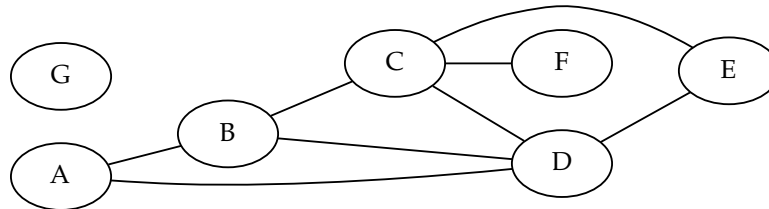
<sup>10</sup>Englisch: adjacent or neighboring

<sup>11</sup>Englisch: incident

<sup>12</sup>Englisch: neighborhood

<sup>13</sup>Englisch: degree

<sup>14</sup>Englisch: leaf



Ein Graph  $G' = (V', E')$  heisst *Teilgraph*<sup>15</sup> von  $G$  falls  $V' \subseteq V$  und  $E' \subseteq E$ . (Da  $G'$  ein Graph ist, gilt hierbei, dass die Kanten in  $E'$  jeweils zwei Knoten in  $V'$  verbinden. Daher bildet nicht jede Knotenmenge  $V' \subseteq V$  und Kantenmenge  $E' \subseteq E$  einen Teilgraph von  $G$ .)

Wir verwenden auch die Notation  $V(G)$  und  $E(G)$  für die Knotenmenge und Kantenmenge eines Graphen  $G$ .

### Eigenschaft: Summe der Knotengrade

Der folgende Satz hält eine allgemeingültige Beziehung zwischen der Summe der Knotengrade und der Anzahl der Kanten fest.

**Satz:** Jeder Graph  $G$  mit  $n$  Knoten  $v_1, \dots, v_n$  und  $m$  Kanten erfüllt

$$\deg_G(v_1) + \dots + \deg_G(v_n) = m.$$

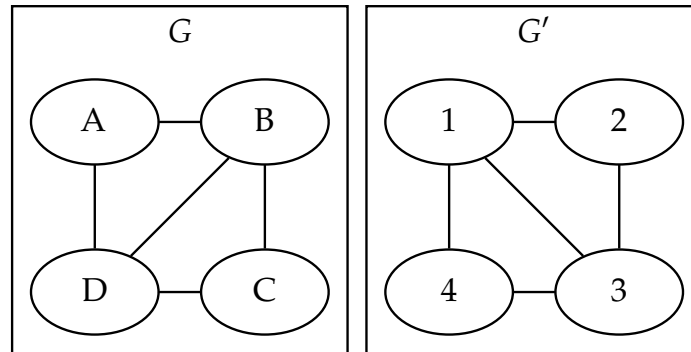
*Beweis:* Wir verteilen  $2m$  Goldtaler gerecht an die Kanten des Graphen, so dass jeder Kante zwei Taler bekommt. Nun verteilt jede Kante ihre Taler an ihre beiden Endpunkte. Auf diese Weise bekommt jeder Knoten von jeder seiner inzidenten Kanten genau einen Taler. Damit erhält jeder Knoten  $v_i$  insgesamt  $\deg_G(v_i)$  Taler. Die Gesamtzahl der Taler können wir nun mit der Summe  $\deg_G(v_1) + \dots + \deg_G(v_n)$  berechnen. Da unsere Umverteilung der Taler die Gesamtzahl nicht verändert hat, gilt  $2m = \deg_G(v_1) + \dots + \deg_G(v_n)$ .  $\square$

### Isomorphie

Falls zwei Graphen  $G$  und  $G'$  sich nur in der Bezeichnung ihrer Knoten unterscheiden (wie in folgendem Beispiel), betrachten wir typischerweise diesen Unterschied als unwesentlich und möchten die Graphen als gleich behandeln.

---

<sup>15</sup>Englisch: subgraph

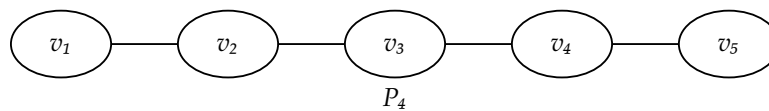


Um diese Idee zu formalisieren, führen wir den Begriff der Isomorphie ein. Zwei Graphen  $G = (V, E)$  und  $G' = (V', E')$  heißen *isomorph*, falls sie sich nur in der Bezeichnung ihrer Knoten unterscheiden, was formell bedeutet, dass eine bijektive Abbildung  $f: V \rightarrow V'$  existiert, so dass

$$\forall u, v \in V. \{u, v\} \in E \Leftrightarrow \{f(u), f(v)\} \in E'.$$

### Besondere Graphen

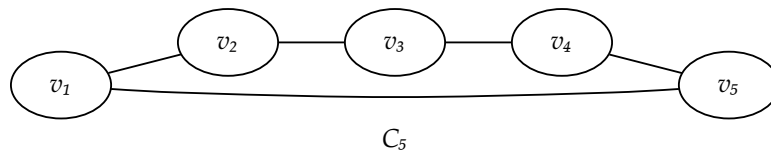
Ein *Pfad*<sup>16</sup>  $P_\ell$  der Länge  $\ell \geq 1$  besteht aus  $\ell + 1$  Knoten  $v_1, \dots, v_{\ell+1}$  und  $\ell$  Kanten  $\{v_1, v_2\}, \{v_2, v_3\} \dots, \{v_\ell, v_{\ell+1}\}$ .



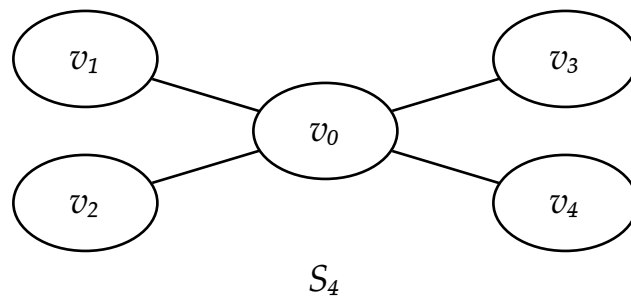
Wir erhalten einen *Kreis*<sup>17</sup>  $C_\ell$  der Länge  $\ell \geq 3$ , indem wir zu einem Pfad  $P_{\ell-1}$  der Länge  $\ell - 1$  eine Kante zwischen den Endpunkte  $v_{\ell-1}, v_\ell$  des Pfads hinzufügen.

<sup>16</sup>Englisch: *path*

<sup>17</sup>Englisch: *cycle*



Ein Stern<sup>18</sup>  $S_\ell$  mit  $\ell \geq 0$  Spitzen besteht aus  $\ell + 1$  Knoten  $v_0, v_1, \dots, v_\ell$  und  $\ell$  Kanten  $\{v_0, v_1\}, \{v_0, v_2\}, \dots, \{v_0, v_\ell\}$ .

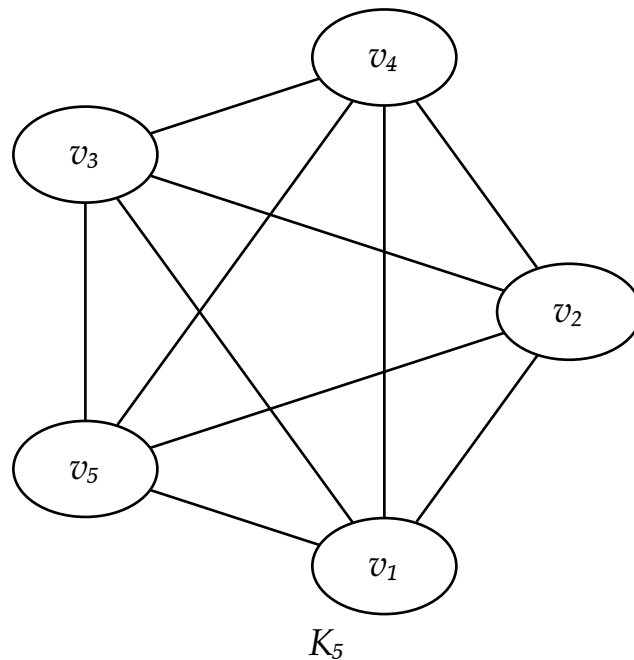


Ein vollständiger Graph<sup>19</sup>  $K_n$  besteht aus  $n \geq 1$  Knoten  $v_1, \dots, v_n$  und allen Kanten  $\{v_i, v_j\}$  mit  $i \neq j$ .

---

<sup>18</sup>Englisch: *star*

<sup>19</sup>Englisch: *complete graph*



Zusammen mit dem Begriff des Teilgraphen lassen sich aus diesen besonderen Graphen interessante Problemstellungen für allgemeine Graphen ableiten. Zum Beispiel können wir fragen, ob ein gegebener Graph einen Kreis als Teilgraph enthält. Hierfür werden wir später effiziente Algorithmen kennenlernen. Wenn wir allerdings fragen, ob ein gegebener Graph mit  $n$  Knoten einen Kreis der Länge  $n$  enthält,<sup>20</sup> dann ist hierfür kein effizienter Algorithmus bekannt und es wird vermutet dass es keinen geben kann.<sup>21</sup>

### Problemstellung: Knotenfärbung

Knotenfärbung ist eine wichtige Problemstellung für Graphen. Sie tritt in vielen Anwendungsbereichen auf. Wir haben zuvor gesehen, dass sich Kartenfärbung und Prüfungsplanung darauf zurückführen lassen.

<sup>20</sup>Ein Kreis, der alle Knoten eines Graphen enthält, wird Hamiltonkreis genannt.

<sup>21</sup>Die berühmte  $P \neq NP$  Vermutung impliziert, dass es keinen effizienten Algorithmus geben kann, um zu entscheiden, ob ein Graph einen Hamiltonkreis enthält.



**Definition:** Eine  $k$ -Färbung eines Graphen  $G = (V, E)$  ist eine Abbildung  $f: V \rightarrow \{1, \dots, k\}$ . Eine Färbung heisst *zulässig*, falls alle Paare von benachbarten Knoten unterschiedliche Farben erhalten, also

$$\forall \{u, v\} \in E. f(u) \neq f(v).$$

Ein Graph  $G$  heisst  $k$ -partit oder  $k$ -färbbar, falls eine zulässige  $k$ -Färbung existiert. (Hierbei werden 2-partite und 3-partite Graphen typischerweise *bipartit* und *tripartit* genannt.) Die *chromatische Zahl*<sup>22</sup>  $\chi(G)$  eines Graphen  $G$  ist die kleinste Zahl  $k$  so dass der Graph  $k$ -partit ist.

Ein vollständiger Graph  $K_n$  hat chromatische Zahl  $n$ . Jeder Pfad  $P_\ell$  mit Länge  $\ell \geq 1$  hat chromatische Zahl 2. Kreise  $C_{2k}$  gerader Länge haben chromatische Zahl 2, wobei Kreise ungerader Länge chromatische Zahl 3 haben.

## Diskussion: Kombinatorische Explosion

Eine wichtige Überlegung zur Knotenfärbung im Zusammenhang mit Algorithmen ist, dass die Zahl aller Färbung exponentiell schnell wächst und es damit für grössere Graphen unmöglich ist, alle Färbungen auszuprobieren, um eine zulässige zu finden. Dieses Phänomen wird oft *kombinatorische Explosion* genannt.

Für einen Graphen mit  $n$  Knoten beträgt die Anzahl der  $k$ -Färbungen  $k^n$  (da es für jeden der  $n$  Knoten  $k$  mögliche Farben gibt).

Um das schnelle Wachstum der Funktion  $k^n$  zu verdeutlichen, betrachten wir folgendes konkretes Beispiel: Für  $k = 2$  und  $n = 50000$  beträgt die Anzahl der  $k$ -Färbungen mehr als  $10^{10000}$  (eine 1 gefolgt von 10000 Nullen). Zum Vergleich schätzt man, dass seit dem Urknall "nur"  $10^{60}$  Planck-Zeiteinheiten<sup>23</sup> vergingen. Also selbst ein unglaublich schneller Computer, der in jeder Planck-Einheit eine 2-Färbung ausprobiert, hätte nur einen verschwindend kleinen Anteil aller 2-Färbungen eines Graphen mit 50000 Knoten durchlaufen. Die Situation würde sich auch nicht merklich verändern, wenn man diese Aufgabe auf viele verschiedene Computer verteilt, da man schätzt, dass es "nur"  $10^{80}$  Atome im Universum gibt, was wohl eine obere Schranke für die mögliche Anzahl an Computern wäre.

Überraschenderweise ist es möglich zulässige 2-Färbungen bipartite Graphen zu finden ohne alle 2-Färbungen durchzuprobieren. Insbesondere werden wir im Laufe dieser Vorlesung Algorithmen kennenlernen, die die kombinatorische Explosion von 2-Färbungen umgehen und sehr schnell zulässige 2-Färbungen grosser bipartiter Graphen finden.

---

<sup>22</sup>Englisch: chromatic number

<sup>23</sup>Planck-Einheit ist die kleinst mögliche Zeiteinheit laut den aktuell bekannten Gesetzen der Physik

Allerdings wird vermutet, dass es nicht immer möglich ist, diese kombinatorische Explosion zu umgehen. Diese Vermutung ist als  $P \neq NP$  Vermutung bekannt und zählt zu den bedeutendsten Vermutungen der Wissenschaften. Eine Konsequenz dieser Vermutung besagt informell, dass es tripartite Graphen gibt, für die es unmöglich ist eine zulässige 3-Färbung zu finden, ohne im wesentlichen alle 3-Färbungen durchzuprobieren.

## Eigenschaft: Knotenfärbung und maximaler Knotengrad

Der folgende Satz stellt eine Beziehung zwischen dem maximalen Knotengrad und der chromatischen Zahl eines Graphen auf.

**Satz:** Jeder Graph mit maximalem Knotengrad  $\Delta$  hat eine zulässige  $(\Delta + 1)$ -Färbung.

Wir werden diesen Satz erst später beweisen. (Als ein Beispiel zur **vollständigen Induktion**.) Später in der Vorlesung werden wir auch einen effizienten Algorithmus kennenlernen, der eine zulässige  $(\Delta + 1)$ -Färbung findet.

Anders formuliert besagt der Satz, dass  $\chi(G) \leq \Delta(G) + 1$  für jeden Graphen gilt, wobei  $\Delta(G)$  der *maximale Knotengrad*<sup>24</sup> des Graphen  $G$  ist.

Diese obere Schranke für die chromatische Zahl ist *scharf* in dem Sinn dass es Graphen  $G$  gibt, so dass Gleichheit  $\chi(G) = \Delta(G) + 1$  gilt. Beispiele solcher Graphen sind vollständige Graphen  $K_n$ , für die  $\chi(K_n) = n$  und  $\Delta(G) = n - 1$  gilt sowie ungerade Kreise  $C_{2k+1}$ , die chromatische Zahl 3 und maximalen Knotengrad 2 haben. Allerdings gilt Gleichheit nicht für jeden Graphen.<sup>25</sup> Zum Beispiel haben Pfade  $P_\ell$  und gerade Kreise  $C_{2k}$  chromatische Zahl 2 und maximalen Knotengrad 2. Ein extremes Beispiel sind Sterne  $S_\ell$ , die chromatische Zahl 2 und maximalen Knotengrad  $\ell$  haben.

## Diskussion: Beweise und Algorithmen

Im folgenden Abschnitt, beschäftigen wir uns ausgiebig mit mathematischen Beweisen—zunächst hauptsächlich um allgemeine Eigenschaften von Graphen festzustellen. Allerdings spielen Beweise auch eine grosse Rolle im Bereich der Algorithmen, was wir an dieser Stelle kurz erläutern.

Wenn wir Algorithmen untersuchen, werden wir uns zum Einen dafür interessieren, was für Eigenschaften ihre Ausgaben erfüllen, z.B. ob die Ausgaben korrekt sind oder anderen Qualitätsansprüchen gerecht werden. Zum Anderen,

---

<sup>24</sup>Englisch: *maximum vertex degree*

<sup>25</sup>Tatsächlich gilt die Gleichheit im wesentlichen nur für vollständige Graphen und ungerade Kreise.

werden wir uns dafür interessieren, ob der Algorithmus effizient ist, also selbst für grössere Eingaben mit einem akzeptablen Zeitaufwand auskommt.<sup>26</sup>

Man könnte beide Aspekte von Algorithmen auf pragmatische Weise betrachten, indem man z.B. den Algorithmus als Computerprogramm implementiert und auf möglichst vielen verschiedenen Eingaben ausführt. Auf diesen Eingaben könnten dann die gewünschten Eigenschaften wie Korrektheit und Effizienz überprüft werden.

Ein erhebliches Problem dieses pragmatischen Ansatz besteht darin, dass er im Allgemeinen keine Rückschlüsse darauf erlaubt, wie sich der Algorithmus auf Eingaben verhält, die nicht explizit getestet wurden. Konkret ist die Sorge, dass der Algorithmus zwar auf den getesteten Eingaben wie gewünscht funktioniert, es aber andere Eingaben gibt, auf denen der Algorithmus entweder falsche Ergebnisse liefert oder einen unverhältnismässig grossen Zeitaufwand hat.

In dieser Vorlesung (wie in der theoretische Informatik allgemein), haben wir den Anspruch Algorithmen zu entwerfen, die auf *allen* Eingaben wie gewünscht funktionieren—also korrekt und effizient sind. Da es unendlich viele solcher Eingaben gibt, ist es nicht möglich alle zu testen. Die einzig uns bekannte Möglichkeit alle Eingaben abzudecken, ist einen mathematischen Beweis dafür zu liefern, dass sich der Algorithmus auf allen Eingaben wie gewünscht verhält.

## Einschub: Vollständige Induktion

Eine wichtige Strategie um Eigenschaften von Graphen und anderen diskreten Strukturen ist die *vollständige Induktion*<sup>27</sup>.

Wir betrachten hierbei eine mathematische Aussage<sup>28</sup>  $A(n)$ , die von einer natürlichen Zahl  $n$  abhängt. Ziel ist es zu beweisen, dass die Aussage  $A(n)$  für alle natürliche Zahlen  $n \in \mathbb{N}$  gilt.

Zum Beispiel könnte  $A(n)$  die Aussage sein, dass die Ungleichung  $\chi(G) \leq \Delta(G) + 1$  für alle Graphen mit  $n$  Knoten gilt. Andererseits könnte  $A(n)$  auch die Aussage sein, dass eine bestimmte arithmetische Identität gilt, wie zum Beispiel  $1 + 2 + \dots + n = n \cdot (n + 1)/2$ .

**Theorem (Vollständige Induktion):** Sei  $A(n)$  eine mathematische Aussage, die von einer natürlichen Zahl  $n$  abhängt. Ein Beweis, dass die Aussage  $A(n)$  für alle  $n \geq 1$  gilt, kann in zwei Teilen erfolgen:

- *Induktionsanfang:* Hier wird die Aussage  $A(n)$  lediglich für  $n = 1$  bewiesen.
- *Induktionsschritt:* Hier wird für alle  $n \geq 2$  die Aussage  $A(n)$  bewiesen unter der Annahme, dass die Aussage  $A(n - 1)$  gilt.

<sup>26</sup>Wir werden erst später formalisieren, was es für einen Algorithmus mit akzeptablen Zeitaufwand auszukommen. An dieser Stelle ist die genaue Bedeutung davon noch unwesentlich.

<sup>27</sup>Englisch: *mathematical induction*

<sup>28</sup>In der Mathematik ist eine Aussage alles was in sinnvoller Weise ein Wahrheitswert (also "wahr" oder "falsch") zugeordnet werden kann.

Diese Strategie kann Beweise stark vereinfachen, da wir für den Beweis der Aussage  $A(n)$  verwenden können, dass die Aussage  $A(n - 1)$  gilt.

Während des Beweises der Aussage  $A(n)$  für  $n \geq 2$  wird die Annahme, dass  $A(n - 1)$  gilt, die *Induktionshypothese*<sup>29</sup> oder *Induktionsannahme* genannt.

### Beispiel 1: Summenformel

Sei  $A(n)$  die Aussage, dass die Summe  $1 + \dots + n$  der ersten  $n$  natürlichen Zahlen  $n \cdot (n + 1)/2$  beträgt.

Für den Induktionsanfang zeigen wir die Aussage  $A(1)$ . Die Summe der ersten  $n = 1$  Zahlen ist 1, was wir auch erhalten, wenn wir  $n \cdot (n + 1)/2 = 1 \cdot (1 + 1)/2 = 1$  für  $n = 1$  auswerten.

Für den Induktionsschritt zeigen wir die Aussage  $A(n)$  für ein beliebiges  $n \geq 2$ . Die Induktionshypothese ist, dass die Summe der ersten  $n - 1$  Zahlen  $(n - 1) \cdot n/2$  beträgt. Die entscheidende Einsicht ist nun, dass die Summe der ersten  $n$  Zahlen berechnet werden kann, indem wir  $n$  zu der Summe der ersten  $n - 1$  Zahlen addieren. Daher können wir die Induktionshypothese folgendermassen verwenden,

$$\begin{aligned} 1 + \dots + n &= (1 + \dots + n - 1) + n \\ &= (n - 1) \cdot n/2 + n \\ &= (n + 1) \cdot n/2. \end{aligned}$$

Der letzte Schritt folgt durch Anwendung üblicher Rechenregeln.<sup>30</sup>  $\square$

### Beispiel 2: Färbung und maximaler Knotengrad

Sei  $A(n)$  die Aussage, dass für alle Graphen  $G$  mit  $n$  Knoten die Ungleichung  $\chi(G) \leq \Delta(G) + 1$  gilt.

Für den Induktionsanfang müssen wir alle Graphen mit  $n = 1$  Knoten betrachten. In dem Fall hat der Graph keine Kanten. Damit gilt  $\chi(G) = 1$  und  $\Delta(G) = 0$ , was die gewünschte Ungleichung erfüllt.

Für den Induktionsschritt betrachten wir einen beliebigen Graphen  $G = (V, E)$  mit  $n \geq 2$  Knoten. Sei  $v \in V$  ein beliebiger Knoten. Indem wir  $v$  und alle inzidenten Kanten entfernen, erhalten wir einen Teilgraphen  $G' = (V', E')$  von  $G$  mit  $n - 1$  Knoten. Per Konstruktion gilt  $\Delta(G') \leq \Delta(G)$ . Laut Induktionshypothese gilt  $\chi(G') \leq \Delta(G') + 1 \leq \Delta(G) + 1$ . Sei  $f': V' \rightarrow \{1, \dots, k\}$  nun eine zulässige Färbung von  $G'$  mit  $k = \Delta(G) + 1$  Farben. Da  $k$  grösser als der Grad von  $v$  in  $G$  ist, gibt es eine Farbe  $c \in \{1, \dots, k\}$  die nicht in der Nachbarschaft  $N_G(v)$

<sup>29</sup>Englisch: *induction hypothesis*

<sup>30</sup>Konkret verwenden wir das Distributivgesetz um  $(n - 1) \cdot n/2 + n = (n - 1 + 2) \cdot n/2 = (n + 1) \cdot n/2$  zu erhalten.

vorkommt. Daher können wir  $f'$  zu einer zulässigen Färbung von  $G$  erweitern, indem wir  $v$  die Farbe  $c$  zuweisen.  $\square$

### Beispiel 3: Summe der Knotengrade

Sei  $G$  ein Graph mit  $n$  Knoten  $v_1, \dots, v_n$  und  $m$  Kanten. Dann gilt die Identität

$$\deg_G(v_1) + \dots + \deg_G(v_n) = 2m.$$

In diesem Abschnitt beweisen wir diese Identität per Induktion über die Anzahl der Kanten  $m$ .

Sei  $A(m)$  die Aussage, dass die obige Identität für alle Graphen mit  $m$  Kanten gilt.

Für den Induktionsanfang müssen wir alle Graphen mit  $m = 1$  Kanten betrachten. In dem Fall hat der Graph genau zwei Knoten, jeweils mit Grad 1. Damit ist die Summe der Grade  $2 = 2m$ , wie gewünscht.

Für den Induktionsschritt betrachten wir einen beliebigen Graphen  $G = (V, E)$  mit  $m \geq 2$  Kanten. Sei  $G' = (V, E')$  der Teilgraph, den wir erhalten indem wir eine beliebige Kante  $e = \{v_i, v_j\} \in E$  entfernen. Also gilt  $|E'| = m - 1$  und per Induktionshypothese  $\deg_{G'}(v_1) + \dots + \deg_{G'}(v_n) = 2(m - 1)$ . Ferner haben alle Knoten in  $G'$  denselben Grad wie in  $G$  bis auf  $v_i$  und  $v_j$ , deren Grad in  $G'$  um eins kleiner als in  $G$  ist. Damit gilt

$$\begin{aligned} \deg_G(v_1) + \dots + \deg_G(v_n) &= 2 + \deg_{G'}(v_1) + \dots + \deg_{G'}(v_n) \\ &= 2 + 2(m - 1) = 2m. \quad \square \end{aligned}$$

### Beispiel 4: Nicht-Bipartite Graphen

Bei bipartiten Graphen kann man leicht kenntlich machen, dass eine zulässige 2-Färbung existiert, indem man alle Knoten der einen Farbe auf der linken Seite und alle Knoten der anderen Farbe auf der rechten Seite aufzeichnet. Wird der Graph so aufgezeichnet, ist ersichtlich, dass der Graph bipartit ist, da alle Kanten zwischen der linken und der rechten Seite verlaufen (und es keine Kanten gibt die innerhalb der linken Seite oder innerhalb der rechten Seite verlaufen).

Kann man bei einem nicht-bipartiten Graph auch leicht kenntlich machen, dass keine zulässige 2-Färbung existiert? Die Herausforderung hierbei scheint zu sein, dass wir von exponentiell vielen 2-Färbung ausschliessen müssen, dass sie zulässig sind. Allgemein scheint es schwieriger die Nichtexistenz eines Objektes (hier: die Nichtexistenz einer zulässigen 2-Färbung) kenntlich zu machen als die Existenz eines Objektes.<sup>31</sup>

<sup>31</sup>Diese Idee wird durch die  $NP \neq coNP$ -Vermutung in der theoretischen Informatik formalisiert, welche mit der  $P \neq NP$ -Vermutung zusammenhängt.

Der folgende Satz zeigt, dass es überraschenderweise für nicht-bipartite Graphen immer möglich ist, die Nichtexistenz einer 2-Färbung kenntlich zu machen, indem man einen Kreis ungerader Länge als Teilgraph hervorhebt.

**Satz:** Ein Graph ist nicht bipartit genau dann wenn er einen Kreis ungerader Länge als Teilgraph enthält.

Anders formuliert geht die Nichtexistenz einer 2-Färbung immer mit der Existenz eines ungeraden Kreises einher.

Dieser Satz vereint zwei Aussagen, die wir getrennt voneinander beweisen werden.

1. Ein Graph ist nicht bipartit, wenn er einen Kreis ungerader Länge enthält.
2. Wenn ein Graph nicht bipartit ist, dann enthält er einen Kreis ungerader Länge.

Um die erste Aussage zu beweisen, reicht aus zu verwenden, dass Kreise ungerader Länge nicht bipartit sind. Jede 2-Färbung des Graphen ist auch eine 2-Färbung des enthaltenen ungeraden Kreis und kann daher nicht zulässig sein.

Um die zweite Aussage zu beweisen, müssen wir zeigen dass die Nichtexistenz einer zulässigen 2-Färbung bedeutet dass ein ungerader Kreis im Graphen existiert. Äquivalent dazu, müssen wir zeigen dass die Nichtexistenz eines ungeraden Kreises im Graphen bedeutet dass eine zulässige 2-Färbung existiert.

Wir werden diesen Satz mittels vollständiger Induktion beweisen.

Sei  $A(n)$  die Aussage, dass alle Graphen mit höchstens  $n$  Knoten, die keinen Kreis ungerader Länge enthalten, bipartit sind.

Für den Induktionsanfang müssen wir alle Graphen mit  $n = 1$  Knoten betrachten. Solche Graphen können keine Kante enthalten und sind damit trivialerweise bipartit.

Für den Induktionsschritt müssen wir  $A(n)$  für ein allgemeines  $n \geq 2$  zeigen unter der Annahme, dass  $A(n - 1)$  gilt. Sei  $G = (V, E)$  ein Graph mit  $n$  Knoten, der keinen ungeraden Kreis enthält. Wir müssen zeigen, dass  $G$  bipartit ist. Wir können dabei annehmen, dass  $G$  mindestens eine Kante enthält (da  $G$  ansonsten trivialerweise bipartit wäre). Sei  $v \in V$  ein beliebiger Knoten in  $G$  mit Grad  $\deg(v) \geq 1$ . Indem wir  $v$  mit seiner Nachbarschaft  $N(v)$  zu einem neuen Knoten  $v'$  verschmelzen, erhalten wir einen neuen Graphen  $G' = (V', E')$  mit  $n - \deg(v) \leq n - 1$  Knoten.<sup>32</sup> Da  $G$  keinen ungeraden Kreis hat, stellt sich heraus, dass  $G'$  auch keinen ungeraden Kreis enthält. (Diesen wichtigen Schritt werden wir später sorgfältiger rechtfertigen.) Laut Induktionshypothese  $A(n - 1)$  ist demnach  $G'$  bipartit. Sei  $f': V' \rightarrow \{1, 2\}$  eine zulässige 2-Färbung von  $G'$ . Wir

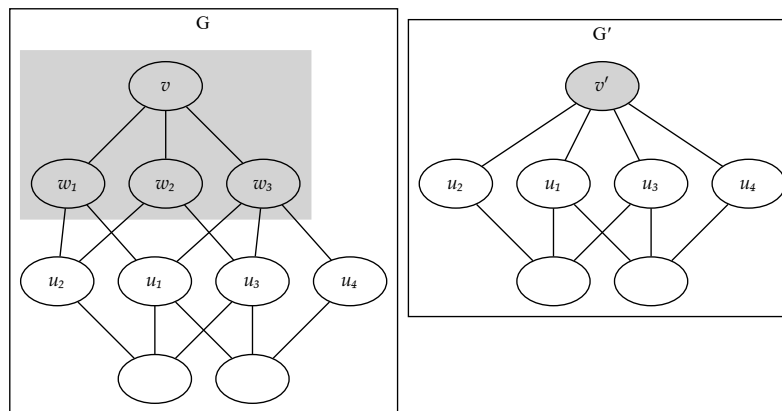
---

<sup>32</sup>Konkret erhalten wir  $G'$  indem wir  $v$  und die Nachbarn von  $v$ , sowie all inzidenten Kanten entfernen und einen neuen Knoten  $v'$  hinzufügen, der mit allen Knoten verbunden ist, die einen Nachbarn von  $v$  als Nachbarn haben.

erhalten die folgende zulässige 2-Färbung  $f$  von  $G$ ,

$$f(u) = \begin{cases} f'(v') & \text{für } u \in N_G(v), \\ 3 - f'(v') & \text{für } u = v, \\ f'(u) & \text{sonst.} \end{cases}$$

Also erhalten die Knoten in der Nachbarschaft von  $v$  dieselbe Farbe wie  $v'$  in der Färbung  $f'$ , der Knoten  $v$  erhält die verbleibende Farbe, und alle anderen Knoten erhalten dieselbe Farbe wie in  $f'$ .



Um zu prüfen, dass  $f$  eine zulässige 2-Färbung ist, reicht es aus alle Kanten  $e \in E$  zu betrachten, die zu einem Nachbarn von  $v$  inzident sind. (Alle anderen Kanten existieren unverändert in  $G'$ .) Es gibt keine Kanten zwischen zwei Nachbarn von  $v$ , da  $G$  keinen ungeraden Kreis (der Länge 3) enthält. Also verläuft  $e$  entweder zwischen  $v$  und einem Nachbarn von  $v$  oder zwischen einem Nachbarn von  $v$  und einem Knoten in  $V'$ . Im ersten Fall haben die Endpunkte von  $e$  Farben  $f'(v')$  und  $3 - f'(v')$  (laut Konstruktion von  $f$ ) und sind damit zulässig gefärbt. Im zweiten Fall entspricht  $e$  einer Kante  $e'$  in  $G'$ , die inzident zu  $v'$  ist, und die Endpunkte von  $e$  haben in  $f$  dieselbe Farben wie die Endpunkte von  $e'$  in  $f'$ . Da  $f'$  ein zulässige Färbung ist, sind damit auch die Endpunkte von  $e$  auch unterschiedlich gefärbt in  $f$ .

Wir müssen noch begründen warum  $G'$  keinen ungeraden Kreis enthält (was uns erlaubt die Induktionshypothese auf  $G'$  anzuwenden). Dazu konstruieren wie zu jedem Kreis  $C'$  in  $G'$  einen Kreis  $C$  in  $G$ , so dass die Längen von  $C$  und  $C'$  gleich sind oder sich genau um 2 unterscheiden. Da  $G$  keinen ungeraden Kreis hat, folgt daraus, dass auch  $G'$  keinen ungeraden Kreis enthält. Sei  $C'$  ein beliebiger Kreis in  $G'$ . Für den Fall, dass  $C'$  nicht den Knoten  $v'$  enthält, dann ist  $C'$  auch ein Kreis in  $G$  und wir können  $C = C'$  wählen. Also müssen wir

nur noch den Fall betrachten, dass  $v'$  in  $C'$  enthalten ist. Seien  $w_1$  und  $w_2$  die Knoten die unmittelbar vor und nach  $v'$  in  $C'$  vorkommen. Laut Konstruktion von  $G'$  existieren Knoten  $u_1 \in N_G(w_1) \cap N_G(v)$  und  $u_2 \in N_G(w_2) \cap N_G(v)$ . Falls  $u_1 = u_2$ , dann erhalten wir aus  $C'$  einen Kreis  $C$  in  $G$  derselben Länge, indem wir  $v'$  durch den Knoten  $u_1 = u_2$  ersetzen. Falls  $u_1 \neq u_2$ , dann erhalten wir aus  $C'$  einen Kreis  $C$  in  $G$ , indem wir  $v'$  durch die Sequenz  $w_1, v, w_2$  ersetzen. In diesem Fall ist die Länge von  $C$  um genau 2 grösser als die Länge von  $C'$ .  $\square$

**Bemerkung:** Es ist nützlich zu fragen, ob wir den Graphen  $G'$  im Induktionsschritt auch einfacher konstruieren könnten. Zum Beispiel hätten wir wie im **Beweis** des Satz über Färbungen und maximalen Knotengrad einfach einen beliebigen Knoten entfernen können. Tatsächlich kann auf einen solchen Teilgraphen die Induktionshypothese leicht angewendet werden. Allerdings ist es im allgemeinen nicht möglich die Färbung eines solchen Teilgraphen zu einer Färbung des ganzen Graphen zu erweitern. In unserem Beweis wurde  $G'$  so konstruiert, dass zum einen die Induktionshypothese anwendbar ist und zum anderen jede Färbung von  $G'$  auf  $G$  erweitert werden kann.

## Gerichtete Graphen

In manchen Netzwerken, haben Verknüpfungen eine Richtung. Zum Beispiel kann in einem Strassennetzwerk eine Einbahnstrasse nur in eine Richtung durchlaufen werden; auch in sozialen Netzwerken, wie dem "follower" Netzwerk von Twitter, bestehen viele Verbindungen nur in eine Richtung. Ein weiteres wichtiges Beispiel ist das Netzwerk der "hyperlinks" im "World Wide Web" (WWW), das eine grosse Bedeutung für Suchmaschinen wie Google hat, da es erlaubt indirekt Rückschlüsse auf die Relevanz und Qualität von Dokumenten zu ziehen.

Wir modellieren Netzwerke in denen Verknüpfungen möglicherweise nur in eine Richtung verlaufen durch *gerichtete Graphen*.

**Definition:** Ein gerichteter Graph  $G = (V, E)$  besteht (wie ein ungerichteter Graph) aus einer Knotenmenge  $V$  und einer Kantenmenge  $E$ , wobei jede Kante  $e \in E$  jedoch ein *geordnetes Paar*  $e = (u, v)$  zweier Knoten  $u, v \in V$  mit  $u \neq v$  ist.

Wir nennen  $u$  den *Fuss*<sup>33</sup> und  $v$  den *Kopf*<sup>34</sup> der gerichteten Kante  $e = (u, v)$ . Falls der Graph  $G$  eine gerichtete Kante  $(u, v) \in E$  enthält, sagen wir  $u$  ist ein (*direkter*) *Vorgänger*<sup>35</sup> von  $v$  in  $G$  und  $v$  ist ein (*direkter*) *Nachfolger*<sup>36</sup> von  $u$  in  $G$ . Wir bezeichnen mit  $N_G^-(v)$  die Menge der Vorgänger von  $v$  und mit  $N_G^+(u)$  die Menge der Nachfolger von  $u$ . Die Anzahl der Vorgänger nennen wir *Eingangsgrad*<sup>37</sup>

<sup>33</sup>Englisch: *tail*

<sup>34</sup>Englisch: *head*

<sup>35</sup>Englisch: *predecessor*

<sup>36</sup>Englisch: *successor*

<sup>37</sup>Englisch: *indegree*



$\text{deg}_G^-(v) := |N_G^-(v)|$  und die Anzahl der Nachfolger nennen wir *Ausgangsgrad*<sup>38</sup>  
 $\text{deg}_G^+(u) := |N_G^+(u)|$ .

Ein Knoten mit Eingangsgrad 0 nennen wir *Quelle*<sup>39</sup> und ein Knoten mit Ausgangsgrad 0 nennen wir *Senke*<sup>40</sup>.

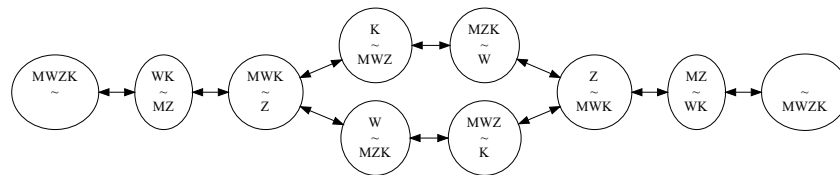
Jeder ungerichteten Graph  $G = (V, E)$  kann auch als ein gerichteter Graph betrachtet werden, indem wir jede ungerichtete Kante  $\{u, v\}$  durch die zwei gerichtete Kanten  $(u, v)$  und  $(v, u)$  ersetzen.

## Beispiel: Flussüberquerung und Zustandsübergänge

In den nächsten Abschnitten werden wir einige neue Begriffe über gerichtete und ungerichtete Graphen einführen, die wir an dieser Stelle durch folgendes Rätsel motivieren. (Tatsächlich ist die graphentheoretische Lösung dieses Rätsels ein Spezialfall von Model Checking, für dessen Entwicklung Clarke, Emerson und Sifakis den Turing Award 2007 erhielten.)

Ein Mann möchte zusammen mit einem Wolf, einer Ziege und einem Kohlkopf einen Fluss überqueren. Sein Boot fasst ausser ihm nur einen weiteren Passagier. Er kann weder den Wolf mit der Ziege noch die Ziege mit dem Kohl unbeaufsichtigt an einem Ufer lassen.

Wir modellieren dieses Rätsel mit Hilfe eines Graphen. Jeder Knoten beschreibt einen möglichen Zustand des "Systems", also auf welcher Seite des Flusses (~) sich der Mann (M), der Wolf (W), die Ziege (Z), und der Kohl (K) befinden. Die Kanten des Graphen geben an welche Zustandsübergänge möglich sind durch eine Bootsfahrt über den Fluss.



Ziel ist es vom Anfangszustand "MWZK~" (alle Beteiligten sind auf der einen Seite des Flusses) zum Zielzustand "~MWZK" (alle Beteiligten sind auf der anderen Seite des Flusses) zu gelangen. Eine Lösung entspricht einer Sequenz von Zuständen beginnend mit dem Anfangszustand und endend mit dem Zielzustand, so dass aufeinanderfolgende Zustände einem möglichen Zustandsübergang entsprechen.

<sup>38</sup>Englisch: *outdegree*

<sup>39</sup>Englisch: *source*

<sup>40</sup>Englisch: *sink*

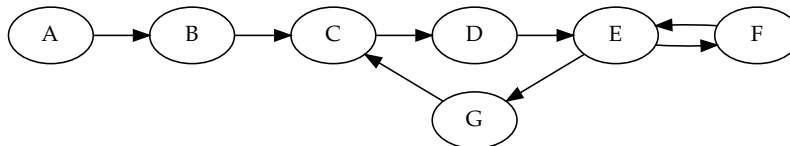
In dem obigen Graphen, entspricht eine Lösung also einer Sequenz von Knoten, so dass zwischen aufeinanderfolgenden Knoten eine Kante verläuft. Anhand des Graphen ist auch ersichtlich, dass eine Lösung existiert und dass die kürzeste Lösung mit 7 Überfahrten auskommt.

Andere Rätsel dieser Art können auch dadurch gelöst werden, indem wir für alle möglichen Zustände Knoten einführen und für alle möglichen Zustandsübergänge Kanten hinzufügen.

### Definition: Wege und verwandte Begriffe

In einem Graphen  $G = (V, E)$ , besteht ein *Weg*<sup>41</sup> mit Startknoten  $v_0$  und Endknoten  $v_\ell$  der Länge  $\ell \geq 0$  aus einer Sequenz  $v_0, v_1, \dots, v_\ell$  von Knoten in  $V$ , so dass alle aufeinander folgenden Knoten  $v_i, v_{i+1}$  durch eine gerichtete Kante  $(v_i, v_{i+1}) \in E$  verbunden sind.

Ein *Pfad*<sup>42</sup> ist ein Weg, in dem jeder Knoten höchstens einmal vorkommt. Ein *geschlossener Weg*<sup>43</sup> ist ein Weg, so dass  $v_0 = v_\ell$  gilt. Ein *Zyklus*<sup>44</sup> ist ein geschlossener Weg der Länge  $\ell \geq 2$ . Ein (*einfacher*) *Kreis*<sup>45</sup> ist ein geschlossener Weg der Länge  $\ell \geq 3$ , so dass in der Sequenz  $v_1, \dots, v_\ell$  jeder Knoten höchstens einmal vorkommt.



In diesem Graph sind A, B, C, D, E, F und A, B, C, D, E, G, C, D, E, F zwei Wege von A nach F. Aber nur der erste Weg ist auch ein Pfad.

Die Sequenzen C, D, E, F, E, G, C und E, F, E sind Zyklen, aber keine Kreise. Die Sequenz C, D, E, G, C ist ein Kreis.

**Bemerkung:** Wir verwenden die Begriffe Weg, Pfad, Zyklus, und Kreis auch für ungerichtete Graphen, indem wir die Begriffe auf die entsprechenden gerichteten Graphen anwenden (wie im Abschnitt **Gerichtete Graphen** beschrieben).

<sup>41</sup>Englisch: *walk*

<sup>42</sup>Englisch: *path*

<sup>43</sup>Englisch: *closed walk*

<sup>44</sup>Englisch: *circuit*

<sup>45</sup>Englisch: (*simple*) *cycle*

## Definition: Erreichbarkeit und Distanz

Ein Knoten  $v$  heisst *erreichbar* in einem Graphen  $G$  von einem Knoten  $u$ , falls  $G$  einen Weg von  $u$  nach  $v$  enthält.

Die *Distanz* von  $u$  nach  $v$  in  $G$  ist die Länge des kürzesten Wegs von  $u$  nach  $v$  in  $G$ .

## Eigenschaft: Minimale Wege sind Pfade

Falls ein Weg mit Startknoten  $v_0$  und Endknoten  $v_\ell$  der Länge  $\ell$  kein Pfad ist, kann der Weg abgekürzt werden ohne den Startknoten oder Endknoten zu verändern. Auf diese Weise können wir Wege solange abkürzen bis wir einen Pfad erhalten.

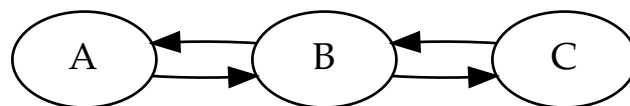
**Satz:** Falls ein Graph einen Weg der Länge höchstens  $\ell$  von  $v_0$  nach  $v_\ell$  enthält für  $v_0 \neq v_\ell$ , so enthält er auch einen Pfad der Länge höchstens  $\ell$  von  $v_0$  nach  $v_\ell$ .

Wir beweisen den Satz per Induktion. Sei  $A(\ell)$  die Aussage, dass der Satz für eine bestimmte Länge  $\ell \geq 1$  gilt.

Die Aussage  $A(1)$  gilt, da jeder Weg der Länge 1 mit unterschiedlichem Start- und Endknoten ein Pfad sein muss.

Sei  $\ell \geq 2$ . Für den Induktionsschritt zeigen wir, dass  $A(\ell)$  gilt, wenn  $A(\ell - 1)$  gilt. Sei  $v_0, \dots, v_\ell$  ein Weg der Länge  $\ell$  von  $v_0$  nach  $v_\ell$ . Falls dieser Weg kein Pfad ist, gibt es einen Knoten  $v_i = v_j$  mit  $i < j$ , der mehr als einmal vorkommt. Dann ist  $v_0, \dots, v_i, v_{j+1}, \dots, v_\ell$  ein Weg von  $v_0$  nach  $v_\ell$  der Länge  $\ell - (j - i) \leq \ell - 1$ . Laut Induktionshypothese gibt es daher einen Pfad von  $v_0$  nach  $v_\ell$  der Länge höchstens  $\ell$ .

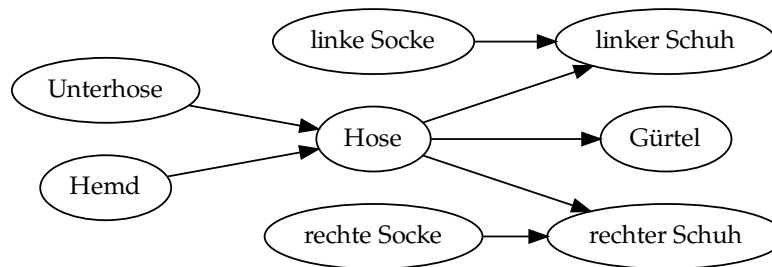
**Bemerkung:** An dieser Stelle könnten wir fragen, ob minimale Zyklen Kreise sind. Leider ist hier nicht leicht eine Verbindung zu formulieren, wie das folgende Beispiel zeigt.



Dieser Graph enthält zwar einen Zyklus A, B, C, B, A mit drei verschiedenen Knoten, aber keinen Kreis.

## Problemstellung: Topologische Sortierung

Graphen können Abhängigkeiten zwischen verschiedenen Vorgängen ausdrücken. Zum Beispiel stellt der folgende Graph Abhängigkeiten beim Anziehen verschiedener Kleidungsstücke dar.



Eine wichtige Problemstellung für solche Graphen ist eine Reihenfolge der Knoten zu finden, so dass alle Abhängigkeiten erfüllt werden. Zum Beispiel erfüllt die Reihenfolge "Unterhose", "Hemd", "Hose", "Gürtel", "linke Socke", "linker Schuh", "rechte Socke", "rechter Schuh" alle oben dargestellten Abhängigkeiten beim Anziehen von Kleidungsstücken. Diese Reihenfolgen werden "topologische Sortierungen" genannt.

**Definition:** Eine *topologische Sortierung* eines gerichteten Graphen  $G = (V, E)$  ist eine Aufzählung  $v_1, \dots, v_n$  aller Knoten in  $V$ , so dass für jede Kante  $(v_i, v_j) \in E$  gilt, dass  $i < j$ .

Wenn wir die Knoten entsprechend einer topologischen Sortierung  $v_1, \dots, v_n$  von links nach rechts in einer Reihe anordnen, dann sind alle Kanten des Graphen von links nach rechts gerichtet.

An dieser Stelle untersuchen wir die Frage, welche gerichtete Graphen eine topologische Sortierung haben. Später in der Vorlesung werden wir effiziente Algorithmen kennenlernen, die topologische Sortierungen berechnen können. Tatsächlich enthält der Beweis in diesem Abschnitt die Kernidee, die den effizienten Algorithmen zugrunde liegt.

**Satz:** Ein gerichteter Graph hat eine topologische Sortierung genau dann wenn er keinen Zyklus enthält.

Ein gerichteter Graph ohne Zyklus wird *azyklisch* genannt. Um den Satz zu beweisen, zerlegen wir ihn in zwei Teile:

1. Jeder Graph, der eine topologische Sortierung hat, ist azyklisch.
2. Jeder azyklische Graph hat eine topologische Sortierung.

Der erste Teil ist leicht einzusehen, da in einer topologischen Sortierung jeder Weg von links nach rechts verläuft und damit nicht denselben Start- und Endknoten haben kann.

Um den zweiten Teil zu zeigen, beweisen wir zunächst den folgenden Satz.

**Satz:** Jeder azyklische Graph  $G = (V, E)$  mit mindestens einem Knoten enthält eine Senke.

*Beweis:* Wir können annehmen, dass  $G$  mindestens eine Kante hat (ansonsten sind alle Knoten Senken). Sei  $P$  ein Pfad maximaler Länge in  $G$  und sei  $v$  der Endknoten von  $P$ . Der Knoten  $v$  kann keinen Nachfolger in  $V(P)$  haben, da  $G$  azyklisch ist. Er kann auch keinen Nachfolger in  $V \setminus V(P)$ , da  $P$  ein Pfad maximaler Länge ist. Also kann  $v$  keine Nachfolger haben und ist damit eine Senke.  $\square$

**Satz:** Jeder azyklische Graph  $G = (V, E)$  hat eine topologische Sortierung.

*Beweis:* Wir verwenden Induktion über die Anzahl der Knoten  $n$ . Der Fall  $n = 1$  ist trivial (jede Aufzählung der Knoten eine topologische Sortierung). Falls  $n > 1$ , verwenden wir den vorherigen Satz um eine Senke  $v$  in  $G$  zu erhalten und erhalten einen azyklischen Graphen  $G'$  mit  $n - 1$  Knoten, indem wir  $v$  und dessen inzidenten Kanten von  $G$  entfernen. Laut Induktionshypothese hat  $G'$  eine topologische Sortierung  $v_1, \dots, v_{n-1}$ . Dann ist  $v_1, \dots, v_{n-1}, v$  eine topologische Sortierung von  $G$ .  $\square$

## Definition: Zusammenhang

In einem Graph  $G = (V, E)$  heisst eine Knotenmenge  $U \subseteq V$  (*stark*) *zusammenhängend*,<sup>46</sup> falls  $G$  für jedes Paar von Knoten  $u \neq v \in U$  einen Weg von  $u$  nach  $v$  enthält.

Ein Graph  $G = (V, E)$  heisst (*stark*) *zusammenhängend*, falls  $V$  in  $G$  zusammenhängend ist.

## Eigenschaft: Zusammenhangskomponenten in ungerichteten Graphen

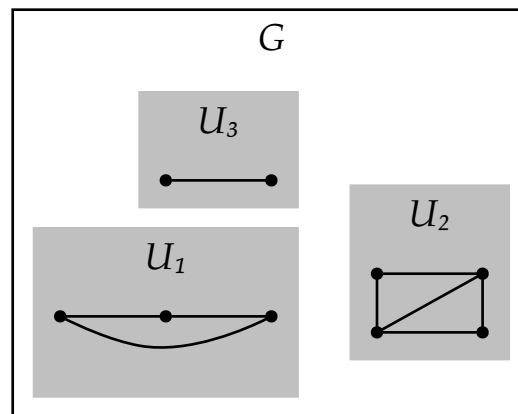
In ungerichteten Graphen haben zusammenhängende Knotenmenge eine besonders einfache Struktur, die durch den folgenden Satz ausgedrückt wird. (Für gerichtete Graphen gibt es eine ähnliche Struktur, auf die wir an dieser Stelle nicht weiter eingehen.)

---

<sup>46</sup>Wir verwenden diese Definition von Zusammenhang sowohl für gerichtete als auch für ungerichtete Graphen. Für gerichtete Graphen wird hier oft der Begriff "starker Zusammenhang" verwendet. Es gibt für gerichtete Graphen auch den Begriff "schwacher Zusammenhang", auf den wir in diesem Text allerdings nicht eingehen.

**Satz:** Für jeden ungerichteten Graph  $G = (V, E)$  gibt es eine Partition  $U_1, \dots, U_k$  der Knotenmenge  $V$ , so dass jede Menge  $U_i$  zusammenhängend in  $G$  ist und jede Kante  $(u, v) \in E$  beide Endpunkte in derselben Menge  $U_i$  hat.

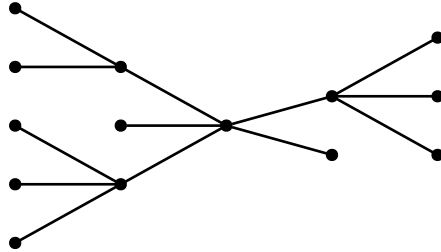
Desweiteren ist diese Partition eindeutig und die Mengen  $U_1, \dots, U_k$  heissen *Zusammenhangskomponenten*.



### Definition: Baum

Ein zusammenhängender ungerichteter Graph ohne Kreis heisst *Baum*.<sup>47</sup>

<sup>47</sup>Englisch: *tree*



Ein ungerichteter Graph dessen Zusammenhangskomponenten Bäume sind heisst *Wald*.<sup>48</sup> (Also ist jeder ungerichteter Graph ohne Kreis ein Wald.)

### Eigenschaften: Bäume und Wälder

Analog zu der Tatsache, dass jeder azyklische gerichtete Graph mindestens eine Senke hat, hat jeder Graph ohne Kreis ein Blatt.

**Satz:** Jeder ungerichtete kreislose Graph  $G$  mit mindestens zwei Knoten hat mindestens zwei Blätter.

*Beweis:* Betrachte einen Pfad  $P = (v_0, \dots, v_\ell)$  maximaler Länge in  $G$ . Die Endknoten  $v_0, v_\ell$  können keine Nachbarn ausserhalb von  $P$  haben, da der Pfad maximale Länge hat. Gleichzeitig kann  $v_0$  ausser  $v_1$  keinen Nachbarn innerhalb von  $P$ , da  $G$  keinen Kreis enthält. Es folgt, dass  $v_0$  ein Blatt ist. Auf dieselbe Weise können wir argumentieren, dass  $v_\ell$  ein Blatt ist.  $\square$

**Satz:** Jeder Baum  $T$  mit  $n$  Knoten hat  $n - 1$  Kanten.

*Beweis:* Wir führen den Beweis per Induktion  $n$ . Falls  $n = 1$ , so enthält der Baum keine Kante und erfüllt damit die gewünschte Aussage. Wir betrachten nun den Fall  $n > 1$  und können annehmen, dass die gewünschte Aussage für  $n - 1$  gilt. Laut des vorherigen Satzes enthält  $T$  mindestens ein Blatt  $v$ . Wir können einen Baum  $T'$  mit  $n - 1$  Knoten erhalten, indem wir  $v$  und dessen inzidenten Kante von  $T$  entfernen. Laut Induktionshypothese hat  $T'$  genau  $|V(T')| - 1 = n - 2$  Kanten. Da  $T$  eine Kante mehr als  $T'$  hat, gilt  $|E(T)| = |E(T')| + 1 = n - 1$  Kanten hat.  $\square$

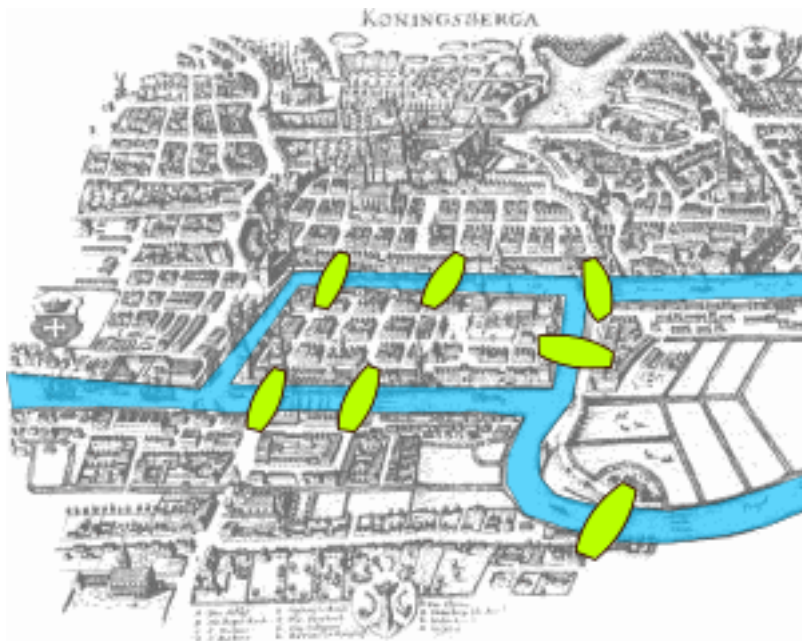
---

<sup>48</sup>Englisch: *forest*

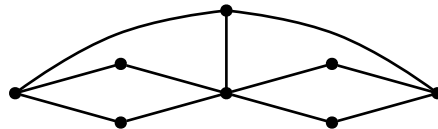
## Problemstellung: Eulertour

Leonhard Euler (geboren 1707 in Basel, gestorben 1783 in Sankt Petersburg) war einer der bedeutendsten und produktivsten Mathematiker, die je gelebt haben. Auf ihn geht folgende Fragestellung zurück:

Kann man einen Rundgang durch Königsberg machen, so dass man jede der sieben Pregelbrücken genau einmal überquert?



Graphentheoretisch formuliert fragen wir nach einem geschlossenen Weg im folgenden Graphen, der jede Kante genau einmal durchläuft.<sup>49</sup>



Euler bemerkte, dass dieser Graph keinen solchen Weg enthalten kann, da der

<sup>49</sup>Wir sagen, dass ein Weg  $v_0, \dots, v_\ell$  eine Kante  $e$  durchläuft, falls  $e = (v_i, v_{i+1})$  für ein  $i \geq 0$  gilt.



Graph Knoten mit ungeradem Grad enthält (wohingegen in jedem geschlossenen Weg jeder Knoten notwendigerweise geraden Grad hat).

Im Rest dieses Abschnitts halten wir diese Beobachtung in allgemeiner Form fest und zeigen zudem, dass zusammenhängende Graphen mit nur geraden Knotengrade einen geschlossenen Weg enthalten, der jede Kante genau einmal durchläuft.

**Definition:** Eine *Eulertour* in einem Graphen  $G$  ist ein geschlossener Weg, der jede Kante genau einmal durchläuft.

Der folgende Satz liefert eine Charakterisierung von Graphen, die Eulertouren enthalten. Der Beweis enthält die Kernidee eines effizienten Algorithmus um Eulertouren zu berechnen, auf den wir an dieser Stelle aber nicht weiter eingehen.

**Satz:** Ein ungerichteter zusammenhängender Graph enthält eine Eulertour, genau dann wenn alle Knotengrade gerade sind.

Um den Satz zu beweisen, zerlegen wir ihn in zwei Teile:

1. Jeder ungerichtete Graph mit einer Eulertour ist zusammenhängend und hat nur gerade Knotengrade.
2. Jeder ungerichtete zusammenhängende Graph mit nur geraden Knotengraden hat eine Eulertour.

Der erste Teil ist leicht einzusehen, da in einem Zyklus alle Knotengrade gerade sind.

Den zweiten Teil beweisen wir per Induktion über die Anzahl der Kanten.

**Satz:** Jeder ungerichtete zusammenhängende Graph mit nur geraden Knotengraden und höchstens  $m$  Kanten hat eine Eulertour.

*Beweis:* Falls  $m = 0$ , dann ist jeder geschlossener Weg der Länge 0 eine Eulertour und damit erfüllt der Graph die Aussage.

Wir betrachten nun den Fall  $m > 1$  und können annehmen, dass die gewünschte Aussage für  $m - 1$  gilt.

Sei  $G = (V, E)$  ein ungerichteter zusammenhängender Graph mit nur geraden Knotengraden und  $m$  Kanten. Wir können annehmen, dass  $G$  keine isolierten Knoten enthält (ansonsten können wir solche Knoten von  $G$  entfernen). Da alle Knotengrade gerade sind gilt,

$$2m = \sum_{v \in V} \deg(v) \geq 2 \cdot |V| > n - 1.$$

Also ist  $G$  kein Baum. Da  $G$  zusammenhängend ist, folgt, dass  $G$  einen Kreis  $v_0, \dots, v_\ell$  mit  $v_\ell = v_0$  und  $\ell \geq 3$  enthält. Wir erhalten einen Graphen  $G'$ , indem wir die Kante in diesem Kreis von  $G$  entfernen. Der Graph  $G'$  hat also  $m - \ell \leq m - 1$  Kanten. Ferner hat  $G'$  nur gerade Knotengrade, da sowohl  $G$  als auch der entfernte Kreis nur gerade Knotengrade haben. Laut Induktionshypothese

haben die Zusammenhangskomponenten von  $G'$  Eulertouren  $T_1, \dots, T_k$ . Wir erhalten nun eine Eulertour für  $G$ , indem wir die Touren  $T_1, \dots, T_k$  mit dem Kreis  $v_0, \dots, v_\ell$  kombinieren.  $\square$

