Theoretische Informatik

Vorlesungsskript

Zoltán Ádám Mann und András Recski

30. September 2014

Vorwort

Dieses Vorlesungsskript basiert weitgehend auf [3]. Weitere Quellen sind [1, 2, 4, 5].

Leserkreis:
Studenten in der Deutschsprachigen Ingenieurausbildung der Technischen und Wirtschaftswissenschaftlichen Universität Budapest. Das Vorlesungsskript ist für folgende Vorlesungen bestimmt:

- Studienrichtung Informatik:
  - „Einführung in die theoretische Informatik I"
  - „Einführung in die theoretische Informatik II"
  - „Algorithmentheorie"
- Studienrichtung Elektroingenieurwesen:
  - „Grundlagen der theoretischen Informatik"

Zielsetzung:
Das Vorlesungsskript wird in seiner endgültigen Fassung alle für die Prüfung benötigten Definitionen, Sätze, Algorithmen usw. enthalten. Damit soll es Studenten, die die relevanten Sachverhalte in der Vorlesung verstanden haben, die Prüfungsvorbereitung erleichtern. Es ist nicht Ziel des Vorlesungsskripts, die Vorlesung zu ersetzen.

Änderungen des Dokuments:
Dieses Dokument wird noch erweitert und möglicherweise auch geändert, um die Qualität weiter zu erhöhen. Die Autoren sind dankbar für entsprechende Hinweise (z.B. gefundene Fehler). Die jeweils aktuelle Version des Dokuments ist unter `http://www.cs.bme.hu/~mann/edu/bsz/` zu finden.

Inhaltsverzeichnis

Kapitel 0

Grundlegendes

$\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ bezeichnet die Menge der natürlichen Zahlen. $\mathbb{R}$ bezeichnet die Menge der reellen Zahlen. $\mathbb{R}^+$ bezeichnet die Menge der positiven reellen Zahlen, $\mathbb{R}_0^+$ die Menge der nicht negativen reellen Zahlen.

Mit i, j, k, m und n werden – wenn nicht explizit anders angegeben – natürliche Zahlen bezeichnet.

Sei M eine endliche Menge. Die Teilmenge $T \subseteq M$ ist *maximal* hinsichtlich einer Eigenschaft, wenn T die Eigenschaft besitzt und es keine Teilmenge T' mit $T \subsetneq T' \subseteq M$ gibt, die die Eigenschaft auch besitzen würde. $T \subseteq M$ ist eine *Teilmenge maximaler Mächtigkeit* hinsichtlich einer Eigenschaft, wenn T die Eigenschaft besitzt und für alle $T' \subseteq M$, die die Eigenschaft besitzen, $|T'| \leq |T|$ gilt. Es ist klar, dass eine Teilmenge maximaler Mächtigkeit auch maximal ist, aber umgekehrt gilt das nicht. Ähnlicherweise müssen auch minimale Teilmengen und Teilmengen minimaler Mächtigkeit unterschieden werden.

Seien A und B endliche Mengen, $f : A \to B$ eine eineindeutige Abbildung. Dann gilt $|A| = |B|$.

Definition 0.1. Seien $f, g : \mathbb{N} \to \mathbb{R}$. $f = \mathcal{O}(g)$, wenn $\exists c, n_0 > 0 \quad \forall n \geq n_0 \quad |f(n)| \leq c|g(n)|$.

Das Ende eines Beweises wird mit dem Symbol $\square$ markiert. Sätze, deren Beweis nicht angegeben wird (z.B. weil der Beweis zu lang oder nicht relevant ist), werden mit dem Symbol $\boxtimes$ gekennzeichnet.

Kapitel 1

Elementare Kombinatorik

1.1 Abzählprobleme

Definition 1.1 (Permutation)**.** Sei M eine endliche Menge. Die möglichen Reihenfolgen der Elemente von M sind die *Permutationen* von M.

Definition 1.2 (Fakultät)**.** Sei $n > 0$. $n! := n \cdot (n-1) \cdot \ldots \cdot 2 \cdot 1$. Des Weiteren sei $0! := 1$.

Satz 1.3. Eine Menge mit n Elementen hat $n!$ Permutationen.

Beweis. Wie viele Möglichkeiten hat man, eine Reihenfolge der n Elemente festzulegen? Für die erste Stelle der Reihenfolge kommen alle n Elemente in Frage, d.h. dafür hat man n Möglichkeiten. Da damit ein Element schon seinen Platz in der Reihenfolge gefunden hat, gibt es für die zweite Stelle nur noch $n-1$ Möglichkeiten, unabhängig davon, welches konkrete Element für die erste Stelle ausgewählt wurde. Für die dritte Stelle gibt es nur noch $n-2$ Möglichkeiten, usw. Für die vorletzte Stelle gibt es noch zwei Möglichkeiten, für die letzte Stelle nur noch eine. Damit ist die Anzahl der Möglichkeiten eben $n \cdot (n-1) \cdot \ldots \cdot 2 \cdot 1$. □

Satz 1.4 (Stirling-Formel)**.** $n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$. ⊠

Definition 1.5 (Permutation mit Wiederholung)**.** Sei M eine endliche Menge, in der einige Elemente nicht unterscheidbar sind: $M = M_1 \cup \ldots \cup M_m$, wobei die M_i Mengen paarweise disjunkt und die Elemente innerhalb einer M_i nicht unterscheidbar sind. Elemente in verschiedenen M_i Mengen sind unterschiedlich. Die möglichen Reihenfolgen der Elemente von M sind die *Permutationen mit Wiederholung* von M.

Satz 1.6. Seien M und $M_1, \ldots, M_m$ wie in Definition 1.5. Sei $|M_i| = k_i$, $|M| = n$. Dann ist die Anzahl der unterschiedlichen Permutationen mit Wiederholung:

$$\frac{n!}{k_1! \cdot \ldots \cdot k_m!}.$$

Beweis. Die Anzahl aller Permutationen ist gemäß Satz 1.3 $n!$. Permutationen, die sich nur in der Reihenfolge von nicht unterscheidbaren Elementen unterscheiden, wurden dabei mehrfach gezählt, gelten jetzt aber nicht als unterschiedlich. Wie oft wurde dabei jede Permutation mit Wiederholung gezählt? Genau so oft, wie die Anzahl der Möglichkeiten, die man hat, die nicht unterscheidbaren Elemente untereinander zu sortieren. Dafür gibt es eben $k_1! \cdot \ldots \cdot k_m!$ Möglichkeiten. □

Definition 1.7 (Variation)**.** Sei M eine Menge mit n Elementen, $k \leq n$. Die Möglichkeiten, eine geordnete Reihenfolge aus k verschiedenen Elementen von M zu bilden, sind die *Variationen von M zur Klasse k*.

Satz 1.8. Seien M, n und k wie in Definition 1.7. Die Anzahl der Variationen von M zur Klasse k ist $n \cdot (n-1) \cdot \ldots \cdot (n-k+1) = n!/(n-k)!$.

Beweis. Analog zum Beweis von Satz 1.3: Für die erste Stelle hat man n Möglichkeiten, für die zweite Stelle $n-1$ Möglichkeiten, usw., für die k-te Stelle hat man $n-k+1$ Möglichkeiten. □

Bemerkung 1.9. Permutation ist der Spezialfall von Variation mit $k = n$.

Definition 1.10 (Variation mit Wiederholung)**.** Sei M eine Menge mit n Elementen, $k \geq 1$. Die Möglichkeiten, eine geordnete Reihenfolge der Länge k aus Elementen von M zu bilden, wobei ein Element auch mehrfach in der Reihenfolge erscheinen kann, sind die *Variationen mit Wiederholung von M zur Klasse k*.

Satz 1.11. Seien M, n und k wie in Definition 1.10. Die Anzahl der Variationen mit Wiederholung von M zur Klasse k ist n^k.

Beweis. Für die erste Stelle hat man n Möglichkeiten, für die zweite Stelle wieder n Möglichkeiten, usw., für die k-te Stelle auch n Möglichkeiten. □

Korollar 1.12. Die Anzahl von 0-1-Folgen der Länge n beträgt 2^n.

Beweis. Die 0-1-Folgen der Länge n sind eben die Variationen mit Wiederholung von der Menge $\{0, 1\}$ zur Klasse n, und damit ist die Anzahl 2^n. □

Definition 1.13 (Kombination)**.** Sei M eine Menge mit n Elementen, $k \leq n$. Die Teilmengen von M mit k Elementen sind die *Kombinationen von M zur Klasse k*.

Definition 1.14 (Binomialkoeffizient, n über k, k aus n)**.** Sei $k \leq n$. Dann ist $\binom{n}{k}$ wie folgt definiert:

$$\binom{n}{k} := \frac{n!}{k! \cdot (n-k)!}.$$

Satz 1.15. Seien M, n und k wie in Definition 1.13. Die Anzahl der Kombinationen zur Klasse k ist $\binom{n}{k}$.

Beweis. Gemäß Satz 1.8 gibt es $n!/(n-k)!$ Variationen, d.h. geordnete Reihenfolgen mit k verschiedenen Elementen. Damit wurden alle Teilmengen mit k Elementen mehrfach gezählt, nämlich so oft, wie die Anzahl der Möglichkeiten, die man hat, die k Elemente zu ordnen. Da es dafür gemäß Satz 1.3 $k!$ Möglichkeiten gibt, ist die Anzahl der Kombinationen $\frac{n!}{(n-k)! \cdot k!}$. □

Korollar 1.16. Die Anzahl von 0-1-Folgen der Länge n mit genau k 1-en beträgt $\binom{n}{k}$.

Beweis. So viele Möglichkeiten hat man, die Position der k 1-en auszuwählen. □

Definition 1.17 (Kombination mit Wiederholung)**.** Sei M eine Menge mit n Elementen, $k \geq 1$. Die Möglichkeiten, insgesamt k Elemente aus M auszuwählen, wobei ein Element auch mehrfach ausgewählt werden kann, sind die *Kombinationen mit Wiederholung von M zur Klasse k*.

Satz 1.18. Seien M, n und k wie in Definition 1.17. Die Anzahl der Kombinationen mit Wiederholung von M zur Klasse k ist

$$\binom{n+k-1}{k}.$$

Beweis. Sei $M = \{a_1, \ldots, a_n\}$. Betrachten wir nun eine Kombination mit Wiederholung. Sei b_i die Anzahl, wie oft a_i darin vorkommt. Wir ordnen zu dieser Kombination mit Wiederholung einen Code von 0-en und 1-en zu, wie folgt. Der Code beginnt mit b_1 Stück 1-en, gefolgt von einer 0. Dann kommen b_2 Stück 1-en, wieder gefolgt von einer 0, usw. Am Ende stehen b_n Stück 1-en. Insgesamt also $\sum b_i = k$ Stück 1-en und $n-1$ Stück 0-en.

Es ist klar, dass das eine eineindeutige Abbildung zwischen den Kombinationen mit Wiederholung und der Menge der 0-1-Codes der Länge $n+k-1$, in denen genau k Stück 1-en vorkommen, ist. Letzteres hat gemäß Korollar 1.16 eine Mächtigkeit von $\binom{n+k-1}{k}$. □

1.2 Eigenschaften von Binomialkoeffizienten

Satz 1.19.

$$\binom{n}{k} = \binom{n}{n-k}$$

Beweis. Trivial aus Definition 1.14 □

Satz 1.20 (Binomischer Satz, Newton, 1664)**.** Für beliebige Zahlen x und y gilt:

$$(x+y)^n = \binom{n}{0}x^n + \binom{n}{1}x^{n-1}y + \binom{n}{2}x^{n-2}y^2 + \ldots + \binom{n}{n}y^n = \sum_{k=0}^{n}\binom{n}{k}x^{n-k}y^k.$$

Beweis. $(x+y)^n = (x+y)\cdot(x+y)\cdot\ldots\cdot(x+y)$, wobei das Produkt aus n Termen besteht. Wenn man diesen Ausdruck ausmultipliziert, erhält man eine Summe von Gliedern der Form $x^{n-k}y^k$. Das Glied $x^{n-k}y^k$ kommt zustande, wenn man aus $n-k$ der Terme x und aus den restlichen k Termen y zur Multiplikation auswählt. Die Anzahl solcher Kombinationen ist gemäß Satz 1.15 eben $\binom{n}{k}$. □

Satz 1.21.

$$\sum_{k=0}^{n}\binom{n}{k} = 2^n$$

Beweis 1. Folgt aus Satz 1.20, wenn $x = y = 1$. □

Beweis 2. Wir bestimmen auf zwei verschiedenen Wegen die Anzahl der 0-1-Folgen der Länge n. Einerseits ist das gemäß Korollar 1.12 eben 2^n. Andererseits können wir die 0-1-Folgen nach der Anzahl der enthaltenen 1-en gruppieren. Es gibt laut Korollar 1.16 $\binom{n}{k}$ 0-1-Folgen mit k 1-en, daher ist die Anzahl aller 0-1-Folgen $\sum_{k=0}^{n}\binom{n}{k}$. Das Ergebnis der beiden Berechnungen muss gleich sein, also $2^n = \sum_{k=0}^{n}\binom{n}{k}$. □

Satz 1.22. Sei $1 \leq k \leq n$. Dann gilt:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

Beweis. Laut Korollar 1.16 ist $\binom{n}{k}$ die Anzahl aller 0-1-Folgen der Länge n, die genau k 1-en enthalten. Wir bestimmen jetzt auch auf einem anderen Weg dieselbe Anzahl. Die betrachteten 0-1-Folgen können in zwei Gruppen aufgeteilt werden, je nach dem, mit welcher Ziffer sie anfangen. In der ersten Gruppe sind die 0-1-Folgen, die mit einer 1 anfangen und an den weiteren $n-1$ Stellen $k-1$ weitere 1-en enthalten. Die Anzahl solcher 0-1-Folgen ist dementsprechend $\binom{n-1}{k-1}$. In der zweiten Gruppe sind die 0-1-Folgen, die mit einer 0 beginnen und an den weiteren $n-1$ Stellen k 1-en enthalten. Die Anzahl solcher 0-1-Folgen ist $\binom{n-1}{k}$. Damit ist die Anzahl aller gesuchten 0-1-Folgen $\binom{n-1}{k-1} + \binom{n-1}{k}$. Das Ergebnis der beiden Berechnungen muss gleich sein, also $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$. □

Basierend auf diesem Satz kann man die Binomialkoeffizienten mit einer Rekursion berechnen und im so genannten Pascalschen Dreieck darstellen (siehe Abbildung 1.1).

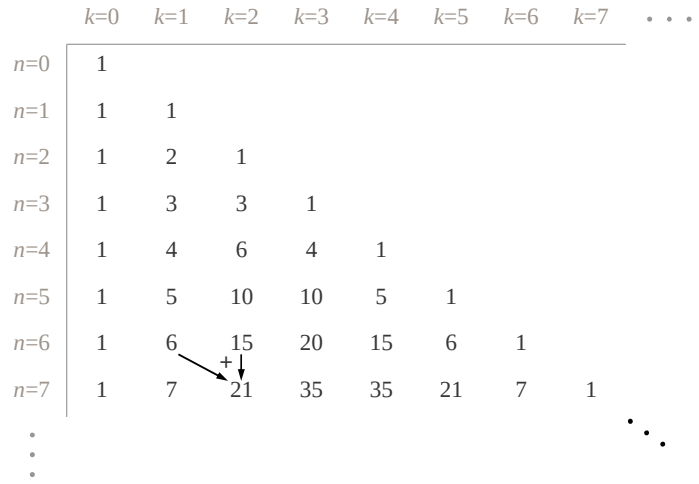

Abbildung 1.1: Die Werte von $\binom{n}{k}$ im Pascalschen Dreieck

1.3 Homogene lineare Rekursionen

Definition 1.23 (homogene lineare Rekursion, characteristische Gleichung)**.** Sei a_n ($n = 0, 1, 2, \ldots$) eine Zahlenfolge. Die Rekursionsgleichung

$$a_n = b_1 a_{n-1} + b_2 a_{n-2} \quad (n \geq 2) \tag{1.1}$$

zusammen mit den Anfangsbedingungen

$$a_0 = c_1,\ a_1 = c_2 \tag{1.2}$$

(b_1, b_2, c_1 und c_2 sind Konstanten) ist eine *homogene lineare Rekursion zweiter Ordnung*. Die *characteristische Gleichung* der Rekursion ist $x^2 - b_1 x - b_2 = 0$.

Definition 1.24 (Fibonacci-Zahlen)**.** Die Zahlenfolge F_n, definiert durch die homogene lineare Rekursion $F_n = F_{n-1} + F_{n-2}$, $F_0 = 0$, $F_1 = 1$, ist die *Fibonacci-Folge*.

Satz 1.25. Sei die Zahlenfolge a_n durch die Gleichungen (1.1)-(1.2) gegeben. Nehmen wir an, die characteristische Gleichung der Rekursion hat zwei verschiedene reelle Lösungen q_1 und q_2. Dann ist $a_n = d_1 q_1^n + d_2 q_2^n$, wobei $d_1 = \frac{c_2 - c_1 q_2}{q_1 - q_2}$ und $d_2 = \frac{c_1 q_1 - c_2}{q_1 - q_2}$.

Beweis. Mit vollständiger Induktion sieht man sofort, dass die Zahlenfolge durch die Gleichungen (1.1)-(1.2) eindeutig bestimmt ist. Betrachten wir zuerst nur die Gleichung (1.1). Wenn zwei Zahlenfolgen a_n und a'_n die Gleichung erfüllen und d_1 und d_2 beliebige Konstanten sind, dann ist es klar, dass die Zahlenfolge $d_1 a_n + d_2 a'_n$ die Gleichung auch erfüllt. Versuchen wir nun die Lösung der Gleichung in der Form $a_n = q^n$ zu suchen, wobei q eine noch zu bestimmende Konstante ist. Dann muss gelten: $q^n = b_1 q^{n-1} + b_2 q^{n-2}$, wofür $q^2 = b_1 q + b_2$ eine hinreichende (und für $q \neq 0$ auch notwendige) Bedingung ist. D.h. wenn q eine Lösung der characteristischen Gleichung ist, dann erfüllt $a_n = q^n$ Gleichung (1.1). Daraus folgt, dass $a_n = d_1 q_1^n + d_2 q_2^n$ die Gleichung auch erfüllt. Aus Gleichung (1.2) erhalten wir folgende Gleichungen für d_1 und d_2: $d_1 + d_2 = c_1$, $d_1 q_1 + d_2 q_2 = c_2$. Aus diesen beiden Gleichungen können wir d_1 und d_2 ermitteln: $d_1 = \frac{c_2 - c_1 q_2}{q_1 - q_2}$ und $d_2 = \frac{c_1 q_1 - c_2}{q_1 - q_2}$. □

Korollar 1.26.

$$F_n = \frac{1}{\sqrt{5}} \left(\frac{1 + \sqrt{5}}{2} \right)^n - \frac{1}{\sqrt{5}} \left(\frac{1 - \sqrt{5}}{2} \right)^n$$

□

Kapitel 2

Graphentheorie

2.1 Grundbegriffe

Definition 2.1 (Graph, Knoten, Kante)**.** Sei V eine nicht leere endliche Menge. Ein *Graph* G ist ein Paar (V, E), wobei die Elemente von E 2-elementige Teilmengen von V sind. Die Elemente von V sind die *Knoten* des Graphen (auch Knotenpunkte, Punkte oder Ecken genannt), die Elemente von E die *Kanten*. Die Anzahl der Knoten wird mit n, die Anzahl der Kanten mit m bezeichnet.

Definition 2.2. Sei $e = \{x, y\}$ eine Kante (kürzere Notation: $e = xy$). Dann sind x und y die *Endknoten* von e; e ist *inzident* zu x und y bzw. *verbindet* sie. Wenn $x = y$, dann ist e eine *Schlinge*. Wenn für zwei Kanten gilt, dass sie die gleichen Endknoten haben, dann sind sie *Mehrfachkanten* (auch Parallelkanten genannt). Enthält ein Graph weder Schlingen noch Mehrfachkanten, ist es ein *einfacher* Graph (oder: schlichter Graph). Wenn zwei Knoten mit einer Kante verbunden sind, dann sind sie *adjazent* (*benachbart*). Sind alle Knotenpaare eines einfachen Graphen benachbart, so handelt es sich um einen *vollständigen* Graphen. Ein vollständiger Graph mit n Knoten wird mit K_n bezeichnet.

Bemerkung 2.3. Streng genommen erlaubt Definition 2.1 weder Schlingen noch Mehrfachkanten. Da diese jedoch oft nützlich sind, werden wir sie trotzdem erlauben. Die Definition könnte man auch entsprechend erweitern, dies würde sie aber nur verkomplizieren, weshalb hier darauf verzichtet wird.

Satz 2.4. K_n hat $\binom{n}{2}$ Kanten. Ein einfacher Graph mit n Knoten hat höchstens $\binom{n}{2}$ Kanten.

Beweis. Aus n Knoten können laut Satz 1.15 $\binom{n}{2}$ Paare gebildet werden. Wenn alle Paare mit einer Kante verbunden sind, hat der Graph so viele Kanten, sonst weniger. □

Korollar 2.5. In einfachen Graphen gilt: $m = \mathcal{O}\left(n^2\right)$. □

Definition 2.6. Die Menge der Nachbarn eines Knotens v wird mit $N(v)$ bezeichnet. Der *Grad* von v ist die Anzahl der zu v inzidenten Kanten (Notation: $d(v)$). Eine Schlinge erhöht den Grad um 2. Hat ein Knoten keine Nachbarn, so ist er ein *isolierter* Knoten. Der niedrigste Grad im Graphen wird durch δ, der höchste Grad durch Δ bezeichnet. Wenn der Grad von jedem Knoten k ist, ist der Graph *k-regulär*.

Satz 2.7 (Handshake-Satz)**.** In jedem Graphen gilt:

$$\sum_{v \in V} d(v) = 2m$$

Beweis. In $\sum d(v)$ wird jede Kante e doppelt gezählt: Wenn e keine Schlinge ist, dann bei ihren beiden Endknoten jeweils einmal; wenn sie eine Schlinge ist, dann bei ihrem einzigen Endknoten zweimal. □

Korollar 2.8. Die Anzahl der Knoten mit einem ungeraden Grad ist gerade.

Beweis. Laut Satz 2.7 ist $\sum d(v)$ gerade. □

Definition 2.9 (Isomorphie)**.** Die Graphen $G = (V, E)$ und $G' = (V', E')$ sind *isomorph*, wenn es eine eineindeutige Abbildung zwischen V und V' gibt, so dass zwei Knoten in G genau dann benachbart sind wenn die entsprechenden Knoten in G' benachbart sind. (Wenn zwei Knoten in G durch mehrere Kanten verbunden sind, dann müssen die entsprechenden Knoten in G' durch dieselbe Anzahl von Kanten verbunden sein.)

Definition 2.10 (Teilgraph, induzierter Teilgraph)**.** Der Graph $G' = (V', E')$ ist ein *Teilgraph* von $G = (V, E)$ (geschrieben $G' \subseteq G$), wenn $V' \subseteq V$ und $E' \subseteq E$. Wenn $G' \subseteq G$ und G' alle Kanten $xy \in E$ enthält, für die $x, y \in V'$, dann ist G' ein *induzierter Teilgraph* von G.

Bemerkung 2.11. In der Definition von $G' \subseteq G$ ist wichtig zu erkennen, dass G' auch ein Graph sein muss. D.h., wenn $xy \in E'$, dann müssen $x, y \in V'$ sein.

Definition 2.12 (Komplement)**.** Das *Komplement* $\bar{G}$ des einfachen Graphen $G = (V, E)$ ist der Graph mit Knotenmenge V, in dem zwei Knoten genau dann benachbart sind, wenn sie es in G nicht sind.

Definition 2.13 (Kantenzug, geschlossener Kantenzug, offener Kantenzug, Weg, Kreis)**.** Ein *Kantenzug* ist eine Folge $(v_0, e_1, v_1, e_2, v_2, \ldots, v_{k-1}, e_k, v_k)$ von abwechselnd Knoten und Kanten mit $e_i = v_{i-1}v_i$ für alle i. Ist $v_0 = v_k$, dann ist es ein *geschlossener Kantenzug*, sonst ein *offener Kantenzug*. Wenn in einem offenen Kantenzug alle Knoten verschieden sind, ist es ein *Weg*. Wenn die beiden Endknoten eines offenen Kantenzuges v_0 und v_k sind, dann ist es ein $v_0 \sim v_k$ Kantenzug (bzw. ein $v_0 \sim v_k$ Weg, wenn es ein Weg ist). Wenn in einem geschlossenen Kantenzug alle Knoten außer v_0 und v_k verschieden sind, ist es ein *Kreis*. Ein Graph, der keinen Kreis enthält, ist *kreisfrei*.

Definition 2.14 (Länge)**.** Die *Länge* eines Kantenzuges W, bezeichnet mit $l(W)$, ist die Anzahl der Kanten in W.

Definition 2.15 (Konkatenation von Kantenzügen)**.** Seien $Z = (v_0, e_1, v_1, e_2, v_2, \ldots, v_{k-1}, e_k, v_k)$ und $Z' = (v'_0, e'_1, v'_1, e'_2, v'_2, \ldots, v'_{\ell-1}, e'_\ell, v'_\ell)$ zwei Kantenzüge mit $v_k = v'_0$. Dann bezeichnet $Z + Z' = (v_0, e_1, v_1, e_2, v_2, \ldots, v_{k-1}, e_k, v_k, e'_1, v'_1, e'_2, v'_2, \ldots, v'_{\ell-1}, e'_\ell, v'_\ell)$ die *Konkatenation* der beiden Kantenzüge.

Bemerkung 2.16. Wenn Z ein $u \sim v$ Kantenzug und Z' ein $v \sim w$ Kantenzug ist, dann ist $Z + Z'$ offensichtlich ein $u \sim w$ Kantenzug. Wenn W ein $u \sim v$ Weg und W' ein $v \sim w$ Weg ist, dann ist $W + W'$ zwar sicherlich ein $u \sim w$ Kantenzug, aber nicht unbedingt ein Weg.

Definition 2.17 (Teilweg)**.** Sei W ein Weg, x und y zwei Knoten in W. Dann bezeichnet $W[x, y]$ den Teilweg von W zwischen x und y.

Lemma 2.18. Wenn zwei Knoten x und y ($x \neq y$) durch einen Kantenzug verbunden sind, dann sind sie auch durch einen Weg verbunden.

Beweis. Sei $(v_0, e_1, v_1, \ldots, e_k, v_k)$ ein Kantenzug mit $v_0 = x$ und $v_k = y$. Wenn das kein Weg ist, dann gibt es $0 \leq i < j \leq k$ mit $v_i = v_j$. Dann kann man den Teil des Kantenzuges zwischen i und j weglassen: $(v_0, e_1, v_1, \ldots, e_i, v_i, e_{j+1}, v_{j+1}, \ldots, e_k, v_k)$. Das Ergebnis ist ein kürzerer Kantenzug zwischen x und y. Dieses Verfahren wiederholt man solange der Kantenzug kein Weg ist. Da der Kantenzug dabei immer kürzer wird, terminiert der Algorithmus nach endlich vielen Schritten mit einem Weg. □

Definition 2.19 (zusammenhängend, Komponente)**.** $G = (V, E)$ heißt *zusammenhängend*, wenn alle Knotenpaare mit einem Weg verbunden sind. Die maximalen zusammenhängenden Teilgraphen eines Graphen sind seine *Komponenten*. Die Anzahl der Komponenten von G wird mit $c(G)$ bezeichnet.

Definition 2.20 (Schnittmenge, minimale Schnittmenge, Brücke)**.** Sei $G = (V, E)$. $X \subseteq E$ ist eine *Schnittmenge*, wenn $c(G)$ durch das Weglassen der Kanten in X erhöht wird. Eine Schnittmenge X ist eine *minimale Schnittmenge*, wenn keine der echten Teilmengen von X eine Schnittmenge ist. $e \in E$ ist eine *Brücke*, wenn $\{e\}$ eine Schnittmenge ist.

Definition 2.21 (gerichteter Graph)**.** Sei V eine nicht leere, endliche Menge, V^2 die Menge der geordneten Paare aus V. $\vec{G} = (V, \vec{E})$ ist ein *gerichteter Graph* wenn $\vec{E} \subseteq V^2$. Die Kante $e = (x, y)$ (oder kurz $e = xy$) ist von ihrem *Anfangsknoten* x zu ihrem *Endknoten* y gerichtet.

Definition 2.22 (Quelle, Senke)**.** Ein Knoten in einem gerichteten Graphen heißt *Quelle*, wenn es keine Kante gibt, deren Endknoten dieser Knoten wäre. Ein Knoten in einem gerichteten Graphen heißt *Senke*, wenn es keine Kante gibt, deren Anfangsknoten dieser Knoten wäre.

Definition 2.23 (gerichteter Weg, gerichteter Kreis)**.** $(v_0, e_1, v_1, e_2, v_2, \ldots, v_{k-1}, e_k, v_k)$ in einem gerichteten Graphen ist ein *gerichteter Weg* ($v_0 \rightsquigarrow v_k$ Weg), wenn $e_i = v_{i-1}v_i$ für alle i und $v_0, \ldots, v_k$ alle verschieden sind. $(v_0, e_1, v_1, e_2, v_2, \ldots, v_{k-1}, e_k, v_0)$ ist ein *gerichteter Kreis*, wenn $e_i = v_{i-1}v_i$ für alle i und $v_0, \ldots, v_{k-1}$ alle verschieden sind.

Definition 2.24 (stark zusammenhängend)**.** Ein gerichteter Graph heißt *stark zusammenhängend*, wenn er für alle geordneten Paare $x, y \in V$ einen $x \rightsquigarrow y$ Weg enthält.

2.2 Bäume

Definition 2.25 (Wald, Baum)**.** Ein Graph, der keinen Kreis enthält, ist ein *Wald*. Ein zusammenhängender Graph, der keinen Kreis enthält, ist ein *Baum*.

Es ist klar, dass die Komponenten eines Waldes Bäume sind.

Satz 2.26. Sei $B = (V, E)$ ein Baum, $x, y \in V$. Dann gibt es in B genau einen $x \sim y$ Weg.

Beweis. Da B zusammenhängend ist, gibt es mindenstens einen $x \sim y$ Weg. Wenn es zwei solche Wege gäbe, wäre ihre Konkatenation ein geschlossener Kantenzug, woraus man – analog zum Beweis von Lemma 2.18 – einen Kreis auswählen könnte. □

Definition 2.27 (Blatt)**.** Ein Knoten mit Grad 1 in einem Baum heißt *Blatt*.

Lemma 2.28. In einem Baum mit mindestens zwei Knoten gibt es mindestens zwei Blätter.

Beweis. Man betrachte einen längsten Weg W im Baum. Da der Graph zusammenhängend ist und aus mindestens zwei Knoten besteht, hat der längste Weg mindestens Länge 1. Wir werden beweisen, dass die Endknoten dieses Weges Grad 1 haben. Sei x ein Endknoten des Weges, sein Nachbar im Weg y. Nehmen wir indirekt an, dass x noch einen weiteren Nachbar z hat. Dieser Nachbar kann nicht in W enthalten sein, weil dann die Kante xz zusammen mit $W[x, z]$ einen Kreis bilden würde. Wenn z aber nicht in W ist, dann kann man W mit der Kante xz verlängern, was nicht sein kann, da es ein längster Weg ist. □

Satz 2.29. Ein Baum mit n Knoten hat $n - 1$ Kanten.

Beweis. Vollständige Induktion. Für $n = 1$ ist der Satz trivial. Nehmen wir nun an, dass der Satz für $n - 1$ bereits bewiesen ist und betrachten wir einen Baum B mit n Knoten ($n \geq 2$). Gemäß Lemma 2.28 hat B einen Knoten x mit Grad 1; sei e die einzige, zu x inzidente Kante. Entfernen wir x und e. Der so entstandene Graph B' ist ein Baum mit $n - 1$ Knoten. Gemäß der Induktionsannahme hat B' also $n - 2$ Kanten. Folglich hat B $n - 1$ Kanten. □

Satz 2.30. Ein Wald mit n Knoten und c Komponenten hat $n - c$ Kanten.

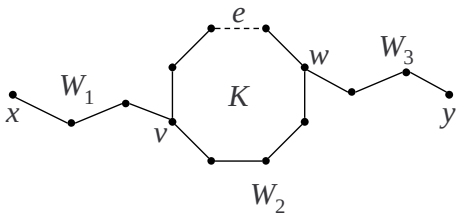

Abbildung 2.1: Zum Beweis von Lemma 2.35

Beweis. Sei W ein Wald mit n Knoten und c Komponenten. Die Komponenten von W sind Bäume mit Knotenanzahl $n_1, \ldots, n_c$. Gemäß Satz 2.29 haben diese Bäume $n_1 - 1, \ldots, n_c - 1$ Kanten. Insgesamt hat W damit

$$\sum_{i=1}^{c}(n_i - 1) = n - c$$

Kanten. □

Korollar 2.31. Ein kreisfreier Graph mit n Knoten hat höchstens $n - 1$ Kanten. Ein kreisfreier Graph mit n Knoten hat genau dann $n - 1$ Kanten wenn er ein Baum ist. □

2.2.1 Spannbäume

Definition 2.32 (Spannbaum)**.** Sei $G = (V, E)$ ein zusammenhängender Graph, $B = (V, E')$ ein Baum mit derselben Knotenmenge und $E' \subseteq E$. Dann ist B ein *Spannbaum* von G.

Definition 2.33 (Spannwald)**.** Sei $G = (V, E)$ ein Graph, $W = (V, E')$ ein Wald mit derselben Knotenmenge und $E' \subseteq E$. Wenn W in jeder Komponente von G einen Spannbaum ergibt, dann ist W ein *Spannwald* von G.

Satz 2.34. Sei $B = (V, E')$ ein Spannbaum des Graphen $G = (V, E)$. Sei $e \in E \backslash E'$ und sei $B' := B \cup \{e\}$. Dann enthält B' genau einen Kreis.

Beweis. Sei $e = xy$, wobei $x, y \in V$. Gemäß Satz 2.26 gibt es einen $x \sim y$ Weg W in B. $W + e$ ist ein Kreis in B'. Andererseits, gäbe es zwei verschiedene Kreise K_1 und K_2 in B', dann müssten beide e enthalten, so dass $K_1 \setminus \{e\}$ und $K_2 \setminus \{e\}$ zwei verschiedene $x \sim y$ Wege in B wären, was wieder wegen Satz 2.26 nicht möglich ist. □

Lemma 2.35. Sei $G = (V, E)$ ein zusammenhängender Graph, $K \subseteq G$ ein Kreis, e eine Kante im Kreis. Entfernen wir e, der resultierende Graph sei G'. Dann ist G' auch zusammenhängend.

Beweis. Nehmen wir indirekt an, G' ist nicht zusammenhängend. Dann gibt es $x, y \in V$, zwischen denen in G' kein Weg läuft. In G gab es allerdings einen Weg W zwischen x und y. Das kann nur sein wenn $e \in W$. Sei v der erste Knoten auf W von x kommend, der in K enthalten ist, w der letzte solche Knoten. Sei $W_1 := W[x, v]$, $W_3 := W[w, y]$. Zwischen v und w gibt es zwei Wege innerhalb von K, der eine enthält e, der andere nicht. Letzteres bezeichnen wir mit W_2. Dann ergeben W_1, W_2 und W_3 zusammen einen Weg in G' von x zu y (siehe Abbildung 2.1), was der Annahme widerspricht. □

Satz 2.36. Ein Graph ist genau dann zusammenhängend wenn er einen Spannbaum hat.

Beweis. Es ist trivial, dass der Graph zusammenhängend sein muss, um einen Spannbaum zu haben. Jetzt widmen wir uns der anderen Richtung. Sei $G = (V, E)$ ein zusammenhängender Graph, $B = (V, E')$ ein minimaler zusammenhängender Teilgraph von G mit derselben Knotenmenge. Dann ist B ein Spannbaum von G. Das Einzige, was dazu bewiesen werden muss, ist dass B kreisfrei ist. Würde B einen Kreis enthalten, so könnte man gemäß Lemma 2.35 eine beliebige Kante aus dem Kreis entfernen, und der entstehende Teilgraph B' wäre ein kleinerer zusammenhängender Teilgraph von G mit derselben Knotenmenge, was jedoch der Minimalität von B widerspricht. □

Korollar 2.37. Ein zusammenhängender Graph mit n Knoten hat mindestens $n - 1$ Kanten. Ein zusammenhängender Graph mit n Knoten hat genau dann $n - 1$ Kanten, wenn er ein Baum ist. □

2.2.2 Spannbaum mit minimalen Kosten

Definition 2.38. Im Folgenden sei $G = (V, E)$ ein zusammenhängender Graph. Gegeben ist zudem eine Funktion $k : E \to \mathbb{R}$; $k(e)$ gibt die *Kosten* der Kante e an. Wenn $E' \subseteq E$, dann bezeichne $k(E') := \sum_{e \in E'} k(e)$ die Kosten von E'. Ein Spannbaum $B = (V, E')$ von G ist ein *Spannbaum mit minimalen Kosten*, wenn $k(E')$ unter allen Spannbäumen von G minimal ist. $k(E')$ wird auch mit $k(B)$ bezeichnet.

Algorithmus 2.39 (Kruskal-Algorithmus).
`Input:`
G und k wie in Definition 2.38.
`Output:`
Ein Spannbaum von G mit minimalen Kosten.
`Ablauf:`
Wir fangen mit $(V, \emptyset)$ an. In jedem Schritt nehmen wir eine neue Kante aus E dazu, und zwar eine, die mit den bisher ausgewählten Kanten keinen Kreis bildet. Wenn es mehrere solche Kanten gibt, nehmen wir die mit den niedrigsten Kosten. Der Algorithmus terminiert wenn es keine solche Kante mehr gibt.

Bemerkung 2.40. Der Kruskal-Algorithmus wird als *gieriger Algorithmus* bezeichnet, denn er macht in jedem Schritt das, was in dem Moment das Beste zu sein scheint, ohne zu bedenken, dass ein aktuell schlechterer Schritt langfristig vorteilhaft sein könnte. Wie wir gleich sehen werden, führt diese gierige Strategie bei diesem Problem zum optimalen Ergebnis, aber bei vielen anderen Problemen ist das nicht der Fall.

Satz 2.41. Der Kruskal-Algorithmus liefert einen Spannbaum mit minimalen Kosten.

Beweis. Es ist klar, dass der Algorithmus einen Spannbaum liefert, denn wir achten darauf, keinen Kreis zu machen und solange die ausgewählten Kanten noch keinen zusammenhängenden Teilgraphen bilden oder noch nicht alle Knoten überdecken, ist es immer möglich, noch eine weitere Kante auszuwählen.

Sei der vom Algorithmus gelieferte Spannbaum $B_A = (V, E_A)$. Nehmen wir indirekt an, es gibt einen Spannbaum $B_m = (V, E_m)$ mit minimalen Kosten, mit $k(B_m) < k(B_A)$. Wenn es mehrere gibt, dann nehmen wir einen, bei dem die Anzahl der mit B_A gemeinsamen Kanten maximal ist.

Sei $e \in E_m \setminus E_A$. $B_A \cup \{e\}$ enthält einen Kreis K, der aus $e = xy$ und einem $x \sim y$ Weg $W \subseteq B_A$ besteht. Für eine beliebige Kante e' dieses Weges gilt, dass $k(e') \leq k(e)$, weil der Algorithmus sonst e statt e' gewählt hätte.

Würde man e aus B_m entfernen, so würde dieser in zwei Komponenten zerfallen. W läuft zwischen diesen beiden Komponenten, daher gibt es mindestens eine Kante $e' \in W$, die zwischen den beiden Komponenten läuft (siehe Abbildung 2.2). Dann muss aber $k(e') \geq k(e)$ gelten, sonst wäre $B'_m := B_m \setminus \{e\} \cup \{e'\}$ ein Spannbaum mit niedrigeren Kosten als B_m.

Aus diesen beiden Überlegungen folgt, dass $k(e') = k(e)$. Dann ist aber B'_m auch ein Spannbaum mit minimalen Kosten, mit $k(B'_m) = k(B_m) < k(B_A)$, aber B'_m hat eine Kante mehr gemeinsam mit B_A als B_m (nämlich e'). Das widerspricht aber der Wahl von B_m. □

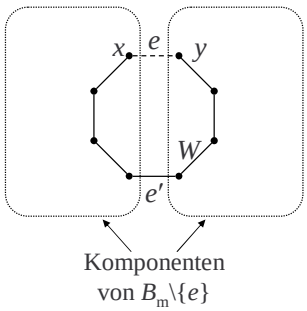

Abbildung 2.2: Zum Beweis von Satz 2.41

Korollar 2.42. Wenn man den Kruskal-Algorithmus auf einen nicht unbedingt zusammenhängenden Graphen anwendet, erhält man einen Spannwald mit minimalen Kosten.

Beweis. Der Algorithmus liefert in jeder Komponente des Graphen einen minimalen Spannbaum. Diese ergeben zusammen einen minimalen Spannwald. □

2.2.3 Anzahl verschiedener Bäume bei gegebener Knotenmenge

Im Folgenden sei $V = \{1, 2, \ldots, n\}$.

Algorithmus 2.43 (Prüfer-Code).
`Input`:
Ein Baum $B = (V, E)$.
`Output`:
Eine Folge der Länge $n - 2$ aus $\{1, 2, \ldots, n\}$, bezeichnet als der *Prüfer-Code* des Baumes (Notation: $PC(B)$).
`Ablauf`:
In jedem Schritt betrachtet man das Blatt mit der niedrigsten Nummer (laut Lemma 2.28 gibt es mindestens zwei Blätter). Sei das in Schritt i der Knoten x_i, sein einziger Nachbar sei y_i. Dann steht auf Platz i in $PC(B)$ die Nummer y_i. Der Knoten x_i und die Kante $x_i y_i$ werden entfernt und der Algorithmus läuft weiter. Dabei geht i von 1 bis $n - 2$. Macht man noch einen weiteren Schritt (d.h. geht i von 1 bis $n - 1$), so erhält man den $n - 1$ langen *erweiterten Prüfer-Code* $ePC(B)$.

Lemma 2.44. $ePC(B)$ ist $PC(B)$ ergänzt um n am Ende.

Beweis. Solange der Baum mindestens zwei Knoten hat, hat er auch mindestens zwei Blätter. Da immer das Blatt mit der kleinsten Nummer entfernt wird, wird n während des ganzen Algorithmus nicht entfernt. Nach $n - 2$ Schritten sind also n und noch ein Nachbar von n vorhanden. Im letzten Schritt wird dieser Nachbar entfernt und n wird an die letzte Stelle von $ePC(B)$ geschrieben. □

Lemma 2.45. Sei B ein Baum, v ein Knoten mit $d(v) = k$. Dann kommt v genau $k - 1$-mal in $PC(B)$ vor.

Beweis. Beim Algorithmus zur Bestimmung von $PC(B)$ werden alle Knoten außer n und eines weiteren, den wir mit n' bezeichnen, entfernt. Nehmen wir zuerst an, dass $v \notin \{n, n'\}$ und betrachten wir den Zeitpunkt in dem v entfernt wird. Zu diesem Zeitpunkt ist $d(v) = 1$. Das heißt, dass $k - 1$ Nachbarn von v bereits entfernt worden sind. Daher wurde v genau $k - 1$-mal in $PC(B)$ geschrieben. Da v nun entfernt wird, wird es kein weiteres Mal in $PC(B)$ geschrieben.

Bei n und n' ist es ähnlich: am Ende besteht der Baum nur noch aus der Kante nn', so dass n und n' am Ende auch Grad 1 haben. Daher wurden sie $d(n) - 1$-mal bzw. $d(n') - 1$-mal in $PC(B)$ geschrieben. □

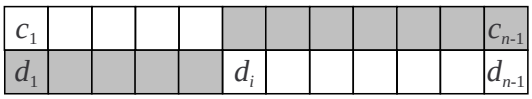

Abbildung 2.3: Zum Algorithmus für die Wiederherstellung eines Baumes aus seinem Prüfer-Code

Algorithmus 2.46 (Dekodierung des Prüfer-Codes).
`Input`:
$C = (c_1, c_2, \ldots, c_{n-2})$ eine Folge der Länge $n-2$ aus $\{1, 2, \ldots, n\}$.
`Output`:
Graph $G(C)$.
`Ablauf`:
Wir ergänzen zuerst C mit $c_{n-1} = n$.
Sei d_i ($i = 1, 2, \ldots, n-1$) die kleinste positive ganze Zahl, die unter $\{d_1, \ldots, d_{i-1}, c_i, c_{i+1}, \ldots, c_{n-1}\}$ nicht vorkommt (da das nur $n-1$ Zahlen sind, ist $d_i \leq n$. Siehe auch Abbildung 2.3).
Der Graph $G(C)$ hat genau die Kanten $c_i d_i$ ($i = 1, 2, \ldots, n-1$).

Lemma 2.47. Sei $C = (c_1, c_2, \ldots, c_{n-2})$ eine Folge der Länge $n-2$ aus $\{1, 2, \ldots, n\}$. Sei $G(C)$ der Graph, der durch obigen Algorithmus erstellt wird. Dann ist $G(C)$ ist ein Baum.

Beweis. $G(C)$ hat n Knoten und $n-1$ Kanten. Laut Korollar 2.31 reicht es zu beweisen, dass $G(C)$ kreisfrei ist. Nehmen wir indirekt an, dass einige $c_i d_i$ Paare einen Kreis bilden. Sei $c_k d_k$ das Paar mit dem niedrigsten Index im Kreis. Da das ein Kreis ist, muss d_k auch noch in einem zweiten Paar enthalten sein. D.h. entweder $d_k = d_j$ oder $d_k = c_j$ muss für ein $j > k$ gelten. Ersteres ist nicht möglich, weil bei der Bestimmung von d_j alle d_is mit $i < j$ ausgeschlossen sind, und zweiteres ist nicht möglich, weil bei der Bestimmung von d_k alle c_is mit $i > k$ ausgeschlossen sind. □

Lemma 2.48. Seien C und $G(C)$ wie in Lemma 2.47. Dann gilt $PC(G(C)) = C$ und $G(C)$ ist der einzige Baum B mit $PC(B) = C$. ⊠

Satz 2.49 (Cayley)**.** Die Anzahl verschiedener Bäume mit einer gegebenen Knotenmenge ist n^{n-2}. Anders ausgedrückt: K_n hat n^{n-2} verschiedene Spannbäume. (Dabei werden zwei Bäume $B_1 = (V, E_1)$ und $B_2 = (V, E_2)$ mit $E_1 \neq E_2$ als verschieden angesehen, auch wenn sie isomorph sind.)

Beweis. Man kann ohne Beschränkung der Allgemeinheit annehmen, dass $V = \{1, 2, \ldots, n\}$. Sei $\mathcal{B}$ die Menge aller Bäume mit dieser Knotenmenge und sei $\mathcal{C}$ die Menge der Folgen der Länge $n-2$ aus $\{1, 2, \ldots, n\}$. Gemäß Lemma 2.48 definiert $PC(\cdot)$ eine eineindeutige Abbildung zwischen $\mathcal{B}$ und $\mathcal{C}$. Damit ist $|\mathcal{B}| = |\mathcal{C}| = n^{n-2}$. □

2.2.4 Analyse elektrischer Netzwerke

Bemerkung 2.50. Ein aus Zweipolelementen aufgebautes elektrisches Netzwerk kann als ungerichteter, zusammenhängender Graph aufgefasst werden. Für jede Kante des Graphen gibt es 2 Unbekannten: die Stromstärke und die Spannung. Es gibt folgende Gleichungen:

- Die Ohmsche Gleichung für jede Kante.
- Die Kirchhoffsche Knotengleichung für jeden Knoten.
- Die Kirchhoffsche Maschengleichung für jeden Kreis.

Satz 2.51. Die Ohmschen Gleichungen der einzelnen Kanten sind unabhängig voneinander.

Beweis. Trivial, da jede Gleichung andere Variablen enthält. □

Satz 2.52. Die Kirchhoffschen Knotengleichungen der einzelnen Knoten sind nicht unabhängig. Aber von den n Gleichungen sind beliebige $n-1$ unabhängig.

Beweis. Wenn man alle Gleichungen summiert, dann kommt in der Summe die Stromstärke jeder Kante zweimal vor, einmal mit positivem und einmal mit negativem Vorzeichen, so dass die resultierende Gleichung $0 = 0$ ist. Also ist die Menge aller Kirchhoffschen Knotengleichungen nicht unabhängig.

Umgekehrt, nehmen wir an, dass für $\emptyset \neq X \subsetneq V$ und entsprechende Koeffizienten, die nicht gleich 0 sind, die lineare Kombination der Knotengleichungen $0 = 0$ ergibt. Wenn $x \in X$ und $xy \in E$, dann muss auch $y \in X$ sein, weil sonst die Stromstärke der Kante xy in der Summe nicht verschwinden könnte. Da der Graph zusammenhängend ist, folgt daraus, dass $X = V$. Folglich ist jede Menge von weniger als n Knotengleichungen unabhängig. □

Satz 2.53. Sei B ein Spannbaum. Wenn man für alle Kanten $e \notin B$ die Maschengleichung für den einzigen Kreis in $B \cup \{e\}$ aufschreibt, so erhält man $m - (n-1)$ unabhängige Maschengleichungen.

Beweis. Die Spannung der Kante e kommt nur in der Gleichung für den Kreis in $B \cup \{e\}$ vor. D.h., jede Gleichung enthält eine Variable, die in keiner anderen Gleichung vorkommt. Also sind die Gleichungen unabhängig. □

Insgesamt hat man also $m + n - 1 + m - (n-1) = 2m$ Gleichungen für die $2m$ Unbekannten. Man kann beweisen, dass diese Gleichungen auch alle zusammen unabhängig sind.

2.3 Graphdurchlauf

Definition 2.54 (erreichbar)**.** Sei $G = (V, E)$ ein gerichteter / ungerichteter Graph, $x, y \in V$. y ist erreichbar aus x, wenn es einen gerichteten / ungerichteten Weg von x nach y gibt.

2.3.1 Breitensuche

Algorithmus 2.55 (Breitensuche).
`Input`:
gerichteter oder ungerichteter Graph G und Anfangsknoten $s \in V$.
`Output`:
Menge der aus s erreichbaren Knoten.
`Ablauf`:
Sei $S_0 := \{s\}$. In einer allgemeinen Iteration sei $S_k := \{$alle Knoten, die aus S_{k-1} über eine Kante erreichbar sind und noch nicht besucht wurden$\}$. Der Algorithmus terminiert wenn $S_k = \emptyset$.

Satz 2.56. (1) S_k enthält genau die Knoten, zu denen der kürzeste Weg von s aus aus genau k Kanten besteht.
(2) $\cup S_k$ enthält genau die Knoten, die von s aus erreichbar sind.
(3) Der Zeitbedarf des Algorithmus ist $\mathcal{O}(n+m)$.

Beweis. (1) Vollständige Induktion; für $k = 0$ ist die Behauptung trivial. Sei $x \in V$ ein Knoten, zu dem der kürzeste Weg W von s aus aus k Kanten besteht. Sei y der letzte Knoten in W vor x. Dann besteht der kürzeste Weg von s zu y aus $k-1$ Kanten, so dass laut Induktionsannahme $y \in S_{k-1}$. Daraus folgt, dass x spätestens in S_k aufgenommen wird. Wenn x schon in S_j, $j < k$ enthalten wäre, dann gäbe es einen anderen Knoten z in S_{j-1}, so dass $zx \in E$. Laut Induktionsbedingung besteht dann der kürzeste Weg von s zu z aus $j-1$ Kanten, so dass es einen Weg von s zu x der Länge j gäbe. Das kann jedoch wegen $j < k$

nicht sein.
(2) Folgt aus (1).
(3) Jeder von s aus erreichbare Knoten wird genau einmal besucht. Dabei werden alle aus ihm ausgehenden Kanten genau einmal untersucht. Die Anzahl der Schritte für einen Knoten v sind also $\mathcal{O}(1 + d(v))$. Damit ist die Gesamtanzahl der Schritte $\mathcal{O}\left(n + \sum_{v \in V} d(v)\right) = \mathcal{O}(n + m)$. □

2.3.2 Tiefensuche

Algorithmus 2.57 (Tiefensuche).
`Input`:
gerichteter oder ungerichteter Graph G und Anfangsknoten $s \in V$.
`Output`:
Menge der aus s erreichbaren Knoten.
`Ablauf`:
Für alle Knoten, die von s aus über eine Kante erreichbar sind und noch nicht besucht wurden, wird rekursiv die Tiefensuche mit diesem Anfangsknoten durchgeführt.
Am Ende des Algorithmus wird die Menge der besuchten Knoten zurückgegeben.

Bemerkung 2.58. Ein Knoten gilt als besucht, wenn der Aufruf der Tiefensuche für diesen Knoten gestartet wurde.

Satz 2.59. (1) Die Tiefensuche liefert genau die Knoten, die von s aus erreichbar sind.
(2) Der Zeitbedarf des Algorithmus ist $\mathcal{O}(n + m)$.

Beweis. (1) Es ist trivial, dass der Algorithmus nur Knoten besucht, die von s aus erreichbar sind.
Umgekehrt, sei $T \subseteq V$ die Menge der aus s erreichbaren Knoten, $F \subseteq T$ die Menge der vom Algorithmus gefundenen Knoten, und nehmen wir indirekt an: $\exists v \in T \setminus F$. Da $v \in T$, gibt es einen $s \rightsquigarrow v$ Weg W. Da der Anfangsknoten von W in F, aber der Endknoten von W außerhalb von F ist, gibt es zwei nacheinander kommende Knoten in W, x und y, so dass $x \in F$ und $y \notin F$. Das heißt aber, dass der Algorithmus für x aufgerufen wurde. Da $xy \in E$, hat der Algorithmus bei diesem Aufruf auch den Knoten y untersucht und dafür gesorgt, dass y auch besucht wird, also muss auch $y \in F$ gelten.
(2) Analog zum Beweis von Satz 2.56/3. □

Bemerkung 2.60. Sei G ein Graph, in dem alle Knoten von s aus erreichbar sind. Betrachten wir für jeden Knoten außer s jene Kante, über die der Algorithmus zuerst den gegebenen Knoten erreicht. Dann ist es klar, dass diese Kanten einen Spannbaum des Graphen bilden. Disesen nennen wir den vom Algorithmus gelieferten Spannbaum.

Satz 2.61. Sei G ein ungerichteter, zusammenhängender Graph, B der durch die Tiefensuche gelieferte Spannbaum, xy eine Kante, die in B nicht enthalten ist. Nehmen wir an, dass die Tiefensuche zuerst x besucht hat, y erst danach. Dann ist x ein Vorfahr von y im Baum.

Beweis. Als der Aufruf der Tiefensuche für x startet, wurde y noch nicht besucht. Da $xy \in E$, wird während des Aufrufs für x irgendwann der Knoten y untersucht. Da die Kante xy nicht in den Spannbaum aufgenommen wird, heißt das, dass y vor dieser Untersuchung besucht wurde. Also wurde y während des Aufrufs für x besucht. Daraus folgt, dass x ein Vorfahr von y ist. □

Bemerkung 2.62. Gemäß Satz 2.53 ist es eine wichtige Aufgabe, die Kanten des Kreises in $B \cup \{xy\}$ aufzählen zu können. Dies ist gemäß Satz 2.61 einfach, wenn B der durch die Tiefensuche generierte Spannbaum ist. Man muss nur die Reihenfolge, wie die Knoten besucht wurden, sowie die Kanten, über die die einzelnen Knoten beim Algorithmus zuerst erreicht wurden, speichern.

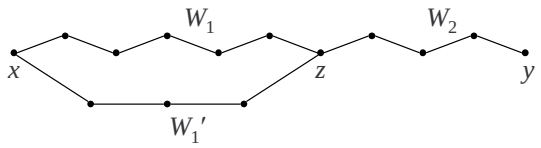

Abbildung 2.4: Zum Beweis von Korollar 2.65

2.4 Günstigste Wege

Definition 2.63. Sei $G = (V, E)$ ein gerichteter, stark zusammenhängender Graph. Für $e \in E$ bezeichne $k(e)$ die *Kosten* der Kante e. Für einen Kantenzug W bezeichne $k(W)$ die Kosten des Kantenzuges, definiert als die Summe der Kosten der enthaltenen Kanten. Der *günstigste Weg* zwischen zwei Knoten ist der Weg zwischen ihnen mit minimalen Kosten. Für $x, y \in V$ sei

$$d(x,y) := \begin{cases} 0 & \text{falls } x = y, \\ \text{Kosten des günstigsten } x \rightsquigarrow y \text{ Weges} & \text{sonst.} \end{cases}$$

2.4.1 Algorithmus von Bellman und Ford

Lemma 2.64. Sei $G = (V, E)$ ein gerichteter Graph, $k : E \to \mathbb{R}$ mit der Eigenschaft, dass es keinen gerichteten Kreis K mit $k(K) < 0$ gibt. Wenn es einen Kantenzug mit Kosten L zwischen den Knoten x und y gibt, dann gilt $d(x, y) \leq L$.

Beweis. Analog zu Lemma 2.18 kann man aus dem Kantenzug eventuelle Zyklen entfernen, bis er ein $x \rightsquigarrow y$-Weg wird. Da die Kosten der entfernten Kreise nicht negativ sind, werden dadurch die Kosten des Kantenzuges nicht erhöht, so dass es auch einen Weg mit Kosten $\leq L$ zwischen x und y gibt. □

Korollar 2.65 (Optimalitätsprinzip)**.** Sei $G = (V, E)$ ein gerichteter Graph, $k : E \to \mathbb{R}$ mit der Eigenschaft, dass es keinen gerichteten Kreis K mit $k(K) < 0$ gibt. Sei W ein günstigster Weg zwischen den Knoten x und y. Sei z ein weiterer Knoten in W. Sei $W_1 = W[x, z]$ und sei $W_2 = W[z, y]$. Dann ist W_1 ein günstigster Weg zwischen x und z.

Beweis. Nehmen wir indirekt an, es gibt einen Weg W_1' zwischen x und z mit $k(W_1') < k(W_1)$. Dann wäre aber $W_1' + W_2$ ein günstigerer Kantenzug zwischen x und y als W (siehe Abbildung 2.4). Wegen Lemma 2.64 würde daraus $d(x, y) \leq k(W_1') + k(W_2) < k(W)$ folgen. □

Definition 2.66 (D, Relax-Schritt)**.** Seien G, k wie oben, s ein ausgezeichneter Knoten von G. Sei $D : V \to \mathbb{R}$ eine Funktion, die wie folgt initialisiert wird:

$$D(v) := \begin{cases} 0 & \text{falls } v = s, \\ \infty & \text{falls } sv \notin E, \\ k(sv) & \text{sonst.} \end{cases}$$

Sei $xy \in E$, dann ist der *Relax-Schritt* für die Kante xy:

Relax(xy):
wenn $D(y) > D(x) + k(xy)$, dann sei $D(y) := D(x) + k(xy)$.

Lemma 2.67. Nach dem Aufruf von Relax(xy) gilt $D(y) \leq D(x) + k(xy)$.

Beweis. Trivial. □

Lemma 2.68. Nach einer beliebigen Sequenz von Relax-Schritten gilt für alle Knoten: $D(v) \geq d(s,v)$. Wenn für einen Knoten $D(v) = d(s,v)$ gilt, dann ändert sich $D(v)$ bei weiteren Relax-Schritten nicht mehr.

Beweis. Vollständige Induktion. Nach der Initialisierung von D gilt die Aussage. Man muss nur beweisen, dass sie nach jeder Änderung von D auch gilt. Wenn bei Relax(xy) der Wert von $D(y)$ geändert wird, dann gilt:

$$D^{neu}(y) = D(x) + k(xy) \overset{1}{\geq} d(s,x) + k(xy) \overset{2}{\geq} d(s,y).$$

(1: Induktionsannahme; 2: $d(s,x)$ ist gleich den Kosten eines $s \rightsquigarrow x$ Weges, der zusammen mit der Kante xy einen $s \rightsquigarrow y$ Kantenzug mit Kosten $d(s,x) + k(xy)$ ergibt, so dass $d(s,y)$ laut Lemma 2.64 höchstens so viel betragen kann.)

Die zweite Aussage folgt daraus, dass $D(v)$ bei einem Relax-Schritt immer nur verringert werden kann, was in diesem Fall nicht mehr möglich ist. □

Algorithmus 2.69 (Algorithmus von Bellman und Ford).
`Input`:
$G = (V,E)$ gerichteter, stark zusammenhängender Graph, $s \in V$, $k : E \to \mathbb{R}$ mit der Eigenschaft, dass es keinen gerichteten Kreis K mit $k(K) < 0$ gibt.
`Output`:
$d(s,v)$ für alle $v \in V$.
`Ablauf`:
Initialisierung: D wird initialisiert wie in Definition 2.66
Wiederhole $n-1$-mal:
    Wiederhole für jede Kante xy:
        Relax(xy).

Satz 2.70. Man betrachte einen Zwischenstand nach j Iterationen der äußeren Schleife in der Durchführung des Algorithmus. Sei v ein Knoten, für den es einen günstigsten $s \rightsquigarrow v$ Weg W gibt, der aus höchstens j Kanten besteht. Dann gilt $D(v) = d(s,v)$.

Beweis. Vollständige Induktion. Nach der Initialisierung ist $j = 0$ und s ist der einzige Knoten, der über 0 Kanten von s aus erreichbar ist. Für s gilt die Aussage auch.

Sei nun $j > 0$ und $v \neq s$. Der letzte Knoten auf W vor v sei u. Gemäß Korollar 2.65 ist $W[s,u]$ ein günstigster $s \rightsquigarrow u$ Weg, der aus höchstens $j-1$ Kanten besteht. Dann galt nach $j-1$ Iterationen laut Induktionsbedingung $D(u) = d(s,u)$ und das ändert sich im weiteren Verlauf des Algorithmus laut Lemma 2.68 nicht mehr. Daher gilt $d(s,v) = k(W) = d(s,u) + k(uv) = D(u) + k(uv)$. In der j-ten Iteration wird dafür gesorgt, dass $D(v) \leq D(u) + k(uv)$, also $D(v) \leq d(s,v)$. Zusammen mit Lemma 2.68 ergibt das die gewünschte Aussage. □

Korollar 2.71. Nach $n-1$ Iterationen der äußeren Schleife liefert der Algorithmus von Bellman und Ford für alle Knoten $D(v) = d(s,v)$.

Beweis. Da ein Weg aus höchstens $n-1$ Kanten besteht, gilt die Aussage von Satz 2.70 nach $j = n-1$ Iterationen für alle Knoten. □

Bemerkung 2.72. Wenn man noch eine weitere Iteration der äußeren Schleife durchführt, darf sich kein D-Wert mehr ändern. Andernfalls gab es doch einen gerichteten Kreis mit negativen Kosten im Graphen. D.h. mit dieser Ergänzung ist der Algorithmus in der Lage zu entscheiden, ob es einen gerichteten Kreis mit negativen Kosten gibt.

Bemerkung 2.73. Die Anzahl der Schritte beträgt $\mathcal{O}(nm)$.

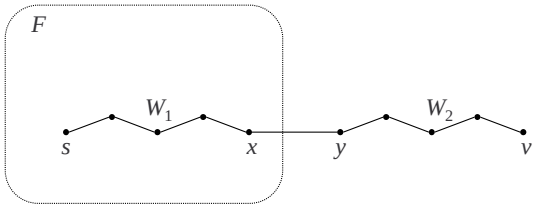

Abbildung 2.5: Zum Beweis von Satz 2.77

Bemerkung 2.74. Man kann den Algorithmus erweitern, so dass er nicht nur die Kosten der günstigsten Wege, sondern auch die günstigsten Wege selbst liefert. Dazu definiert man für jeden Knoten x einen Pointer $p(x)$, der auf den Vorgänger von x in einem günstigsten $s \rightsquigarrow x$ Weg zeigt. Bei der Initialisierung wird $p(x) := s$ gesetzt. Später wird immer, wenn im Relax-Schritt die Anweisung $D(y) := D(x) + k(xy)$ ausgeführt wird, $p(y) := x$ gesetzt. Nach der Durchführung des Algorithmus kann man einen günstigsten $s \rightsquigarrow v$ Weg rückwärts, von v zu s, über die p Pointer verfolgen.

Bemerkung 2.75. Der Algorithmus funktioniert auch für nicht stark zusammenhängende Graphen. In diesem Fall findet der Algorithmus die günstigsten Wege von s zu allen anderen Knoten, die von s aus erreichbar sind. Am Ende ist $D(x) = \infty$ für alle anderen Knoten.

2.4.2 Algorithmus von Dijkstra

In diesem Abschnitt betrachten wir den Spezialfall, in dem alle Kosten nicht-negativ sind.

Algorithmus 2.76 (Algorithmus von Dijkstra).
`Input`:
$G = (V, E)$ gerichteter, stark zusammenhängender Graph, $k : E \to \mathbb{R}_0^+$, $s \in V$.
`Output`:
$d(s, v)$ für alle $v \in V$.
`Ablauf`:
Initialisierung: D wird initialisiert wie in Definition 2.66. Sei weiterhin $F := \{s\}$.
In einer allgemeinen Iteration macht man Folgendes:

1. Man wählt $v \in V \setminus F$, wofür $D(v)$ minimal ist.
2. $F := F \cup \{v\}$.
3. Für jedes $x \in V \setminus F$ mit $vx \in E$ wird Relax(vx) durchgeführt.

Der Algorithmus terminiert wenn $F = V$, d.h. nach $n - 1$ Iterationen.

Satz 2.77. Während der gesamten Durchführung des Algorithmus gilt für alle Knoten in F: $D(v) = d(s, v)$.

Beweis. Vollständige Induktion. Nach der Initialisierung besteht F nur aus s und $D(s) = 0 = d(s, s)$.

Nehmen wir nun an, dass die Aussage nach Iteration i erfüllt war. In der Iteration $i + 1$ wird der Knoten v ausgewählt und soll in F aufgenommen werden. Da für die anderen Knoten in F der D-Wert nicht geändert wird, müssen wir nur beweisen, dass die Aussage für v gilt.

Sei W ein günstigster $s \rightsquigarrow v$ Weg, d.h. $k(W) = d(s, v)$. W verbindet einen Knoten in F mit einem Knoten in $V \setminus F$. Sei y der erste Knoten des Weges von s kommend, der nicht in F ist; sei x der Knoten davor. Sei $W_1 = W[s, x]$ und $W_2 = W[y, v]$ (siehe Abbildung 2.5). Dann ist W_1 gemäß Korollar 2.65 ein günstigster $s \rightsquigarrow x$ Weg (d.h. $k(W_1) = d(s, x)$). Da $x \in F$, gilt weiterhin laut Induktionsbedingung $D(x) = d(s, x)$.

Des Weiteren wissen wir noch, dass in dem Schritt, als x in F aufgenommen wurde, der Algorithmus dafür gesorgt hat, dass $D(y) \leq D(x)+k(xy)$ gilt. Seitdem wurde $D(x)$ nicht geändert, $D(y)$ kann nur verringert worden sein, also gilt das immer noch. All das zusammen ergibt:

$$D(y) \leq D(x) + k(xy) = d(s,x) + k(xy) = k(W_1) + k(xy) \overset{1}{\leq} k(W) = d(s,v) \overset{2}{\leq} D(v) \overset{3}{\leq} D(y).$$

(1: $W_1 + xy$ ist ein Teil von W und die Kosten der Kanten sind nicht negativ; 2: Lemma 2.68; 3: der Algorithmus hat v gewählt und nicht y, obwohl auch $y \in V \setminus F$.) Das kann nur sein, wenn überall Gleichheit steht, im Speziellen auch bei $d(s,v) \leq D(v)$. □

Korollar 2.78. Der Algorithmus von Dijkstra liefert die Kosten der günstigsten Wege von s zu allen Knoten.

Beweis. Am Ende ist $V = F$, also gilt für jeden Knoten x: $D(x) = d(s,x)$, so dass der Algorithmus korrekt ist. □

Bemerkung 2.79. Die Anzahl der Schritte hängt von der gewählten Datenstruktur ab. Wenn die D-Werte in einem Feld gespeichert sind, beträgt die Anzahl der Schritte $\mathcal{O}\left(n^2\right)$. Der aufwendigste Teil des Algorithmus ist das Finden des Knotens mit minimalem D-Wert, wofür man in jeder Iteration $\mathcal{O}\left(n\right)$ Schritte braucht. Mit raffinierteren Datenstrukturen sind etwas bessere Ergebnisse zu erzielen: $\mathcal{O}\left(m + n \log n\right)$. Man kann sogar beweisen, dass das von der Größenordnung her optimal ist.

Bemerkung 2.80. Ähnlich wie der Algorithmus von Bellman und Ford kann auch dieser Algorithmus so erweitert werden, dass er nicht nur die Kosten der günstigsten Wege, sondern die günstigsten Wege selbst liefert.

Bemerkung 2.81. Der Algorithmus von Dijkstra funktioniert auch für nicht stark zusammenhängende Graphen. In diesem Fall findet der Algorithmus die günstigsten Wege von s zu allen Knoten, die von s aus erreichbar sind. Danach ist $D(x) = \infty$ für alle $x \in V \setminus F$ und der Algorithmus kann abgebrochen werden.

Bemerkung 2.82. Der Algorithmus von Dijkstra funktioniert auch für ungerichtete Graphen.

2.4.3 Algorithmus von Floyd

Aufgabe: Berechnung der günstigsten Wege zwischen allen Knotenpaaren.

Definition 2.83 ($D^{(z)}$)**.** Sei $G = (V,E)$ ein gerichteter Graph, $k : E \to \mathbb{R}$ eine Kostenfunktion mit der Eigenschaft, dass es keinen gerichteten Kreis K mit $k(K) < 0$ gibt. Sei weiterhin $V = \{v_1, v_2, \ldots, v_n\}$. Dann bezeichne man für $1 \leq x,y \leq n$, $0 \leq z \leq n$ mit $D^{(z)}(x,y)$ die Kosten des günstigsten $x \rightsquigarrow y$ Weges, in dem alle inneren Knoten in $v_1, \ldots, v_z$ sind.

Algorithmus 2.84 (Algorithmus von Floyd).
`Input`:
$G = (V,E)$ gerichteter, stark zusammenhängender Graph, $k : E \to \mathbb{R}$ mit der Eigenschaft, dass es keinen gerichteten Kreis K mit $k(K) < 0$ gibt.
`Output`:
$d(x,y)$ für alle $x,y \in V$.
`Ablauf`:
Initialisierung:

$$D^{(0)}(x,y) := \begin{cases} 0 & \text{falls } x = y, \\ \infty & \text{falls } xy \notin E, \\ k(xy) & \text{sonst.} \end{cases}$$

Wiederhole für $z = 1, 2, \ldots, n$:
  Wiederhole für alle Paare $x,y \in V$:
    $D^{(z)}(x,y) = \min(D^{(z-1)}(x,y), D^{(z-1)}(x,z) + D^{(z-1)}(z,y))$.

Output: $D^{(n)}(x,y)$ für alle $x,y \in V$.

Satz 2.85. Der Algorithmus von Floyd berechnet korrekt die Kosten der günstigsten Wege zwischen allen Knotenpaaren.

Beweis. Man muss nur die Korrektheit der Formel zur Berechnung von $D^{(z)}$ einsehen, daraus folgt der Satz sofort. Sei W ein günstigster $x \rightsquigarrow y$ Weg, in dem alle inneren Knoten in $v_1, \ldots, v_z$ sind. Es gibt zwei Möglichkeiten, je nachdem, ob z in W enthalten ist. Wenn z in W nicht enthalten ist, dann ist $D^{(z)}(x,y) = D^{(z-1)}(x,y)$. Sonst besteht W aus zwei Teilwegen: der erste Teilweg (W_1) läuft von x zu z, der zweite Teilweg läuft von z zu y. Wegen des Optimalitätsprinzips sind W_1 und W_2 optimale Wege. Es ist auch klar, dass die inneren Knoten von W_1 und W_2 in $1, 2, \ldots, z-1$ sind. Daraus folgt, dass $k(W_1) = D^{(z-1)}(x,z)$ und $k(W_2) = D^{(z-1)}(z,y)$, was zu $k(W) = D^{(z-1)}(x,z) + D^{(z-1)}(z,y)$ führt. Welcher der beiden Fälle zur Geltung kommt, hängt davon ab, ob $D^{(z-1)}(x,y)$ oder $D^{(z-1)}(x,z) + D^{(z-1)}(z,y)$ niedriger ist. □

Bemerkung 2.86. Die Anzahl der Schritte des Algorithmus von Floyd beträgt $\mathcal{O}\left(n^3\right)$.

Bemerkung 2.87. Man kann den Algorithmus von Floyd erweitern, damit er nicht nur die Kosten der günstigsten Wege berechnet, sondern auch die günstigsten Wege selbst verfügbar macht. Dazu definiert man für jedes Paar $x, y \in V$ einen Zeiger $p(x,y)$. Wenn bei der Berechnung von $D^{(z)}$ festgestellt wird, dass $D^{(z-1)}(x,z) + D^{(z-1)}(z,y)$ niedriger ist als $D^{(z-1)}(x,y)$, wird $p(x,y)$ auf z gestellt. Somit wird erreicht, dass $p(x,y)$ am Ende auf einen inneren Knoten in einem optimalen $x \rightsquigarrow y$ Weg zeigt. Durch Verfolgung der p-Zeiger kann ein optimaler $x \rightsquigarrow y$-Weg in linearer Zeit aufgebaut werden.

2.5 Breiteste Wege

Definition 2.88 (Breite, breitester Weg)**.** Sei $G = (V,E)$ ein gerichteter oder ungerichteter Graph, $b : E \to \mathbb{R}$ eine beliebige Funktion, wobei $b(e)$ die *Breite der Kante* e angibt. Die *Breite eines Weges* W ist die minimale Breite der Kanten in W: $b(W) := \min\{b(e) : e \in W\}$. Für $x, y \in V$ ist ein *breitester Weg* ein Weg mit maximaler Breite von x zu y.

Satz 2.89. Sei $G = (V,E)$ ein ungerichteter, zusammenhängender Graph, $b : E \to \mathbb{R}$. Sei weiterhin S ein Spannbaum maximaler Kosten bzgl. der Kostenfunktion b. Für alle $x, y \in V$ ist der eindeutige $x - y$ Weg in S ein breitester Weg.

Beweis. Sei W der eindeutige $x - y$ Weg in S, und sei e eine Kante minimaler Breite in W, d.h. $b(W) = b(e)$. Nehmen wir indirekt an, es gibt einen breiteren Weg W', d.h. $b(W) < b(W')$. Dann gilt für alle Kanten $e' \in W'$, dass $b(e') > b(e)$. Wenn man e aus S entfernt, zerfällt der Baum in zwei Komponenten; die eine Komponente enthält x, die andere enthält y. Da W' ein $x - y$-Weg ist, muss er die Grenze zwischen den beiden Komponenten überqueren, d.h. es gibt eine Kante $e' \in W'$, die zwischen den beiden Komponenten läuft. Dann ist aber $S' = S \setminus \{e\} \cup \{e'\}$ auch ein Spannbaum, und wegen $b(e') > b(e)$ hat S' höhere Kosten als S, was ein Widerspruch ist. □

Bemerkung 2.90. Der Spannbaum maximaler Kosten kann mit einer kleinen Änderung des Algorithmus von Kruskal bestimmt werden.

Sei nun $G = (V,E)$ ein gerichteter Graph. Dann können die breitesten Wege von $s \in V$ zu allen anderen Knoten mit der folgenden Anpassung des Algorithmus von Dijkstra bestimmt werden.

Algorithmus 2.91 (Modifizierter Dijkstra-Algorithmus).
`Input`:
$G = (V,E)$ gerichteter, stark zusammenhängender Graph, $b : E \to \mathbb{R}_0^+$, $s \in V$.
`Output`:
Breite des breitesten $s \rightsquigarrow v$ Weges für alle $v \in V$.

`Ablauf`:
Initialisierung:

$$B(v) := \begin{cases} \infty & \text{falls } s = v, \\ 0 & \text{falls } sv \notin E, \\ b(sv) & \text{sonst.} \end{cases}$$

Sei weiterhin $F := \{s\}$.
In einer allgemeinen Iteration macht man Folgendes:

1. Man wählt $v \in V \setminus F$, wofür $B(v)$ maximal ist.
2. $F := F \cup \{v\}$.
3. Für jedes $x \in V \setminus F$ mit $vx \in E$ wird ein modifizierter Relax-Schritt durchgeführt:
   Wenn $\min(B(v), b(vx)) > B(x)$, dann sei $B(x) := \min(B(v), b(vx))$.

Der Algorithmus terminiert, wenn $F = V$, d.h. nach $n - 1$ Iterationen.

Satz 2.92. Der modifizierte Dijkstra-Algorithmus berechnet korrekt die breitesten Wege.

Beweis. Analog zum Beweis der Korrektheit des normalen Dijkstra-Algorithmus. □

Bemerkung 2.93. Durch entsprechende Anpassung des Dijkstra-Algorithmus kann man auch die Aufgabe lösen, in der die Kanten sowohl mit Kosten als auch mit Breite versehen sind, und man sucht den breitesten der günstigsten $s \rightsquigarrow v$ Wege, oder den günstigsten der breitesten $s \rightsquigarrow v$ Wege. In beiden Fällen muss man im Dijkstra-Algorithmus sowohl D als auch B speichern, und sowohl den Relax-Schritt als auch die Auswahl des nächstgewählten Knotens v anpassen.

2.6 Gerichtete azyklische Graphen

Definition 2.94 (DAG)**.** Ein gerichteter Graph ist ein *DAG (directed acyclic graph)*, wenn er keinen gerichteten Kreis enthält.

Definition 2.95 (topologische Ordnung)**.** Sei G ein gerichteter Graph. Eine Reihenfolge $v_1, v_2, \ldots, v_n$ der Knoten ist eine *topologische Ordnung*, wenn für alle Kanten $v_i v_j$ gilt, dass $i < j$.

Satz 2.96. Ein gerichteter Graph hat genau dann eine topologische Ordnung, wenn er ein DAG ist. Mit einer einfachen Ergänzung der Tiefensuche kann in $\mathcal{O}(n + m)$ Schritten entweder eine topologische Ordnung oder ein gerichteter Kreis gefunden werden.

Beweis. Es ist klar, dass es nicht möglich ist, dass ein Graph sowohl eine topologische Ordnung als auch einen gerichteten Kreis hat. Es muss nur noch gezeigt werden, dass man mit der Tiefensuche immer entweder eine topologische Ordnung oder einen gerichteten Kreis findet.

Bezeichnen wir den ersten Knoten, für den der Aufruf der Tiefensuche terminiert, mit v_n, den zweiten Knoten, für den der Aufruf terminiert, mit v_{n-1} usw. Der Knoten, für den der Aufruf zuletzt terminiert, ist v_1. Betrachten wir nun zwei Knoten v_i und v_j mit $i < j$. Das heißt: der Aufruf der Tiefensuche terminierte zuerst für v_j, erst später für v_i. Je nach den Anfangszeitpunkten der Aufrufe für v_i und v_j sind drei zeitliche Anordnungen denkbar, die in Abbildung 2.6 dargestellt sind.

- Im Fall a) hat der Aufruf für v_i indirekt auch zum Aufruf für v_j geführt. Daraus folgt, dass v_j über einen gerichteten Weg von v_i aus erreichbar ist. Wenn $v_j v_i \in E$, dann gibt es also einen gerichteten Kreis.

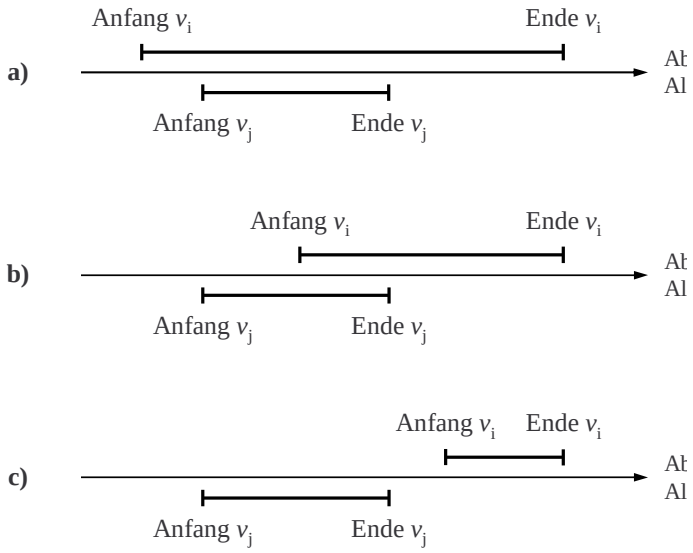

Abbildung 2.6: Zeitliche Anordnungen beim Ablauf der Tiefensuche

- Der Fall b) ist nicht möglich. Dieser würde nämlich bedeuten, dass während des Aufrufs für v_j auch ein Aufruf für v_i gestartet wird. Dann würde aber der Aufruf für v_j nicht terminieren solange der Aufruf für v_i noch läuft.
- Im Fall c) wird v_i erst nach Abschluss des Aufrufes für v_j besucht. Daraus folgt, dass $v_jv_i \notin E$.

Wenn der Graph ein DAG ist, gilt sowohl im Fall a) als auch im Fall c) $v_jv_i \notin E$, so dass die definierte Reihenfolge eine topologische Ordnung ist. Umgekehrt, wenn die definierte Reihenfolge keine topologische Ordnung ist, dann muss mindestens einmal im Fall a) $v_jv_i \in E$ gelten, so dass es einen gerichteten Kreis gibt. Diesen Kreis findet man bei der Untersuchung der aus v_j ausgehenden Kanten: Der Endknoten der Kante ist ein Knoten, für den der Aufruf der Tiefensuche schon begonnen, aber noch nicht beendet wurde. Der Kreis besteht also aus dieser Kante und aus dem Weg entlang dessen der Algorithmus von v_i zu v_j gekommen ist. □

Algorithmus 2.97 (Bestimmung der günstigsten/ungünstigsten Wege in einem DAG).
`Input`:
$G = (V, E)$ DAG, $k : E \to \mathbb{R}$, $s \in V$.
`Output`:
Kosten der günstigsten bzw. ungünstigsten Wege von s zu allen anderen, von s aus erreichbaren Knoten.
`Ablauf`:
1. Bestimmung einer topologischen Ordnung $v_1, v_2, \ldots, v_n$
2. Sei $s = v_i$. Für alle $j < i$ ist v_j aus s nicht erreichbar.
3. Sei $a(s) = b(s) = 0$
4. for($j = i + 1$; $j \leq n$; j++)
$a(v_j) = \min\{a(v_l) + k(v_lv_j) \mid v_lv_j \in E \text{ und } l \geq i\}$;
$b(v_j) = \max\{b(v_l) + k(v_lv_j) \mid v_lv_j \in E \text{ und } l \geq i\}$;
5. Am Ende enthält $a(v)$ die Kosten des günstigsten $s \rightsquigarrow v$ Weges, $b(v)$ die Kosten des ungünstigsten $s \rightsquigarrow v$ Weges.

Satz 2.98. (1) Der Algorithmus liefert in a die Kosten der günstigsten Wege von s zu allen anderen, von s aus erreichbaren Knoten.
(2) Der Algorithmus liefert in b die Kosten der ungünstigsten Wege von s zu allen anderen, von s aus

erreichbaren Knoten.
(3) Die Anzahl der Schritte beträgt $\mathcal{O}(n+m)$.

Beweis. (1) Da der Graph ein DAG ist, enthält er auch keinen gerichteten Kreis mit negativen Kosten, also kann das Optimalitätsprinzip (Korollar 2.65) benutzt werden.

Die Aussage wird mit vollständiger Induktion bzgl. j bewiesen. Für $j = i$ ist die Aussage trivial, da $v_i = s$ und $a(s) = 0$ gesetzt wird. Sei nun $j > i$ und sei W ein günstigster $s \rightsquigarrow v_j$ Weg. Der vorletzte Knoten von W sei v_t. Natürlich gilt dann $t \geq i$. Da $v_t v_j \in E$, muss auch $t < j$ gelten, also enthält $a(v_t)$ wegen der Induktionsbedingung eben die Kosten des günstigsten $s \rightsquigarrow v_t$ Weges. Andererseits ist $W[s, v_t]$ wegen des Optimalitätsprinzips ein günstigster $s \rightsquigarrow v_t$ Weg. Folglich ist $k(W[s, v_t]) = a(v_t)$ und damit $k(W) = k(W[s, v_t]) + k(v_t v_j) = a(v_t) + k(v_t v_j)$. Der Algorithmus stellt $a(v_j)$ auf das Minimum solcher Ausdrücke, d.h. $a(v_j) \leq k(W) = d(s, v_j)$ wird gelten.

Umgekehrt, sei v_q jener Knoten, für den $v_q v_j \in E$, $q \geq i$ und $a(v_q) + k(v_q v_j)$ minimal ist. D.h. der Algorithmus setzt $a(v_j) = a(v_q) + k(v_q v_j)$. Sei W' ein günstigster $s \rightsquigarrow v_q$ Weg. Da $v_q v_j \in E$, gilt $q < j$, und daher ist wegen der Induktionsbedingung $a(v_q) = k(W')$. Da $W' + v_q v_j$ ein $s \rightsquigarrow v_j$ Kantenzug ist, folgt wegen Lemma 2.64: $a(v_j) = a(v_q) + k(v_q v_j) = k(W') + k(v_q v_j) = k(W' + v_q v_j) \geq d(s, v_j)$.

(2) Bezeichne $B(v)$ die Kosten des ungünstigsten $s \rightsquigarrow v$ Weges. Man kann $B(v)$ erhalten, in dem man die Kosten des günstigsten $s \rightsquigarrow v$ Weges bezüglich der Kostenfunktion $k' = -k$ ermittelt und mit -1 multipliziert. Es wurde im vorherigen Schritt bewiesen, dass durch die Formel $A(v_j) = \min\{A(v_l) + k'(v_l v_j) \mid v_l v_j \in E \text{ und } l \geq i\}$ die Kosten der günstigsten $s \rightsquigarrow v$ Wege bezüglich der Kostenfunktion k' berechnet werden können, also ist $B(v) = -A(v)$. Folglich ist $-B(v_j) = \min\{-B(v_l) - k(v_l v_j) \mid v_l v_j \in E \text{ und } l \geq i\}$, woraus durch Multiplikation mit -1 die Formel $B(v_j) = \max\{B(v_l) + k(v_l v_j) \mid v_l v_j \in E \text{ und } l \geq i\}$ folgt, dieselbe Formel wie für $b(v_j)$.

(3) Die topologische Ordnung kann in $\mathcal{O}(n+m)$ Schritten ermittelt werden. Im späteren Verlauf des Algorithmus werden für jeden Knoten v eben $\mathcal{O}(1 + d(v))$ Schritte gemacht. □

Definition 2.99 (CPM-Netzplan)**.** Ein *CPM-Netzplan* ist ein DAG, in dem die Kanten Vorgänge (Teilaufgaben eines größeren Vorhabens), die Knoten Ereignisse (Meilensteine, Teilergebnisse) darstellen. Für jede Kante ist die Dauer des Vorgangs als eine reelle Zahl gegeben. Es gibt zwei ausgezeichnete Knoten: den *Start* (S) und den *Abschluss* (A), die in jeder topologischen Ordnung des Graphen an der ersten bzw. letzten Stelle sind. Jeder Knoten hat einen *frühstmöglichen (ASAP)* und einen *letztmöglichen (ALAP) Zeitpunkt* (diese sind nicht gegeben, sondern müssen berechnet werden). Die Differenz zwischen diesen Zeitpunkten ist die *Pufferzeit* des Knotens. Die Pufferzeit einer Kante xy ist $\text{ALAP}(y) - \text{ASAP}(x) - \text{Dauer}(xy)$. Wenn die Pufferzeit eines Knotens / einer Kante 0 ist, ist es ein *kritischer Knoten* bzw. eine *kritische Kante*. Die kritischen Knoten und Kanten bilden den *kritischen Teilgraphen*, der oft ein Weg ist (der *kritische Weg*).

Algorithmus 2.100 (Critical Path Method, CPM).
`Input`:
CPM-Netzplan, erlaubte Dauer.
`Output`:
ASAP- und ALAP-Zeitpunkte der Knoten.
`Ablauf`:
1. Bestimmung einer topologischen Ordnung $v_1, v_2, \ldots, v_n$.
2. Bestimmung der ungünstigsten Wege von S zu allen anderen Knoten. Die Kosten dieser Wege definieren die ASAP-Zeitpunkte. Speziell der ASAP-Zeitpunkt für A ist die kleinstmögliche Dauer des gesamten Vorhabens.
3. Wenn die kleinstmögliche Dauer höher ist als die erlaubte Dauer, ist es nicht möglich, die erlaubte Dauer einzuhalten; der Algorithmus wird abgebrochen. Ansonsten sei $\text{ALAP}(A) =$ erlaubte Dauer.
4. Die Kanten und die topologische Ordnung werden in umgekehrter Reihenfolge betrachtet und die ungünstigsten Wege von A zu allen anderen Knoten werden berechnet. Diese definieren die ALAP-Zeitpunkte.

Satz 2.101. Dieser Algorithmus liefert in $\mathcal{O}(n+m)$ Schritten die ASAP- und ALAP-Zeitpunkte aller Knoten (und damit auch die Pufferzeiten aller Knoten und Kanten bzw. die kritischen Knoten und Kanten). ⊠

2.7 Paarungen

Definition 2.102 (Paarung)**.** Sei $G = (V, E)$ ein Graph. $M \subseteq E$ ist eine *Paarung* (auch Matching oder unabhängige Kantenmenge genannt), wenn es kein Paar von Kanten in M gibt, die zu demselben Knoten inzident sind. Die Knoten, die zu einer Kante in M inzident sind, sind *gepaart*, die anderen *ungepaart*.

Definition 2.103 (maximale Paarung, Paarung maximaler Mächtigkeit, vollständige Paarung)**.** Sei M eine Paarung in $G = (V, E)$.
M ist eine *maximale Paarung*, wenn $\forall e \in E \setminus M$ die Menge $M \cup \{e\}$ keine Paarung ist.
M ist eine *Paarung maximaler Mächtigkeit*, wenn $|M|$ unter den Paarungen von G maximal ist.
M ist eine *vollständige Paarung*, wenn alle Knoten gepaart sind.

Definition 2.104 (alternierender Weg, Verbesserungsweg)**.** Sei M eine Paarung im Graphen G. Der Weg W mit der Knotenfolge $v_0, v_1, v_2, \ldots, v_k$ ist ein *alternierender Weg* hinsichtlich M, wenn die folgenden Bedingungen erfüllt sind:

- $v_i v_{i+1} \in M$ für $i = 1, 3, 5, \ldots$
- $v_i v_{i+1} \notin M$ für $i = 0, 2, 4, \ldots$

Wenn zudem k ungerade ist und v_0 und v_k ungepaart sind, dann ist W ein *Verbesserungsweg*.

Lemma 2.105. Wenn jeder Knoten in einem Graphen höchstens zwei Nachbarn hat, dann ist jede Komponente des Graphen ein isolierter Knoten, ein Weg oder ein Kreis.

Beweis. Betrachten wir eine Komponente des Graphen, die mindestens zwei Knoten enthält. Sei W ein längster Weg in dieser Komponente; seien x und y die beiden Endknoten von W. Kein Knoten von W kann mit einem Knoten außerhalb W verbunden sein: x und y nicht weil W dann kein längster Weg wäre und ein innerer Knoten von W auch nicht, weil er dann mindestens drei Nachbarn hätte. Daher besteht diese Komponente ausschließlich aus den Knoten von W. Die inneren Knoten können auch zu keinen weiteren Kanten inzident sein. D.h. die Komponente ist entweder eben nur der Weg W oder der Kreis $W + xy$. □

Satz 2.106 (Berge)**.** Sei M eine Paarung im Graphen G. M ist dann und nur dann eine Paarung maximaler Mächtigkeit, wenn es keinen Verbesserungsweg gibt.

Beweis. Nehmen wir zuerst an, dass M eine Paarung maximaler Mächtigkeit ist. Nehmen wir indirekt an, es gibt trotzdem einen Verbesserungsweg W. Bezeichne $E(W)$ die Kanten in W und betrachten wir die symmetrische Differenz $M' := M \triangle E(W) = (M \cup E(W)) \setminus (M \cap E(W))$. M' wäre dann eine Paarung größerer Mächtigkeit als M, denn jeder innere Knoten von W ist in M' wie in M gepaart, und zusätzlich sind in M' die beiden Endknoten von W auch gepaart.

Nun nehmen wir an, dass es keinen Verbesserungsweg gibt. Nehmen wir indirekt an, es gibt eine Paarung M' mit $|M'| > |M|$. Betrachten wir den Graphen $G' = (V, M \triangle M')$. Da zu einem Knoten höchstens eine Kante in M und höchstens eine Kante in M' inzident ist, hat jeder Knoten in G' höchstens zwei Nachbarn. Laut Lemma 2.105 ist jede Komponente von G' ein isolierter Knoten, ein Weg oder ein Kreis. Schauen wir uns diese Komponenten an. Bei Wegen und Kreisen gilt: Da M und M' Paarungen sind, können nicht mehrere Kanten aus M oder mehrere Kanten aus M' nebeneinander sein. D.h. Kanten aus M und M' alternieren. Folglich sind die Anzahl von M-Kanten und M'-Kanten in einem Kreis gleich. Da $|M'| > |M|$, muss es also einen Weg geben, in dem es mehr M'-Kanten gibt als M-Kanten. Das kann nur sein, wenn die erste und letzte Kante des Weges M'-Kanten sind. Dann ist das aber ein Verbesserungsweg hinsichtlich M. □

Definition 2.107. Sei M eine Paarung und v ein gepaarter Knoten. $M(v)$ bezeichnet den Knoten, mit dem v gepaart ist. Sei $X \subseteq V$ eine Menge gepaarter Knoten, dann ist $M(X) := \{M(v) \mid v \in X\}$.

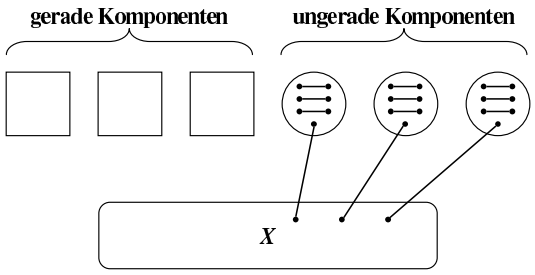

Abbildung 2.7: Zum Satz von Tutte

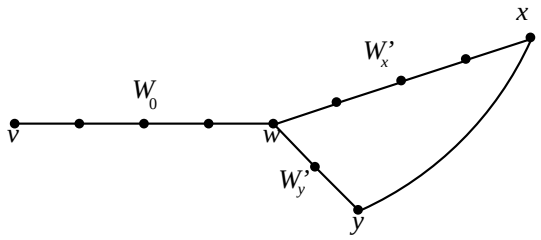

Abbildung 2.8: Zum Beweis von Satz 2.111

Satz 2.108 (Tutte)**.** Im Graphen $G = (V, E)$ gibt es dann und nur dann eine vollständige Paarung, wenn für alle $X \subseteq V$ Folgendes gilt: Wenn die Knoten von X aus G entfernt werden, hat der resultierende Graph höchstens $|X|$ Komponenten mit einer ungeraden Anzahl von Knoten.

Beweis der einfachen Richtung. Nehmen wir an, dass es eine vollständige Paarung M gibt. Sei $X \subseteq V$ und betrachten wir die Komponenten von $G \setminus X$ mit einer ungeraden Anzahl von Knoten (siehe Abbildung 2.7). In einer solchen Komponente können nicht alle Knoten innerhalb der Komponente gepaart sein, d.h. es gibt in jeder solchen Komponente mindestens einen Knoten v, wofür $M(v)$ außerhalb der Komponente ist. Dann kann aber $M(v)$ nur in X liegen. Das heißt: zu jeder solchen Komponente gibt es einen Knoten in X, und diese Knoten müssen unterschiedlich sein. Daher kann die Anzahl der Knoten in X nicht kleiner sein als die Anzahl solcher Komponenten. □

Bemerkung 2.109. Eine Paarung maximaler Mächtigkeit in einem allgemeinen Graphen kann in Polynomialzeit gefunden werden. Der erste polynomiale Algorithmus stammt von Jack Edmonds aus dem Jahr 1965. Der schnellste Algorithmus braucht $\mathcal{O}(\sqrt{n}m)$ Schritte.

2.7.1 Paarungen in bipartiten Graphen

Definition 2.110 (bipartiter Graph)**.** $G = (V, E)$ ist ein *bipartiter Graph*, wenn $V = A \cup B$, $A \cap B = \emptyset$ und es keine Kante gibt, die zwei Knoten in A oder zwei Knoten in B miteinander verbindet. G wird dann auch so bezeichnet: $G = (A, B, E)$. Der *vollständige bipartite Graph* $K_{a,b}$ ist ein bipartiter Graph mit $|A| = a$ und $|B| = b$, in dem alle Knoten in A mit allen Knoten in B benachbart sind.

Satz 2.111. Ein Graph ist genau dann bipartit, wenn er keinen Kreis ungerader Länge enthält.

Beweis. Sei K ein Kreis in einem bipartiten Graphen. Dann sind die Knoten von K abwechselnd in A bzw. B. Daher besteht der Kreis aus einer geraden Anzahl von Knoten.

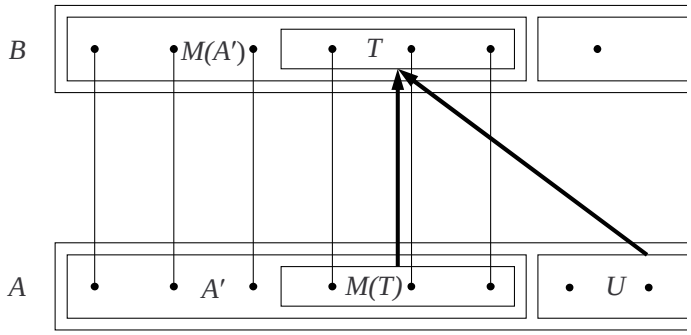

Abbildung 2.9: Struktur einer Paarung maximaler Mächtigkeit in einem bipartiten Graphen. Die dicken Pfeile verdeutlichen, dass Knoten in U und $M(T)$ nur mit Knoten in T benachbart sein können.

Umgekehrt sei G ein Graph, in dem es keinen Kreis ungerader Länge gibt. Dann kann man die Mengen A und B wie folgt konstruieren. Sei S ein Spannwald des Graphen. In jedem Baum von S wähle man willkürlich eine Wurzel; diese werden in A sein. Jeder weitere Knoten ist in S über genau einen Weg aus der Wurzel seiner Komponente zu erreichen. Ist die Länge dieses Weges gerade, so wird der Knoten in A sein, sonst in B. Es muss nun bewiesen werden, dass es keine Kante innerhalb von A oder B gibt. Nehmen wir indirekt an, xy ist eine solche Kante. Sei v die Wurzel in der Komponente von x und y und seien W_x und W_y die Wege von v zu x bzw. y in S (siehe Abbildung 2.8). Sei w der letzte gemeinsame Knoten von W_x und W_y. Sei $W'_x := W_x[w, x]$, $W'_y := W_y[w, y]$ und $W_0 := W_x[v, w] = W_y[v, w]$. Dann ergeben W'_x, W'_y und xy zusammen einen Kreis der Länge $l(W'_x)+l(W'_y)+1 = l(W_x)-l(W_0)+l(W_y)-l(W_0)+1 = l(W_x) + l(W_y) - 2 \cdot l(W_0) + 1$, was ungerade ist, da $l(W_x)$ und $l(W_y)$ die gleiche Parität haben. □

Algorithmus 2.112 (Algorithmus von Kőnig).
`Input`:
Bipartiter Graph $G = (A, B, E)$.
`Output`:
Paarung maximaler Mächtigkeit in G.
`Ablauf`:
Initialisierung: $M := \emptyset$

In einer allgemeinen Iteration sucht man einen Verbesserungsweg mit einer modifizierten Breitensuche: $S_1 = \{$ungepaarte Knoten in $A\}$, $S_{2i} = \{$noch nicht besuchte Knoten, die aus S_{2i-1} über eine in M nicht enthaltene Kante erreichbar sind$\}$, $S_{2i+1} = \{$noch nicht besuchte Knoten, die aus S_{2i} über eine M-Kante erreichbar sind$\}$ $(i = 1, 2, 3, \ldots)$. Findet man in S_{2i} einen ungepaarten Knoten, so hat man einen Verbesserungsweg W gefunden; dann sei $M^{neu} := M \triangle W$ und die nächste Iteration kann beginnen. Hat man in einer Iteration bis zum Ende der Breitensuche keinen Verbesserungsweg gefunden, terminiert der Algorithmus mit M als Output.

Satz 2.113. Für bipartite Graphen liefert der Algorithmus von Kőnig eine Paarung maximaler Mächtigkeit. Die Anzahl der Schritte beträgt $\mathcal{O}(nm)$. ⊠

Bemerkung 2.114. Der beste Algorithmus für dieses Problem stammt von John Hopcroft und Richard Karp aus dem Jahr 1973, und braucht $\mathcal{O}(\sqrt{n}m)$ Schritte.

Definition 2.115. Sei $G = (V, E)$ ein Graph, $X \subseteq V$. $N(X)$ bezeichnet die Menge der Knoten, die mit mindestens einem Knoten in X benachbart sind.

Lemma 2.116. Sei $G = (A, B, E)$ ein bipartiter Graph. Sei M eine Paarung maximaler Mächtigkeit. Bezeichne A' die gepaarten Knoten in A, $U := A \setminus A'$. Sei T die Menge der Knoten in B, die aus U über einen alternierenden Weg erreichbar sind. Dann gelten folgende Eigenschaften (siehe Abbildung 2.9):

(1) $T \subseteq M(A')$, $M(T) \subseteq A'$, $U \cap M(T) = \emptyset$.
(2) Die Knoten in U können nur mit Knoten in T benachbart sein, d.h. $N(U) \subseteq T$.
(3) Die Knoten in $M(T)$ können nur mit Knoten in T benachbart sein, d.h. $N(M(T)) \subseteq T$. Es gilt sogar $N(M(T)) = T$.

Beweis.
(1) Wenn ein Knoten in $B \setminus M(A')$ über einen alternierenden Weg von U aus erreichbar wäre, dann wäre das ein Verbesserungsweg, so dass M nicht maximale Mächtigkeit haben könnte. Daher gilt $T \subseteq M(A')$. Daraus folgt $M(T) \subseteq A'$, was wiederum $U \cap M(T) = \emptyset$ impliziert.
(2) Wenn $u \in U$, dann gilt $N(u) \subseteq T$, weil eine Kante uv ein alternierender Weg der Länge 1 ist.
(3) Sei $x \in M(T)$ und xy eine beliebige zu x inzidente Kante. Nehmen wir indirekt an, dass $y \notin T$. Da $M(x) \in T$, gibt es einen alternierenden Weg W von einem Knoten $u \in U$ zu $M(x)$. x kann in W nicht enthalten sein, weil dann die Kante in W, die zu x führt, $M(x)x$ sein müsste, so dass $M(x)$ zweimal in W erscheinen würde. y kann in W auch nicht enthalten sein, weil dann der Teil von W bis y ein alternierender Weg zu y wäre und somit $y \in T$ gelten würde. Daher kann man W mit den Kanten $M(x)x$ und xy ergänzen und das Ergebnis ist ein alternierender Weg von u zu y, also $y \in T$, was ein Widerspruch ist. Damit haben wir also $N(M(T)) \subseteq T$ bewiesen. $N(M(T)) \supseteq T$ ist trivial. □

Satz 2.117 (Hall)**.** In einem bipartiten Graphen $G = (A, B, E)$ gibt es dann und nur dann eine Paarung, bei der alle Knoten von A gepaart sind, wenn für alle $X \subseteq A$ gilt, dass $|N(X)| \geq |X|$ (Hall-Bedingung).

Beweis. Wenn es eine Paarung M gibt, bei der alle Knoten von A gepaart sind, dann gilt für alle $X \subseteq A$: $|N(X)| \geq |M(X)| = |X|$.

Nehmen wir nun an, es gibt keine solche Paarung. Sei M eine Paarung maximaler Mächtigkeit. Da durch M nicht alle Knoten von A gepaart sind, heißt das in der Terminologie von Lemma 2.116, dass $U \neq \emptyset$. Daraus und aus Lemma 2.116 folgt: $N(M(T) \cup U) = T$ und $|M(T) \cup U| = |M(T)| + |U| = |T| + |U| > |T|$, was der Hall-Bedingung widerspricht. □

Satz 2.118 (Frobenius)**.** Im einem bipartiten Graphen $G = (A, B, E)$ gibt es dann und nur dann eine vollständige Paarung, wenn $|A| = |B|$ und für alle $X \subseteq A$ gilt, dass $|N(X)| \geq |X|$.

Beweis. Die Notwendigkeit beider Bedingungen ist trivial. Umgekehrt, wenn die Bedingungen erfüllt sind, folgt aus Satz 2.117 die Existenz einer Paarung M, bei der alle Knoten in A gepaart sind. Da $|A| = |B|$, sind auch alle Knoten von B durch M gepaart. □

2.8 Unabhängige und überdeckende Knoten- und Kantenmengen

Definition 2.119. Sei $G = (V, E)$ ein Graph. $V' \subseteq V$ ist eine *unabhängige Knotenmenge*, wenn keine Kante in V' läuft, d.h. wenn $x, y \in V'$, dann gilt $xy \notin E$. Die größte Mächtigkeit einer unabhängigen Knotenmenge in G wird mit $\alpha(G)$ bezeichnet.

Definition 2.120. Sei $G = (V, E)$ ein Graph. $V' \subseteq V$ ist eine *überdeckende Knotenmenge*, wenn jede Kante von G zu mindestens einem Knoten in V' inzident ist. Die kleinste Mächtigkeit einer überdeckenden Knotenmenge in G wird mit $\tau(G)$ bezeichnet.

Definition 2.121. Sei $G = (V, E)$ ein Graph. $E' \subseteq E$ ist eine *unabhängige Kantenmenge*, wenn es keine zwei Kanten in E' mit einem gemeinsamen Endknoten gibt. Die größte Mächtigkeit einer unabhängigen Kantenmenge in G wird mit $\nu(G)$ bezeichnet.

Definition 2.122. Sei $G = (V, E)$ ein Graph ohne isolierte Knoten. $E' \subseteq E$ ist eine *überdeckende Kantenmenge*, wenn jeder Knoten von G zu mindestens einer Kante in E' inzident ist. Die kleinste Mächtigkeit einer überdeckenden Kantenmenge in G wird mit $\varrho(G)$ bezeichnet.

Satz 2.123. Für jeden Graphen G gilt: $\nu(G) \leq \tau(G)$.

Beweis. Sei M eine unabhängige Kantenmenge mit $|M| = \nu(G)$. Um die Kanten in M zu überdecken, braucht man schon $|M|$ Knoten, daher gilt $\tau(G) \geq |M|$. □

Satz 2.124. Für jeden Graphen G ohne isolierte Knoten gilt: $\alpha(G) \leq \varrho(G)$.

Beweis. Sei X eine unabhängige Knotenmenge mit $|X| = \alpha(G)$. Um die Knoten in X zu überdecken, braucht man schon $|X|$ Kanten, daher gilt $\varrho(G) \geq |X|$. □

Lemma 2.125. Sei $G = (V, E)$ ein Graph, $X \subseteq V$. Dann ist X genau dann eine unabhängige Knotenmenge, wenn $V \setminus X$ eine überdeckende Knotenmenge ist.

Beweis. Wenn X unabhängig ist, dann muss mindestens ein Endknoten jeder Kante in $V \setminus X$ liegen, d.h. $V \setminus X$ ist überdeckend. Umgekehrt, wenn $V \setminus X$ überdeckend ist, dann muss mindestens ein Endknoten jeder Kante in $V \setminus X$ liegen, d.h. X ist unabhängig. □

Bemerkung 2.126. Sei $xx \in E$ eine Schlinge. Dann ist x in jeder überdeckenden Knotenmenge aber in keiner unabhängigen Knotenmenge enthalten.

Satz 2.127 (Gallai)**.** Für jeden Graphen G gilt: $\alpha(G) + \tau(G) = n$.

Beweis. Sei X eine unabhängige Knotenmenge mit $|X| = \alpha(G)$. Laut Lemma 2.125 ist $V \setminus X$ eine überdeckende Knotenmenge mit $n - \alpha(G)$ Knoten. Daraus folgt $\tau(G) \leq n - \alpha(G)$, also $\alpha(G) + \tau(G) \leq n$.

Sei Y eine überdeckende Knotenmenge mit $|Y| = \tau(G)$. Laut Lemma 2.125 ist $V \setminus Y$ eine unabhängige Knotenmenge mit $n - \tau(G)$ Knoten. Daraus folgt $\alpha(G) \geq n - \tau(G)$, also $\alpha(G) + \tau(G) \geq n$. □

Lemma 2.128. Sei $G = (V, E)$ ein Graph ohne isolierte Knoten, $E' \subseteq E$ eine überdeckende Kantenmenge minimaler Mächtigkeit. Dann ist $G' = (V, E')$ ein Wald, in dem jede Komponente aus mindestens zwei Knoten besteht.

Beweis. Wenn G' einen Kreis enthalten würde, könnte man eine beliebige Kante des Kreises aus E' entfernen, und man würde eine kleinere überdeckende Kantenmenge erhalten. Daher ist G' ein Wald. Da E' alle Knoten von G überdeckt, gibt es keine isolierten Knoten in G'. □

Satz 2.129 (Gallai)**.** Für jeden Graphen G ohne isolierte Knoten gilt: $\nu(G) + \varrho(G) = n$.

Beweis. Sei M eine unabhängige Kantenmenge mit $\nu(G)$ Kanten. Man kann M zu einer überdeckenden Kantenmenge D ergänzen, indem man für jeden der $n - 2\nu(G)$ Knoten, die durch M nicht überdeckt sind, eine beliebige inzidente Kante nimmt. Dann ist $|D| \leq \nu(G) + (n - 2\nu(G)) = n - \nu(G)$. Daraus folgt $\varrho(G) \leq |D| \leq n - \nu(G)$, also $\nu(G) + \varrho(G) \leq n$.

Sei nun $E' \subseteq E$ eine überdeckende Kantenmenge mit $|E'| = \varrho(G)$. Laut Lemma 2.128 ist E' ein Wald; die Anzahl der Komponenten sei k. Damit gilt laut Satz 2.30: $\varrho(G) = n - k$, also $k = n - \varrho(G)$. Andererseits, wenn man aus jeder Komponente eine beliebige Kante auswählt, erhält man offensichtlich eine unabhängige Kantenmenge mit k Kanten. Folglich gilt $\nu(G) \geq k$. Zusammen ergibt das: $\nu(G) \geq k = n - \varrho(G)$, also $\nu(G) + \varrho(G) \geq n$. □

Satz 2.130 (Kőnig)**.** Sei G ein bipartiter Graph. Dann gilt $\nu(G) = \tau(G)$. Wenn G keinen isolierten Knoten enthält, gilt zudem auch $\alpha(G) = \varrho(G)$.

Beweis. Sei M eine Paarung mit $|M| = \nu(G)$. Mit der Terminologie von Lemma 2.116 (siehe Abbildung 2.9), sei $X := T \cup (A' \setminus M(T))$. Dann gilt $|X| = |M| = \nu(G)$ und aus Lemma 2.116 folgt, dass X alle Kanten von G überdeckt. Daher gilt $\tau(G) \leq \nu(G)$. Zusammen mit Satz 2.123 haben wir damit $\nu(G) = \tau(G)$ bewiesen. Aus Satz 2.127 und Satz 2.129 folgt zudem, dass $\alpha(G) + \tau(G) = \nu(G) + \varrho(G)$. Daraus ergibt sich $\alpha(G) = \nu(G) - \tau(G) + \varrho(G) = \varrho(G)$. □

2.9 Flüsse

Definition 2.131 (Netzwerk)**.** Sei $G = (V, E)$ ein gerichteter Graph, $s \neq t$ zwei ausgezeichnete Knoten. Sei $c : E \to \mathbb{R}^+$, $c(e)$ wird die *Kapazität* von e genannt. Dann ist das Tupel $N = (G, s, t, c)$ ein *Netzwerk*.

Definition 2.132 (Fluss)**.** Sei $N = (G, s, t, c)$ ein Netzwerk. $f : E \to \mathbb{R}_0^+$ ist ein *Fluss*, wenn für jede Kante e die Ungleichung $f(e) \leq c(e)$ und für jeden Knoten v außer s und t die Gleichung $m(v) := \sum_{xv \in E} f(xv) - \sum_{vx \in E} f(vx) = 0$ gilt. $m_f := m(t)$ ist die *Stärke* des Flusses.

Definition 2.133 ($s\bar{t}$-Schnitt)**.** Sei $X \subseteq V$ so dass $s \in X$ und $t \in V \setminus X$. Dann ist

$$E(X, \bar{X}) := \{xy \in E : x \in X, y \in V \setminus X \text{ oder } y \in X, x \in V \setminus X\}$$

ein *$s\bar{t}$-Schnitt*. Die *Kapazität* von $E(X, \bar{X})$ ist

$$c(X, \bar{X}) := \sum_{xy \in E, x \in X, y \in V \setminus X} c(xy).$$

Der *Fluss* durch $E(X, \bar{X})$ ist

$$f(X, \bar{X}) := \sum_{xy \in E, x \in X, y \in V \setminus X} f(xy) - \sum_{xy \in E, x \in V \setminus X, y \in X} f(xy).$$

Lemma 2.134. Seien f ein Fluss und $E(X, \bar{X})$ ein $s\bar{t}$-Schnitt im Netzwerk N. Dann gilt:
(1) $f(X, \bar{X}) = m_f$.
(2) $f(X, \bar{X}) \leq c(X, \bar{X})$.
(3) Wenn für alle Kanten, die von X nach $V \setminus X$ laufen, $f = c$, und für alle Kanten, die von $V \setminus X$ nach X laufen, $f = 0$, dann hat f maximale Stärke.

Beweis.
(1) Betrachten wir $M := \sum_{v \in V \setminus X} m(v) = \sum_{v \in V \setminus X} \left(\sum_{xv \in E} f(xv) - \sum_{vx \in E} f(vx) \right)$. Einerseits ist $M = m(t) = m_f$, weil für alle Knoten v in $V \setminus X$, außer t, $m(v) = 0$. Andererseits wird für jede Kante xy, für die sowohl x als auch y in $V \setminus X$ ist, $f(xy)$ zweimal betrachtet, jeweils mit unterschiedlichem Vorzeichen. Damit ist der Beitrag dieser Kanten zu M gleich 0. Nur jene Kanten bleiben übrig, bei denen der Anfangs- oder der Endknoten in X liegt. Eine solche Kante erscheint mit positivem Vorzeichen, wenn sie von X nach $V \setminus X$ läuft und mit negativem Vorzeichen im umgekehrten Fall, genau wie in der Definition von $f(X, \bar{X})$. Damit gilt $M = f(X, \bar{X})$ und daher $f(X, \bar{X}) = m_f$.
(2) Da f ein Fluss ist, gilt für jede Kante e: $0 \leq f(e) \leq c(e)$. Daraus folgt:

$$\begin{aligned} f(X, \bar{X}) = & \sum_{xy \in E, x \in X, y \in V \setminus X} f(xy) \quad - \sum_{xy \in E, x \in V \setminus X, y \in X} f(xy) \quad \leq \\ \leq & \sum_{xy \in E, x \in X, y \in V \setminus X} c(xy) \quad - \quad 0 \quad = \\ = & \, c(X, \bar{X}). \end{aligned}$$

(3) Wenn die Eigenschaft erfüllt ist, dann ist es aus dem Beweis von (2) klar, dass es Gleichheit in der Ungleichung von (2) gibt, so dass $f(X, \bar{X})$ in diesem Fall maximal ist (unter allen möglichen Flüssen im Netzwerk N). Wegen (1) folgt daraus, dass m_f auch maximal ist. □

Definition 2.135 (Hilfsgraph G_f, Verbesserungsweg)**.** Sei $N = (G, s, t, c)$ ein Netzwerk und f ein Fluss in N. Dazu definieren wir einen *Hilfsgraphen* G_f wie folgt. G_f hat dieselbe Knotenmenge wie G und ist auch gerichtet. In G_f gibt es eine Kante vom Knoten x zum Knoten y, wenn entweder $xy \in E(G)$ und $f(xy) < c(xy)$ (Kante vom *Typ 1*) oder $yx \in E(G)$ und $f(yx) > 0$ (Kante vom *Typ 2*). Wenn es in G_f einen gerichteten Weg von s zu t gibt, so ist das ein *Verbesserungsweg*.

Satz 2.136. Sei $N = (G, s, t, c)$ ein Netzwerk und f ein Fluss in N. f hat genau dann maximale Stärke, wenn es in G_f keinen Verbesserungsweg gibt.

Beweis. Nehmen wir zuerst an, es gibt einen Verbesserungsweg W. Seien die Kanten in W $e_1, e_2, \ldots, e_k$. Wenn e_i eine Kante vom Typ 1 ist, dann sei $d_i := c(e_i) - f(e_i) > 0$, bei Kanten vom Typ 2 sei $d_i = f(e_i) > 0$. Sei $d := \min\{d_i\} > 0$. Dann kann man f an allen Kanten in W vom Typ 1 um d erhöhen, an den Kanten in W vom Typ 2 um d verringern. Es ist einfach zu sehen, dass man so auch einen Fluss erhält, undzwar einen, dessen Stärke um d höher ist. Also war f nicht von maximaler Stärke.

Nehmen wir nun an, es gibt keinen Verbesserungsweg. Sei X die Menge der Knoten, die man in G_f von s aus über einen gerichteten Weg erreichen kann. Dann ist $s \in X$ aber $t \notin X$, so dass $E(X, \bar{X})$ in G ein $s\bar{t}$-Schnitt ist. Sei xy eine beliebige Kante von G, die von X nach $V \setminus X$ läuft. Da $y \notin X$, ist xy in G_f nicht enthalten, d.h. $f(xy) = c(xy)$. Sei xy nun eine beliebige Kante von G, die von $V \setminus X$ nach X läuft. Da $x \notin X$, ist yx in G_f nicht enthalten, d.h. $f(xy) = 0$. Damit sind die Bedingungen von Lemma 2.134/(3) erfüllt. Aus dem Lemma folgt, dass f maximale Stärke hat. □

Algorithmus 2.137 (Ford-Fulkerson).
`Input:`
Netzwerk $N = (G, s, t, c)$.
`Output:`
Fluss f mit maximaler Stärke.
`Ablauf:`
Wir fangen mit dem trivialen Fluss an, bei dem $f(e) = 0$ für alle Kanten. In einer allgemeinen Iteration gibt es einen Ausgangsfluss f, zu dem der Hilfsgraph G_f erstellt wird. In G_f wird nach einem Verbesserungsweg gesucht. Findet man so einen Weg, so wird f erhöht, wie im Beweis von Satz 2.136 beschrieben, und die nächste Iteration beginnt mit diesem geänderten Fluss. Findet man keinen Verbesserungsweg, so terminiert der Algorithmus, da f gemäß Satz 2.136 optimal ist.

Satz 2.138 (Edmonds-Karp)**.** Wenn in jeder Iteration der kürzeste Verbesserungsweg (d.h. der Verbesserungsweg mit minimaler Anzahl von Kanten) benutzt wird, terminiert der Algorithmus von Ford und Fulkerson in Polynomialzeit. ⊠

Korollar 2.139. Die Menge der möglichen Flussstärken in einem Netzwerk hat ein Maximum.

Beweis. Der Algorithmus von Ford und Fulkerson liefert dieses Maximum in endlich vielen Schritten. □

Satz 2.140 (Ford-Fulkerson, Max-Flow-Min-Cut)**.** In einem Netzwerk ist die maximale Flussstärke gleich der minimalen $s\bar{t}$-Schnitt-Kapazität.

Beweis. Aus Lemma 2.134 folgt, dass die maximale Flussstärke nicht höher als die minimale $s\bar{t}$-Schnitt-Kapazität sein kann.

Die im Beweis von Satz 2.136 konstruierte Menge X zeigt, dass der Algorithmus von Ford und Fulkerson außer dem Fluss auch einen $s\bar{t}$-Schnitt mit $m_f = f(X, \bar{X}) = c(X, \bar{X})$ liefert. □

Satz 2.141. Wenn alle Kapazitäten ganze Zahlen sind, ist auch die maximale Flussstärke eine ganze Zahl. Es gibt sogar einen Fluss f maximaler Stärke, für den $f(e)$ für jede Kante e eine ganze Zahl ist.

Beweis. Der durch den Algorithmus von Ford und Fulkerson gelieferte Fluss besitzt diese Eigenschaft. □

Bemerkung 2.142. Der ursprüngliche – nicht unbedingt polynomiale – Algorithmus von Ford und Fulkerson stammt aus dem Jahr 1956. Die 1972 veröffentlichte Verbesserung von Edmonds und Karp stellt sicher, dass der Algorithmus in $\mathcal{O}\left(nm^2\right)$ Schritten terminiert. Parallel wurde ein ähnlicher Algorithmus von Dinic entwickelt und 1970 veröffentlicht, der einen Laufzeit von $\mathcal{O}\left(n^2m\right)$ aufweist. 1986 wurde ein ganz anders gearteter Algorithmus, der so genannte Push-Relabel-Algorithmus von Goldberg und Tarjan entwickelt, dessen Laufzeit je nach verwendeter Datenstruktur $\mathcal{O}\left(n^3\right)$, $\mathcal{O}\left(n^2\sqrt{m}\right)$ oder $\mathcal{O}\left(nm\log(n^2/m)\right)$ beträgt.

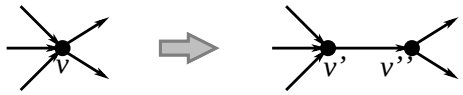

Abbildung 2.10: Konstruktion im Beweis von Satz 2.148

.

2.10 Die Mengerschen Sätze

Lemma 2.143. Sei f ein Fluss positiver Stärke im Netzwerk $N = (G, s, t, c)$. Dann gibt es einen $s \rightsquigarrow t$ Weg in G, dessen Kanten alle einen positiven Flusswert haben.

Beweis. Bezeichne X die Menge der Knoten, die aus s über einen gerichteten Weg, der nur aus Kanten mit positivem Flusswert besteht, erreichbar sind. Nehmen wir indirekt an, dass $t \notin X$. Dann ist $E(X, \bar{X})$ ein $s\bar{t}$-Schnitt. Laut Lemma 2.134 gilt: $0 < m_f = f(X, \bar{X})$. Andererseits, wenn $xy \in E$, $x \in X$ und $y \in V \setminus X$, dann muss $f(xy) = 0$ sein, weil sonst y auch in X wäre. Daraus folgt $f(X, \bar{X}) \leq 0$, was ein Widerspruch ist. □

Lemma 2.144. Sei f ein Fluss mit Stärke k ($k \in \mathbb{N}$) im Netzwerk N, so dass für jede Kante e entweder $f(e) = 0$ oder $f(e) = 1$. Dann kann man aus den Kanten mit $f(e) = 1$ mindestens k paarweise kantendisjunkte $s \rightsquigarrow t$ Wege auswählen.

Beweis. Vollständige Induktion. Für $k = 0$ ist die Aussage trivial. Sei nun $k > 0$ und betrachten wir nur die Kanten mit $f(e) = 1$. Wegen Lemma 2.143 gibt es einen $s \rightsquigarrow t$ Weg W aus solchen Kanten. An den Kanten dieses Weges ändern wir f von 1 zu 0, so erhalten wir f' mit $m_{f'} = k - 1$. Laut Induktionsbedingung gibt es unter den Kanten mit $f'(e) = 1$ mindestens $k - 1$ paarweise kantendisjunkte $s \rightsquigarrow t$ Wege. Diese sind alle auch von W kantendisjunkt, so dass wir zusammen mit W mindestens k paarweise kantendisjunkte Wege erhalten. □

Definition 2.145 (trennende Kantenmenge)**.** Sei $G = (V, E)$ ein (gerichteter/ungerichteter) Graph, $s, t \in V$ zwei Knoten. $E' \subseteq E$ *trennt* s von t, wenn es in $G \setminus E'$ keinen (gerichteten/ungerichteten) Weg von s nach t gibt.

Definition 2.146 (trennende Knotenmenge)**.** Sei $G = (V, E)$ ein (gerichteter/ungerichteter) Graph, $s, t \in V$ zwei Knoten mit $st \notin E$. $V' \subseteq V$ *trennt* s von t, wenn $s, t \notin V'$ und es in $G \setminus V'$ keinen (gerichteten/ungerichteten) Weg von s nach t gibt.

Satz 2.147 (Menger)**.** Sei $G = (V, E)$ ein gerichteter Graph, $s, t \in V$ zwei Knoten. Bezeichne A die maximale Anzahl paarweise kantendisjunkter $s \rightsquigarrow t$ Wege. Bezeichne B die minimale Anzahl der s von t trennenden Kanten. Dann ist $A = B$.

Beweis. Betrachten wir A paarweise kantendisjunkte $s \rightsquigarrow t$ Wege. Um s von t zu trennen, braucht man mindestens eine Kante aus jedem dieser Wege. Da diese alle verschieden sein müssen, gilt $A \leq B$.

Um die andere Richtung zu zeigen, ergänzen wir den Graphen zu einem Netzwerk. Dazu wird für jede Kante e als Kapazität $c(e) := 1$ gesetzt. Sei $E(X, \bar{X})$ ein $s\bar{t}$-Schnitt mit minimaler Kapazität und sei $Y := \{xy \in E(X, \bar{X}) \mid x \in X, y \in V \setminus X\}$. Dann ist $c(X, \bar{X}) = |Y|$. Andererseits, Y trennt s von t, und daher gilt $B \leq |Y|$, also ist die minimale $s\bar{t}$-Schnittkapazität mindestens B. Wegen Satz 2.140 ist die maximale Flussstärke auch mindestens B. Wegen Satz 2.141 gibt es einen Fluss mit Stärke mindestens B, in dem $f(e)$ für jede Kante e entweder 0 oder 1 ist. Gemäß Lemma 2.144 gibt es dann mindestens B paarweise kantendisjunkte $s \rightsquigarrow t$ Wege in G, so dass $A \geq B$. □

Satz 2.148 (Menger)**.** Sei $G = (V, E)$ ein gerichteter Graph, $s, t \in V$ zwei Knoten mit $st \notin E$. Die maximale Anzahl von (abgesehen von den Endknoten) paarweise knotendisjunkten $s \rightsquigarrow t$ Wegen ist gleich der minimalen Anzahl der s von t trennenden Knoten.

Beweis. Wir konstruieren einen anderen gerichteten Graphen G' wie folgt. Zu jedem Knoten v von G gibt es zwei Knoten v' und v'' in G', und es gibt eine Kante von v' zu v''. Für jede Kante uv in G gibt es eine Kante $u''v'$ in G' (siehe Abbildung 2.10).

Betrachten wir eine Menge von (abgesehen von den Endknoten) paarweise knotendisjunkten $s \rightsquigarrow t$ Wegen in G. Es ist klar, dass die entsprechenden $s'' \rightsquigarrow t'$ Wege in G' kantendisjunkt sind. Betrachten wir nun umgekehrt eine Menge von paarweise kantendisjunkten $s'' \rightsquigarrow t'$ Wegen in G'. Dann ist es klar, dass die entsprechenden $s \rightsquigarrow t$ Wege in G abgesehen von den Endknoten paarweise knotendisjunkt sind. Daraus folgt, dass die maximale Anzahl kantendisjunkter Wege in G' und die maximale Anzahl knotendisjunkter Wege in G gleich sind.

Betrachten wir eine Knotenmenge in G, die s von t trennt. Dann ist es klar, dass die entsprechende Kantenmenge in G' s'' von t' trennt. Betrachten wir nun umgekehrt eine Kantenmenge in G', die s'' von t' trennt. Um eine entsprechende Knotenmenge in G zu erhalten, müssen wir zuerst eventuelle Kanten der Form $u''v'$ aus der Menge eliminieren. Das geht, indem die Kante durch $u'u''$ oder durch $v'v''$ ersetzt wird; die Kantenmenge trennt weiterhin s'' von t'. Wenn $u \neq s$, dann kann die erste Variante benutzt werden, wenn $v \neq t$, dann die zweite. Da $st \notin E$, kann immer mindestens eine der Varianten benutzt werden. Durch diese Änderungen kann es im Prinzip vorkommen, dass die Anzahl der ausgewählten Kanten verringert wird. Da das Ergebnis jedoch weiterhin eine s'' von t' trennende Kantenmenge ist, kann das nicht vorkommen, wenn wir von einer minimalen solchen Kantenmenge ausgehen. Die entsprechende Knotenmenge in G trennt s von t. Daraus folgt, dass die minimale Anzahl trennender Kanten in G' und die minimale Anzahl trennender Knoten in G gleich sind. Durch die Anwendung von Satz 2.147 auf G' ist der Beweis nun vollständig. □

Satz 2.149 (Menger)**.** Sei $G = (V, E)$ ein ungerichteter Graph, $s, t \in V$ zwei Knoten. Bezeichne A die maximale Anzahl paarweise kantendisjunkter $s \sim t$ Wege. Bezeichne B die minimale Anzahl der s von t trennenden Kanten. Dann gilt $A = B$.

Beweis. $A \leq B$ ist klar, ähnlich wie bei Satz 2.147. Um die andere Richtung zu zeigen, konstruieren wir einen gerichteten Graphen G' aus G, indem wir alle Kanten von G in beide Richtungen in G' aufnehmen. Wenn eine Menge von Kanten in G' s von t trennt, dann ist es klar, dass die entsprechenden Kanten in G (die höchstens so viele sind) auch s von t trennen. Daraus folgt, dass die minimale Anzahl der s von t trennenden Kanten in G' mindestens B ist. Laut Satz 2.147 gibt es also mindestens B kantendisjunkte $s \rightsquigarrow t$ Wege in G'. Die entsprechenden Wege in G sind nicht unbedingt kantendisjunkt, denn es kann vorkommen, dass ein Weg die Kante uv und ein anderer die Kante vu enthält, d.h. die Wege haben die Form $W_1 + uv + W_2$ und $W_3 + vu + W_4$, wobei W_1 von s zu u, W_2 von v zu t, W_3 von s zu v und W_4 von u zu t läuft (siehe Abbildung 2.11). Dann wählen wir statt dieser beiden Wege zwei andere Wege: $W_1 + W_4$ und $W_3 + W_2$ (nach Entfernung eventueller gerichteter Kreise in diesen neuen Kantenzügen). Es ist leicht zu sehen, dass wir eine Menge von Wegen mit derselben Mächtigkeit erhalten, die in G' weiterhin kantendisjunkt sind und dass die Anzahl der Verletzungen der Kantendisjunktheit in G um mindestens 1 verringert wurde. Diese Änderung wiederholen wir bis die ausgewählten Wege auch in G kantendisjunkt sind. Dann haben wir mindestens B kantendisjunkte Wege in G von s zu t, so dass $A \geq B$. □

Satz 2.150 (Menger)**.** Sei $G = (V, E)$ ein ungerichteter Graph, $s, t \in V$ zwei Knoten, die nicht benachbart sind. Die maximale Anzahl von (abgesehen von den Endknoten) paarweise knotendisjunkten $s \sim t$ Wegen ist gleich der minimalen Anzahl der s von t trennenden Knoten.

Beweis. Wir konstruieren den gerichteten Graphen G' mit der Konstruktion aus dem Beweis von Satz 2.149. Dann entspricht einer Menge von (abgesehen von den Endknoten) knotendisjunkten gerichteten $s \rightsquigarrow t$ Wegen in G' eine Menge von (abgesehen von den Endknoten) knotendisjunkten $s \sim t$ Wegen in G und auch umgekehrt. Daher ist die maximale Anzahl solcher Wege in G und G' identisch. Ähnlicherweise entspricht

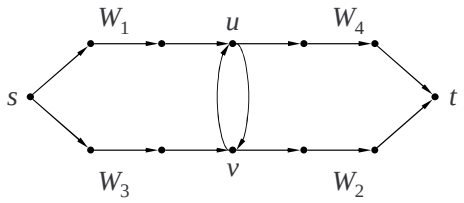

Abbildung 2.11: Zum Beweis von Satz 2.149

einer Menge von Knoten, die in G' s von t trennen, eine Menge von Knoten in G, die s von t in G trennen, und auch umgekehrt. Daher ist die minimale Anzahl solcher Knoten in G und G' identisch. Zusammen mit Satz 2.148 für G' ergibt sich die gewünschte Aussage. □

Behauptung 2.151. Sei $G = (V, E)$ ein gerichteter oder ungerichteter Graph, $s, t \in V$ und $st \notin E$. Seien W_1 und W_2 zwei (abgesehen von ihren Endknoten) knotendisjunkte Wege von s zu t. Dann sind W_1 und W_2 auch kantendisjunkt.

Beweis. Indirekt, wenn W_1 und W_2 eine gemeinsame Kante xy hätten, dann wären x und y gemeinsame Knoten von W_1 und W_2. Da W_1 und W_2 aber abgesehen von ihren Endknoten keine gemeinsamen Knoten haben, würde daraus $x = s$ und $y = t$, und damit $st \in E$ folgen. □

2.11 Mehrfacher Zusammenhang

Definition 2.152 (k-facher Zusammenhang, k-facher Kantenzusammenhang)**.** Ein Graph $G = (V, E)$ ist *k-fach zusammenhängend*, wenn er mindestens $k + 1$ Knoten hat und für alle $X \subseteq V$ mit $|X| < k$ gilt, dass $G \setminus X$ zusammenhängend ist. Ein Graph $G = (V, E)$ ist *k-fach kantenzusammenhängend*, wenn für alle $Y \subseteq E$ mit $|Y| < k$ gilt, dass $G \setminus Y$ zusammenhängend ist.

Bemerkung 2.153. Es ist klar, dass ein Graph mit mindestens 2 Knoten genau dann 1-fach zusammenhängend bzw. 1-fach kantenzusammenhängend ist, wenn er zusammenhängend ist.

Satz 2.154. Ein Graph ist genau dann k-fach kantenzusammenhängend, wenn alle Knotenpaare durch mindestens k kantendisjunkte Wege verbunden sind.

Beweis. Nehmen wir zuerst an, dass der Graph k-fach kantenzusammenhängend ist und betrachten wir zwei Knoten x, y. Da der Graph k-fach kantenzusammenhängend ist, braucht man mindestens k Kanten, um x von y zu trennen. Aus Satz 2.149 folgt, dass es mindestens k kantendisjunkte $x \sim y$ Wege gibt.

Betrachten wir jetzt die andere Richtung und nehmen wir an, dass alle Knotenpaare durch mindestens k kantendisjunkte Wege verbunden sind. Nehmen wir indirekt an, dass der Graph nicht k-fach kantenzusammenhängend ist. Dann gibt es eine Menge Y von höchstens $k - 1$ Kanten, so dass $G \setminus Y$ nicht zusammenhängend ist. Seien u und v zwei Knoten, die in $G \setminus Y$ nicht in derselben Komponente sind. Dann reichen weniger als k Kanten aus, um u und v im ursprünglichen Graphen voneinander zu trennen. Gemäß Satz 2.149 kann es also keine k kantendisjunkte Wege zwischen u und v geben. □

Satz 2.155. Ein Graph ist genau dann k-fach zusammenhängend, wenn er mindestens $k + 1$ Knoten hat und alle Knotenpaare durch mindestens k knotendisjunkte Wege verbunden sind.

Beweisskizze. Analog zum Beweis von Satz 2.154. Es gibt einen einzigen Unterschied: Bei der ersten Richtung kann man Satz 2.150 nur verwenden, wenn $xy \notin E$. Wenn $xy \in E$, dann kann man beweisen, dass man im Graphen $G \setminus \{xy\}$ mindestens $k-1$ Knoten braucht, um x und y zu trennen, so dass Satz 2.150 in diesem Graphen $k - 1$ knotendisjunkte Wege garantiert. Zusammen mit der Kante xy sind das dann k Wege. □

Korollar 2.156. Ein k-fach zusammenhängender Graph ist auch k-fach kantenzusammenhängend (aber umgekehrt gilt das nicht).

Satz 2.157 (Menger)**.** Ein Graph mit mindestens 3 Knoten ist genau dann 2-fach zusammenhängend, wenn es für alle Knotenpaare einen Kreis gibt, der diese Knoten enthält.

Beweis. Wenn der Graph 2-fach zusammenhängend ist, dann gibt es zwischen allen Knotenpaaren laut Satz 2.155 zwei knotendisjunkte Wege. Diese beiden Wege ergeben zusammen einen Kreis, der die beiden Knoten enthält.

Umgekehrt, wenn es für alle Knotenpaare einen solchen Kreis gibt, dann garantiert der jeweilige Kreis zwei knotendisjunkte Wege zwischen allen Knotenpaaren, also ist der Graph laut Satz 2.155 2-fach zusammenhängend. □

Satz 2.158 (Menger)**.** Ein Graph mit mindestens 3 Knoten und ohne isolierte Knoten ist genau dann 2-fach zusammenhängend, wenn es für alle Kantenpaare einen Kreis gibt, der diese Kanten enthält.

Beweis. Wenn es für alle Kantenpaare einen solchen Kreis gibt, dann gibt es auch für alle Knotenpaare einen solchen Kreis. Denn für beide Knoten gibt es eine inzidente Kante, und der Kreis, der diese Kanten enthält, enthält auch die beiden Knoten. (Es ist leicht zu sehen, dass es nicht möglich ist, dass die beiden Knoten nur zu derselben Kante inzident sind, so dass man wirklich zwei verschiedene Kanten wählen kann.) Mit Satz 2.157 folgt daraus, dass der Graph 2-fach zusammenhängend ist.

Umgekehrt, nehmen wir an, dass der Graph 2-fach zusammenhängend ist und betrachten wir zwei beliebige Kanten e und f. Ergänzen wir den Graphen durch zwei neue Knoten x und y und tauschen wir die Kanten e und f gegen zwei Wege der Länge 2 aus, die jeweils durch x bzw. y gehen. (D.h., e und f werden von x bzw. y jeweils in zwei Kanten zerlegt.) Es ist leicht zu sehen, dass der Graph weiterhin 2-fach zusammenhängend bleibt. Gemäß Satz 2.157 gibt es einen Kreis, der x und y enthält. Dieser Kreis enthält im ursprünglichen Graphen e und f. □

Satz 2.159 (Dirac)**.** Sei $k \geq 2$. Wenn G k-fach zusammenhängend ist, dann gibt es zu beliebigen k Knoten einen Kreis, der diese Knoten enthält. ⊠

2.12 Planarität

Definition 2.160 (Planarität, Gebiete)**.** Ein Graph ist *planar*, wenn er eine *planare Zeichnung* hat, d.h. wenn er in der Ebene gezeichnet werden kann, ohne dass die Kanten sich schneiden würden. Die planare Zeichnung des Graphen teilt die Ebene in *Gebiete*. Die Anzahl der Gebiete wird mit g bezeichnet.

Satz 2.161. Eines der Gebiete ist unbeschränkt, alle anderen sind beschränkt. ⊠

Satz 2.162. Wenn die Kante e in einem Kreis enthalten ist, dann gehört e in jeder planaren Zeichnung des Graphen zum Rand von genau zwei Gebieten. Wenn e hingegen in keinem Kreis enthalten ist (d.h. eine Brücke ist), dann gehört e in jeder planaren Zeichnung des Graphen zum Rand genau eines Gebietes. ⊠

Satz 2.163. Bei jeder planaren Zeichnung eines Baumes gilt $g = 1$. ⊠

Satz 2.164. Ein Graph ist genau dann planar, wenn er auf die Oberfläche einer Kugel gezeichnet werden kann, ohne dass die Kanten sich schneiden würden.

Beweisskizze. Die Kugel wird auf die Ebene gestellt; der Berührungspunkt sei der Südpol. Wir definieren eine eineindeutige Abbildung zwischen den Punkten der Ebene und den Punkten der Kugeloberfläche außer des Nordpols (stereographische Projektion aus dem Nordpol). Das Bild eines Punktes P der Ebene erhält man, indem man P und den Nordpol mit einer Gerade verbindet; diese Gerade schneidet die Kugeloberfläche in genau einem weiteren Punkt, der das Bild P' ist. Ähnlicherweise, um das Bild eines Punktes

Q der Kugeloberfläche zu bestimmen, verbindet man Q und den Nordpol mit einer Gerade; diese Gerade schneidet die Ebene in genau einem Punkt, der das Bild Q' ist.

Wenn der Graph kreuzungsfrei in der Ebene gezeichnet ist, dann liefert die stereographische Projektion eine kreuzungsfreie Zeichnung auf der Kugeloberfläche und umgekehrt, wenn der Graph kreuzungsfrei auf der Kugeloberfläche gezeichnet ist, dann liefert die stereographische Projektion eine kreuzungsfreie Zeichnung in der Ebene. □

Bemerkung 2.165. Es gibt auch Oberflächen, die lokal auch so aussehen wie die Ebene, auf die jedoch auch nicht planare Graphen gezeichnet werden können, ohne dass die Kanten sich schneiden würden. Beispiel: Torus.

Satz 2.166 (Euler-Formel)**.** Für einen zusammenhängenden planaren Graphen gilt: $m = n + g - 2$.

Beweis. Betrachten wir eine konkrete planare Zeichnung des Graphen. Da der Graph zusammenhängend ist, gilt $m \geq n - 1$. Für gegebenes n wenden wir vollständige Induktion nach m an. Wenn $m = n - 1$, ist der Graph laut Korollar 2.37 ein Baum, also gilt laut Satz 2.163 $g = 1$ und damit $m = n-1 = n+1-2 = n + g - 2$. Sei jetzt $m > n - 1$ und nehmen wir an, dass der Satz für alle Graphen mit n Knoten und weniger als m Kanten bereits bewiesen wurde. Da $m > n - 1$, enthält der Graph laut Korollar 2.37 einen Kreis. Sei e eine Kante in diesem Kreis. Laut Satz 2.162 gehört e zum Rand von genau zwei Gebieten. Entfernt man die Kante e, so werden diese Gebiete vereinigt, also hat der so entstehende Graph $n' = n$ Knoten, $m' = m - 1$ Kanten und $g' = g - 1$ Gebiete (in seiner aktuell betrachteten planaren Zeichnung). Laut Induktionsbedingung ist $m' = n' + g' - 2$. Daraus folgt $m = m' + 1 = n' + g' - 1 = n + g - 2$. □

Satz 2.167 (Euler-Formel für nicht zusammenhängende Graphen)**.** Für einen planaren Graphen mit c Komponenten gilt: $m = n + g - c - 1$.

Beweis. In Komponente i sei die Anzahl der Knoten n_i, die Anzahl der Kanten m_i und die Anzahl der Gebiete g_i. Dann gilt $n = \sum_{i=1}^{c} n_i$, $m = \sum_{i=1}^{c} m_i$ und $g = 1 + \sum_{i=1}^{c}(g_i - 1) = 1 - c + \sum_{i=1}^{c} g_i$ (bei der Formel für g muss man beachten, dass das unbeschränkte Gebiet nur einmal gezählt werden soll). Des Weiteren gilt wegen Satz 2.166 für jedes $i = 1, \ldots, c$: $m_i = n_i + g_i - 2$. Daraus folgt: $m = \sum_{i=1}^{c} m_i = \sum_{i=1}^{c}(n_i + g_i - 2) = n + g + c - 1 - 2c$, was eben die gewünschte Formel liefert. □

Korollar 2.168. Bei einem planaren Graphen hängt g nur vom Graphen ab, nicht von der planaren Zeichnung.

Lemma 2.169. Sei G ein planarer Graph, der mindestens einen Kreis enthält und in dem jeder Kreis aus mindestens t Kanten besteht. Dann besteht der Rand jedes Gebietes in jeder planaren Zeichnung des Graphen aus mindestens t Kanten. ⊠

Satz 2.170. Sei G ein planarer Graph, der mindestens einen Kreis enthält und in dem jeder Kreis aus mindestens t Kanten besteht, wobei $t \geq 3$. Dann gilt: $m \leq \frac{t(n-2)}{t-2}$.

Beweis. Betrachten wir eine planare Zeichnung des Graphen. Seien die Gebiete $A_1, A_2, \ldots, A_g$ und für jedes i bezeichne a_i die Anzahl der Kanten, aus denen der Rand von A_i besteht. Laut Lemma 2.169 ist $a_i \geq t$ für jedes i. Daher gilt $\sum a_i \geq tg$. Andererseits gehört gemäß Satz 2.162 jede Kante des Graphen zum Rand von höchstens zwei Gebieten. Damit gilt $\sum a_i \leq 2m$. Aus diesen beiden Ungleichungen folgt $tg \leq 2m$. Andererseits folgt aus Satz 2.167: $g = m - n + c + 1 \geq m - n + 2$. Zusammen ergibt das $t(m - n + 2) \leq 2m$ und damit $m(t - 2) \leq t(n - 2)$. Da $t \geq 3$, kann man nun mit $t - 2$ dividieren um die gewünschte Formel zu erhalten. □

Satz 2.171. Sei G ein einfacher planarer Graph mit $n \geq 3$. Dann gilt $m \leq 3n - 6$.

Beweis. Wenn G kreisfrei ist, dann ist gemäß Korollar 2.31 $m \leq n-1$, was für $n \geq 3$ kleiner ist als $3n-6$. Sonst enthält G mindestens einen Kreis und da der Graph einfach ist, besteht jeder Kreis aus mindestens 3 Kanten. Dann folgt die Aussage aus Satz 2.170 mit $t = 3$. □

Korollar 2.172. K_5 ist nicht planar.

Beweis. K_5 ist ein einfacher Graph mit $n \geq 3$. Wenn er planar wäre, müsste laut Satz 2.171 $10 = m \leq 3n - 6 = 9$ gelten. □

Satz 2.173. Sei $G = (V, E)$ ein einfacher planarer Graph und sei der minimale Grad $\delta = \min_{v \in V} d(v)$. Dann gilt $\delta \leq 5$.

Beweis. Für $n \leq 2$ ist die Behauptung trivial, so dass wir im Folgenden annehmen, dass $n \geq 3$. Es reicht zu beweisen, dass der durchschnittliche Grad $\hat{d} = \frac{1}{n} \sum_{v \in V} d(v)$ kleiner als 6 ist. Aus Satz 2.7 folgt, dass $\hat{d} = \frac{2m}{n}$. Zusammen mit Satz 2.171 ergibt sich daraus $\hat{d} \leq \frac{6n-12}{n} < 6$. □

Satz 2.174. $K_{3,3}$, der vollständige bipartite Graph mit $|A| = |B| = 3$, ist nicht planar.

Beweis. Es ist klar, dass $K_{3,3}$ ein einfacher Graph ist, der Kreise enthält. Ein Kreis der Länge 3 ist aber wegen Satz 2.111 nicht möglich. Wenn der Graph planar wäre, könnte man daher Satz 2.170 mit $t = 4$ anwenden und es würde $m \leq 2n - 4$ folgen, was bei diesem Graphen mit $n = 6$ und $m = 9$ nicht erfüllt ist. □

Definition 2.175 (Unterteilung)**.** Sei $G = (V, E)$ ein Graph, X eine nicht leere Menge mit $X \cap V = \emptyset$. Den Graphen $G' = (V \cup X, E')$ konstruiert man aus G, indem man einige Kanten von G durch Wege ersetzt, wobei die inneren Knoten dieser Wege aus X stammen und jedes Element von X in genau einem solchen Weg enthalten ist. Dann ist G' eine *Unterteilung* von G.

Definition 2.176 (topologischer Minor)**.** Der Graph H ist ein *topologischer Minor* des Graphen G, wenn es eine Unterteilung von H gibt, die mit einem Teilgraphen von G isomorph ist.

Satz 2.177. Wenn H ein topologischer Minor von G ist und H nicht planar ist, dann ist G auch nicht planar.

Beweis. Sei H' der Teilgraph von G, der mit einer Unterteilung von H isomorph ist. Wenn G planar wäre, dann wäre H' auch planar und H damit auch. □

Korollar 2.178. Wenn ein Graph K_5 oder $K_{3,3}$ als topologischen Minor hat, ist er nicht planar. □

Satz 2.179 (Kuratowski)**.** Ein Graph ist genau dann planar, wenn er weder K_5 noch $K_{3,3}$ als topologischen Minor hat. ⊠

Satz 2.180 (Fáry-Wagner)**.** Wenn ein Graph planar ist, dann hat er eine planare Zeichnung, in der jede Kante eine gerade Linie ist. ⊠

2.12.1 Dualität

Definition 2.181 (dualer Graph)**.** Sei $G = (V, E)$ ein planarer Graph. Sein *dualer Graph* $G^* = (V^*, E^*)$ wird wie folgt konstruiert. V^* besteht aus den Gebieten von G. Wenn die Kante $e \in E$ zum Rand von zwei Gebieten gehört, dann gibt es eine Kante $e^* \in E^*$ zwischen den entsprechenden Knoten von G^*. (Wenn es mehrere Kanten gibt, die zum Rand von zwei gegebenen Gebieten gehören, dann sind die entsprechenden Knoten in G^* durch mehrere Parallelkanten verbunden.) Wenn die Kante $e \in E$ zum Rand von nur einem Gebiet gehört, dann gibt es eine Schlinge $e^* \in E^*$, die zum entsprechenden Knoten von G^* inzident ist.

Satz 2.182. Sei G ein planarer Graph. Dann ist G^* planar und zusammenhängend. ⊠

Bemerkung 2.183. Die Notation G^* ist etwas irreführend, weil G^* nicht vom Graphen G, sondern von seiner planaren Zeichnung abhängt. Die dualen Graphen, die zu den verschiedenen planaren Zeichnungen eines Graphen gehören, sind nicht notwendigerweise isomorph (siehe Abbildung 2.12).

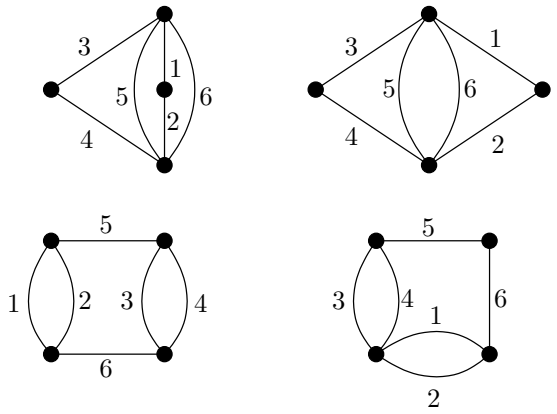

Abbildung 2.12: In der oberen Zeile sind zwei planare Zeichnungen desselben Graphen zu sehen, in der unteren Zeile die entsprechenden Dualen.

Definition 2.184 (schwache Isomorphie)**.** $G = (V, E)$ und $G' = (V', E')$ sind *schwach isomorph*, wenn es eine eineindeutige Abbildung $f : E \to E'$ gibt, so dass eine Menge von Kanten $K \subseteq E$ genau dann einen Kreis in G bildet, wenn die entsprechenden Kanten $\{f(e) \mid e \in K\} \subseteq E'$ einen Kreis in G' bilden.

Satz 2.185 (Whitney)**.** Sei G ein zusammenhängender, planarer Graph. Dann gilt:
(1) Die dualen Graphen, die zu verschiedenen planaren Zeichnungen von G gehören, sind schwach isomorph.
(2) Wenn G und H schwach isomorph sind, dann ist H auch planar und G^* und H^* sind auch schwach isomorph (unabhängig von den gewählten planaren Zeichnungen von G und H).
(3) $(G^*)^*$ ist schwach isomorph zu G (unabhängig von den gewählten planaren Zeichnungen von G und G^*). ⊠

Satz 2.186 (Whitney)**.** Die Graphen G und H sind genau dann schwach isomorph, wenn man aus H zu einem mit G isomorphen Graphen gelangen kann, in dem man die folgenden Operationen benutzt (siehe Abbildung 2.13):

i) Sei x ein Knoten in H mit der Eigenschaft, dass die x enthaltende Komponente von H in zwei Komponenten zerfällt, wenn x entfernt wird. Die Komponente und der Knoten x werden aufgespaltet.

ii) Zwei Knoten x und x' aus verschiedenen Komponenten von H werden verschmolzen.

iii) Nehmen wir an, eine Komponente von H entsteht aus zwei disjunkten Graphen H_1 und H_2, indem der Knoten x aus H_1 mit x' aus H_2 und y aus H_1 mit y' aus H_2 verschmolzen wird. Dann trennen wir H_1 und H_2 und vereinigen sie wieder, indem x mit y' und y mit x' verschmolzen wird.

iv) Isolierte Knoten werden weggelassen oder dazugenommen. ⊠

Korollar 2.187. Seien G und H schwach isomorphe Graphen. Wenn G keinen isolierten Knoten enthält und H 3-fach zusammenhängend ist, dann sind G und H isomorph.

Beweis. Wenn H 3-fach zusammenhängend ist, dann kann keine der Operationen von Satz 2.186 benutzt werden (bis auf das Hinzufügen isolierter Knoten, aber damit kann man nicht zu einem mit G isomorphen Graphen gelangen, weil G keine isolierten Knoten enthält). Wenn G und H doch schwach isomorph sind, kann das nur sein, wenn sie isomorph sind. □

Satz 2.188. Sei G ein planarer Graph, in dem die Kanten $e_1, e_2, \ldots, e_k$ einen Kreis bilden. Dann bilden die Kanten $e_1^*, e_2^*, \ldots, e_k^*$ eine minimale Schnittmenge in G^*. Umgekehrt, wenn $e_1, e_2, \ldots, e_k$ eine minimale Schnittmenge von G bilden, dann bilden $e_1^*, e_2^*, \ldots, e_k^*$ einen Kreis in G^*. ⊠

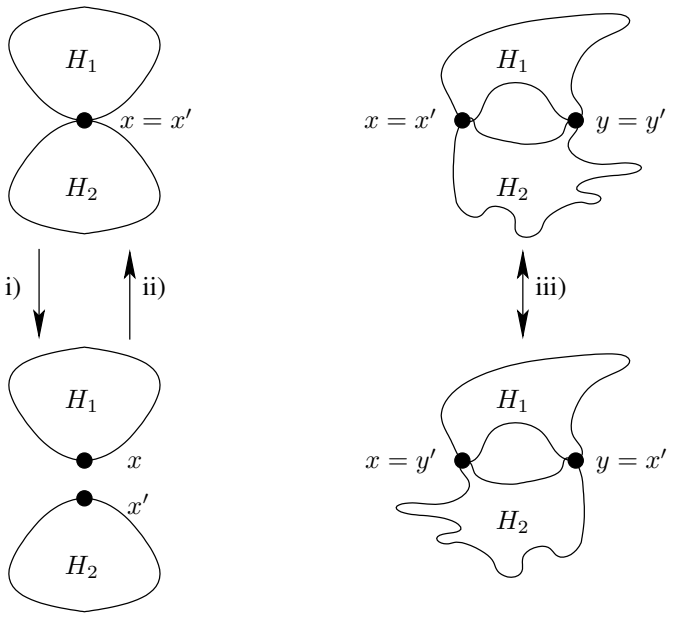

Abbildung 2.13: Operationen, die zu schwach isomorphen Graphen führen

Korollar 2.189. Sei G ein planarer Graph, e eine Schlinge. Dann ist e^* eine Brücke in G^*. Umgekehrt, wenn e eine Brücke in G ist, dann ist e^* eine Schlinge in G^*.

Beweis. Eine Schlinge ist ein spezieller Kreis der Mächtigkeit 1, eine Brücke ist eine spezielle minimale Schnittmenge der Mächtigkeit 1. □

Satz 2.190. Sei G ein zusammenhängender planarer Graph, in dem die Kanten $F := \{e_1, e_2, \ldots, e_k\}$ einen Spannbaum bilden. Dann bilden die Kanten $F^* := E^* \setminus \{e_1^*, e_2^*, \ldots, e_k^*\}$ einen Spannbaum von G^*. ⊠

Korollar 2.191 (Euler-Formel)**.** Für einen zusammenhängenden planaren Graphen gilt: $m = n + g - 2$.

Beweis. Mit der Notation von Satz 2.190, unter Verwendung von Satz 2.29: $|F| = n - 1$, $|F^*| = g - 1$ (weil $|V^*| = g$) und für jede Kante $e \in E$ gilt, dass entweder $e \in F$ oder $e^* \in F^*$. Daher gilt $m = |F| + |F^*| = n - 1 + g - 1 = n + g - 2$. □

Bemerkung 2.192. Die wichtigsten dualen Begriffspaare sind in Tabelle 2.1 zusammengefasst.

Kante	$\Longleftrightarrow$	Kante
Knoten	$\Longleftrightarrow$	Gebiet
Kreis	$\Longleftrightarrow$	Minimale Schnittmenge
Schlinge	$\Longleftrightarrow$	Brücke
Spannbaum	$\Longleftrightarrow$	Komplement eines Spannbaumes

Tabelle 2.1: Duale Begriffe in planaren Graphen

Satz 2.193. Nehmen wir an, dass zwei, aus zweipoligen Elementen bestehende elektrische Netzwerke als Graphen zueinander dual sind. Dann lassen sich die Gleichungen zur Berechnung der Spannungen und Ströme in den beiden Netzwerken ineinander überführen, in dem man die in Tabelle 2.2 angegebenen Begriffe mit ihren Paaren umtauscht. ⊠

Spannung	$\Longleftrightarrow$	Strom
Spannungsquelle	$\Longleftrightarrow$	Stromquelle
Kurzschluss	$\Longleftrightarrow$	Unterbrechung
Parallelschaltung	$\Longleftrightarrow$	Reihenschaltung
Widerstand	$\Longleftrightarrow$	Leitwert
Induktivität	$\Longleftrightarrow$	Kapazität
Maschensatz	$\Longleftrightarrow$	Knotensatz

Tabelle 2.2: Duale Begriffe bei elektrischen Netzwerken

Definition 2.194 (kombinatorisch dual)**.** Der Graph $G' = (V', E')$ ist *kombinatorisch dual* (auch abstrakt dual genannt) zum Graphen $G = (V, E)$, wenn es eine eineindeutige Abbildung $f : E \to E'$ gibt, so dass eine Menge von Kanten $K \subseteq E$ genau dann einen Kreis in G bildet, wenn die entsprechenden Kanten $\{f(e) \mid e \in K\} \subseteq E'$ eine minimale Schnittmenge in G' bilden.

Satz 2.195 (Whitney)**.** Zum Graphen G gibt es genau dann einen kombinatorisch dualen Graphen G', wenn G planar ist. In diesem Fall ist G^* kombinatorisch dual zu G. ⊠

2.13 Färbungen

2.13.1 Knotenfärbung

In diesem Kapitel handelt es sich immer um schlichte Graphen.

Definition 2.196. Sei G ein Graph und $k \geq 1$ eine ganze Zahl. Eine *Färbung* von G mit k Farben ordnet jedem Knoten von G eine von k Farben zu, so dass benachbarten Knoten nicht dieselbe Farbe zugeordnet wird. G ist mit k Farben *färbbar*, wenn es eine solche Färbung gibt. Die Knoten von G, denen bei einer gegebenen Färbung dieselbe Farbe zugeordnet wird, bilden eine *Farbenklasse*.

Behauptung 2.197. Sei G ein Graph mit n Knoten und mindestens einer Kante. Dann gelten folgende Eigenschaften:

- G ist mit 1 Farbe nicht färbbar.
- G ist mit n Farben färbbar.
- Wenn G mit k_1 Farben färbbar ist und $k_2 > k_1$, dann ist G auch mit k_2 Farben färbbar.
- Wenn G mit k_1 Farben nicht färbbar ist und $k_2 < k_1$, dann ist G mit k_2 Farben auch nicht färbbar.

□

Definition 2.198. Die *chromatische Zahl* eines Graphen G ist die kleinste Zahl k, so dass G mit k Farben färbbar ist. Die chromatische Zahl von G wird mit $\chi(G)$ bezeichnet.

Satz 2.199. Sei G ein Graph mit mindestens einer Kante. G ist genau dann bipartit wenn $\chi(G) = 2$.

Beweis. Da G mindestens eine Kante enthält, gilt $\chi(G) \geq 2$. Wenn G bipartit ist, dann ist er mit 2 Farben färbbar, indem man den Knoten in der einen Klasse die erste Farbe, den Knoten in der anderen Klasse die zweite Farbe zuordnet. Umgekehrt, wenn G mit 2 Farben färbbar ist, dann definieren die beiden Farbenklassen eine Bipartition. □

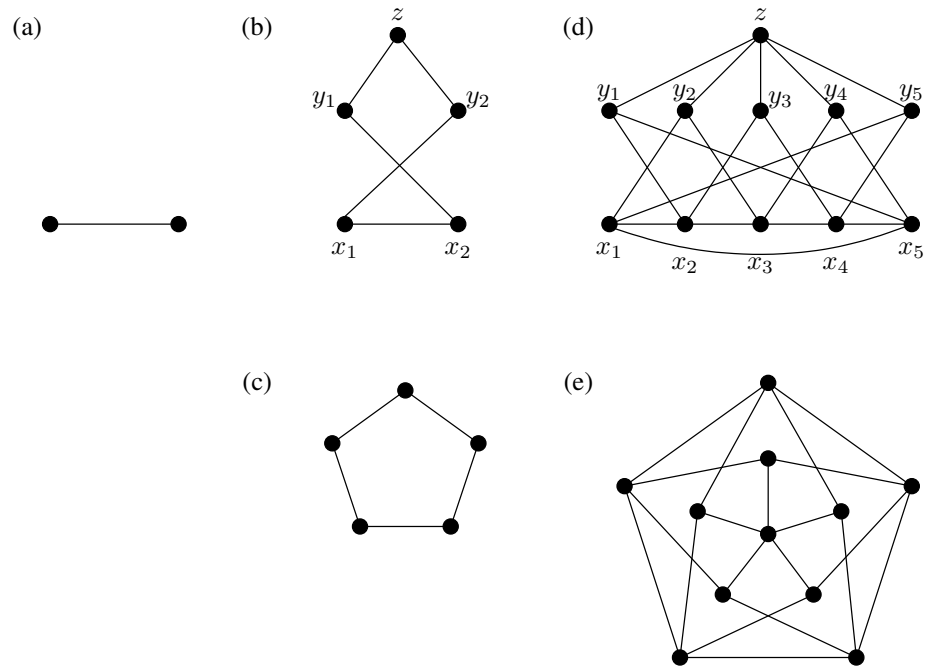

Abbildung 2.14: Beispiele für die Mycielski-Konstruktion: (a) G_2. (b) Konstruktion von G_3 aus G_2. (c) Andere Darstellung von G_3. (d) Konstruktion von G_4 aus G_3. (e) Andere Darstellung von G_4.

Definition 2.200. Im Graphen G ist der Teilgraph H eine *Clique*, wenn H ein vollständiger Graph ist. Die *Cliquezahl* von G ist die maximale Anzahl der Knoten in einer Clique von G. Die Cliquezahl von G wird mit $\omega(G)$ bezeichnet.

Satz 2.201. In jedem Graphen G gilt $\omega(G) \leq \chi(G)$.

Beweis. Sei H eine Clique von G mit $\omega(G)$ Knoten. Da H ein vollständiger Graph ist, muss jedem Knoten von H eine andere Farbe zugeordnet werden. □

Der nächste Satz zeigt, dass die Differenz zwischen $\omega(G)$ und $\chi(G)$ beliebig groß sein kann.

Satz 2.202 (Mycielski-Konstruktion)**.** Für jedes $k \geq 2$ existiert ein Graph G_k mit $\omega(G_k) = 2$ und $\chi(G_k) = k$.

Beweis. G_2 sei der Graph der aus zwei benachbarten Knoten besteht; es gilt offenbar $\omega(G_2) = \chi(G_2) = 2$. Nun nehmen wir an, dass G_k mit den gewünschten Eigenschaften bereits vorliegt, und konstruieren wir daraus G_{k+1}. Die Knoten von G_k seien $x_1, x_2, \ldots, x_n$. Diese ergänzen wir mit $n + 1$ neuen Knoten: $y_1, y_2, \ldots, y_n$ und z. Jeder neue Knoten y_i wird mit den Nachbarn von x_i in G_k verbunden (aber nicht mit x_i selbst). Des Weiteren wird z mit jedem y_i verbunden (siehe Abbildung 2.14).

Zuerst beweisen wir, dass $\omega(G_{k+1}) = 2$. Es ist trivial, dass $\omega(G_{k+1}) \geq 2$. Nehmen wir indirekt an, dass $\omega(G_{k+1}) \geq 3$, d.h. es gibt ein Dreieck in G_{k+1}. Wenn z in diesem Dreieck enthalten wäre, dann müssten die beiden anderen Knoten des Dreiecks y_i und y_j sein, aber diese sind nicht benachbart. Aus diesem Grund kann auch nicht mehr als ein Knoten des Dreiecks in $y_1, \ldots, y_n$ sein. Alle Knoten des Dreiecks können nicht in G_k enthalten sein, weil $\omega(G_k) = 2$. Der einzig übrige Fall ist, wenn das Dreieck aus den Knoten y_i, x_j und x_l besteht. Da y_i mit x_j und mit x_l verbunden ist, folgt daraus einerseits, dass $i \neq j$ und $i \neq l$, andererseits aber auch, dass auch x_i mit x_j und mit x_l verbunden ist. Das würde aber bedeuten, dass x_i, x_j und x_l ein Dreieck in G_k bilden. Das kann nicht sein, da $\omega(G_k) = 2$.

Als Nächstes zeigen wir, dass $\chi(G_{k+1}) \leq k+1$, d.h. G_{k+1} ist mit $k+1$ Farben färbbar. Wir wissen, dass G_k mit k Farben färbbar ist, d.h. die Knoten $x_1, \ldots, x_k$ können mit den ersten k Farben gefärbt werden. Jeder y_i soll dieselbe Farbe erhalten wie der zugehörige x_i. Schließlich erhält z die $k+1$., bisher unbenutzte Farbe. Es ist klar, dass wir so eine gültige Färbung von G_{k+1} mit $k+1$ Farben erhalten.

Es bleibt noch zu zeigen, dass $\chi(G_{k+1}) \geq k+1$, d.h. G_{k+1} ist mit weniger als $k+1$ Farben nicht färbbar. Indirekt nehmen wir an, dass G_{k+1} mit k Farben färbbar ist, und betrachten wir so eine Färbung. Die Farbe eines Knotens v sei $c(v) \in \{1, 2, \ldots, k\}$. Wir können ohne Beschränkung der Allgemeinheit annehmen, dass $c(z) = k$. Daraus folgt für jedes y_i: $c(y_i) \in \{1, 2, \ldots, k-1\}$. Nun konstruieren wir basierend auf c eine andere Färbung c' der x_i Knoten, also des Graphen G_k. Wenn $c(x_i) < k$, dann sei $c'(x_i) := c(x_i)$. Sonst, d.h. wenn $c(x_i) = k$, sei $c'(x_i) := c(y_i)$. In beiden Fällen gilt $c'(x_i) \in \{1, 2, \ldots, k-1\}$, d.h. c' ist eine Färbung mit $k-1$ Farben. Wir müssen aber noch zeigen, dass c' wirklich eine Färbung ist, d.h. benachbarte Knoten haben verschiedene Farben. Seien x_i und x_j zwei benachbarte Knoten. Folgende Fälle müssen unterschieden werden:

- Wenn $c(x_i) < k$ und $c(x_j) < k$, dann ist $c'(x_i) = c(x_i) \neq c(x_j) = c'(x_j)$, also haben sie unterschiedliche Farben in der Färbung c'.
- Wenn z.B. $c(x_i) = k$ aber $c(x_j) < k$, dann gilt $c'(x_i) = c(y_i) \neq c(x_j) = c'(x_j)$, da y_i und x_j in G_{k+1} benachbart sind. Der Fall $c(x_i) < k$, $c(x_j) = k$ ist analog.
- Der letzte Fall wäre $c(x_i) = c(x_j) = k$, das kann aber nicht sein, da c eine Färbung ist.

Alles in allem haben wir also gezeigt, dass c' eine Färbung von G_k mit $k-1$ Farben ist. Wegen $\chi(G_k) = k$ ist das aber ein Widerspruch. □

Satz 2.203. In jedem Graphen G gilt $\frac{n}{\alpha(G)} \leq \chi(G)$.

Beweis. Betrachten wir eine Färbung des Graphen mit χ Farben. Die Anzahl der Knoten in der i. Farbenklasse bezeichnen wir mit a_i. Es ist klar, dass jede Farbenklasse eine unabhängige Knotenmenge ist, und damit gilt $a_i \leq \alpha$. Daraus folgt $n = \sum_{i=1}^{\chi} a_i \leq \chi\alpha$, was eben zur gewünschten Formel führt. □

Bemerkung 2.204. Auch bei dieser unteren Schranke kann die Lücke beliebig groß sein. Zum Beispiel betrachten wir einen Graphen, der aus K_N und N isolierten Knoten besteht. Hier ist $\chi = N$, $\alpha = N+1$, $n = 2N$ und damit $\frac{n}{\alpha} = \frac{2N}{N+1} < 2$.

Algorithmus 2.205 (Gierige Färbung).
`Input`:
Ein Graph G.
`Output`:
Eine Färbung von G mit einer nicht unbedingt optimalen Anzahl von Farben.
`Ablauf`:
Initialisierung: alle Knoten sind ungefärbt.
Wir iterieren in einer beliebigen Reihenfolge über die Knotenmenge und färben einen Knoten nach dem anderen. Der nächste Knoten erhält die Farbe minimaler Index, die unter den Farben seiner schon gefärbten Nachbarn noch nicht vorkommt.

Definition 2.206. Die maximale Gradzahl im Graphen G wird mit $\Delta(G)$ bezeichnet.

Satz 2.207. Die gierige Färbung benutzt höchstens $\Delta(G)+1$ Farben.

Beweis. Als ein Knoten gefärbt wird, schließen seine bereits gefärbten Nachbarn höchstens Δ Farben aus, d.h. im schlimmsten Fall erhält der Knoten die $\Delta+1$. Farbe. □

Korollar 2.208. In jedem Graphen G gilt: $\chi(G) \leq \Delta(G)+1$. □

Bemerkung 2.209. Sei k die Anzahl der von der gierigen Färbung benötigten Farben. Dann ist also $\chi(G) \leq k \leq \Delta(G)+1$. Die Lücke kann beliebig groß sein sowohl zwischen χ und k als auch zwischen k und $\Delta+1$. Dazu betrachten wir als Beispiel den bipartiten Graphen $G = (A, B, E)$ mit $A = \{x_1, \ldots, x_N\}$, $B = \{y_1, \ldots, y_N\}$ und $E = \{x_i y_j : i \neq j\}$. Es ist einfach zu sehen, dass $\chi(G) = 2$ und $\Delta(G)+1 = N$. Wenn man die gierige Färbung in der Reihenfolge $x_1, x_2, \ldots, x_N, y_1, y_2, \ldots, y_N$ ausführt, dann braucht man 2 Farben. Wenn man aber die Reihenfolge $x_1, y_1, x_2, y_2, \ldots, x_N, y_N$ wählt, dann braucht die gierige Färbung N Farben.

Bemerkung 2.210. Wenn G ein vollständiger Graph oder ein Kreis ungerader Länge ist, dann gilt $\chi(G) = \Delta(G)+1$.

Satz 2.211 (Brooks)**.** Sei G ein zusammenhängender Graph, der weder ein vollständiger Graph noch ein Kreis ungerader Länge ist. Dann gilt $\chi(G) \leq \Delta(G)$. ⊠

2.13.2 Färbung planarer Graphen

Satz 2.212 (Fünffarbensatz, Heawood)**.** Jeder einfache planare Graph ist mit 5 Farben färbbar.

Beweis. Wir wenden vollständige Induktion bzgl. n an. Für Graphen mit $n \leq 5$ ist die Aussage trivial.

Betrachten wir nun einen planaren Graphen G mit $n \geq 6$ Knoten und nehmen wir an, dass die Aussage schon für Graphen mit weniger als n Knoten bewiesen wurde. Sei v ein Knoten minimaler Gradzahl in G. Wegen Satz 2.173 ist $d(v) \leq 5$.

1. Fall: $d(v) \leq 4$. Entfernen wir v aus G; der resultierende Graph ist auch planar und hat weniger als n Knoten, also ist er laut Induktionsbedingung mit 5 Farben färbbar. Diese Färbung kann leicht zu einer Färbung von G ergänzt werden: da $d(v) \leq 4$, schließen die Nachbarn von v höchstens 4 Farben aus, so dass noch mindestens eine Farbe für v übrigbleibt.

2. Fall: $d(v) = 5$. Wenn die 5 Nachbarn von v alle zueinander adjazent wären, gäbe es ein K_6 in G, was wegen der Planarität von G nicht möglich ist. Also gibt es zwei Nachbarn von v, die nicht adjazent zueinander sind. Seien diese Knoten x und y. Fügen wir die Knoten v, x und y zu einem einzigen Knoten z zusammen. Der resultierende Graph ist auch planar und hat weniger als n Knoten, also ist er laut Induktionsbedingung mit 5 Farben färbbar. Diese Färbung ergänzen wir nun zu einer Färbung von G. Die Knoten x und y sind nicht benachbart, also können sie beide die Farbe von z bekommen. Der Knoten v hat 5 Nachbarn, von denen aber zwei (x und y) dieselbe Farbe haben, also schließen die 5 Nachbarn höchstens 4 Farben aus, so dass noch mindestens eine Farbe für v übrigbleibt. □

Satz 2.213 (Vierfarbensatz, Appel-Haken)**.** Jeder einfache planare Graph ist mit 4 Farben färbbar. ⊠

2.13.3 Perfekte Graphen

Definition 2.214 (induzierter Teilgraph)**.** Der Graph $G' = (V', E')$ ist ein *induzierter Teilgraph* von $G = (V, E)$, wenn $V' \subseteq V$ und für die Knoten $x, y \in V'$ ist $xy \in E'$ genau dann wenn $xy \in E$.

Definition 2.215 (perfekter Graph)**.** Ein Graph G ist *perfekt*, wenn $\omega(G) = \chi(G)$ und für alle induzierten Teilgraphen G' von G gilt auch $\omega(G') = \chi(G')$.

Definition 2.216 (Intervallgraph)**.** Ein Graph $G = (V, E)$ ist ein *Intervallgraph*, wenn man jedem Knoten $v \in V$ ein Intervall $I_v = [a_v, b_v]$ zuordnen kann, so dass $vw \in E$ genau dann wenn $I_v \cap I_w \neq \emptyset$.

Satz 2.217. Sei G ein Intervallgraph mit den Intervallen $I_v = [a_v, b_v]$ wie in Definition 2.216. Führen wir eine gierige Färbung durch, wobei die Knoten in steigender Reihenfolge der a_v Zahlen gefärbt werden. Sei k die Anzahl der vom Algorithmus benötigten Farben. Dann gelten folgende Eigenschaften:
(1) Es gibt eine Clique der Größe k in G, die mit einer einfachen Erweiterung des Algorithmus gefunden werden kann.
(2) $k = \chi(G) = \omega(G)$, d.h. sowohl die Färbung als auch die Clique sind optimal.

Beweis. (1) Sei $v \in V$ ein Knoten, der die k-te Farbe bekommt. Dass v die k-te Farbe bekommt, muss den Grund haben, dass $k-1$ Nachbarn von v schon mit unterschiedlichen Farben gefärbt worden sind. Seien diese Nachbarn $w_1, w_2, \ldots, w_{k-1}$. Da sie schon gefärbt sind, gilt $a_{w_i} \leq a_v$ für jedes $i = 1, 2, \ldots, k-1$, d.h. diese Intervalle fangen früher an als I_v. Da diese Knoten Nachbarn von v sind, folgt daraus, dass $b_{w_i} \geq a_v$ für jedes $i = 1, 2, \ldots, k-1$. Daher ist $a_v \in I_{w_i}$ für jedes $i = 1, 2, \ldots, k-1$. Folglich sind alle w_i auch untereinander benachbart, d.h. v und die w_i Knoten zusammen ergeben eine Clique der Größe k.

(2) Da es eine Clique der Größe k und eine Färbung mit k Farben gibt, gilt $k \leq \omega(G) \leq \chi(G) \leq k$. Es muss hier überall Gleichheit bestehen. □

Korollar 2.218. In einem Intervallgraphen können χ und ω effizient bestimmt werden. □

Korollar 2.219. Intervallgraphen sind perfekt.

Beweis. Sei $G = (V, E)$ ein Intervallgraph mit den Intervallen I_v wie in Definition 2.216. Wegen Satz 2.217 ist $\chi(G) = \omega(G)$. Sei nun $G' = (V', E')$ ein induzierter Teilgraph von G. Dann ist G' eben der zum Intervallsystem $\{I_v : v \in V'\}$ gehörende Intervallgraph. Daher gilt also auch $\chi(G') = \omega(G')$. □

Satz 2.220 (Lovász, Grötschel, Schrijver, 1984)**.** In perfekten Graphen können χ und ω effizient bestimmt werden. ⊠

Satz 2.221 (schwacher Satz über perfekte Graphen, Lovász, 1972)**.** Der Graph G ist genau dann perfekt, wenn sein Komplement $\overline{G}$ perfekt ist. ⊠

Satz 2.222 (starker Satz über perfekte Graphen, 2002)**.** Der Graph G ist genau dann perfekt, wenn weder G noch sein Komplement $\overline{G}$ einen ungeraden Kreis der Länge ≥ 5 als induzierten Teilgraphen enthält. ⊠

Satz 2.223. Es kann in Polynomialzeit entschieden werden, ob ein Graph perfekt ist. ⊠

2.13.4 Kantenfärbung

Definition 2.224. Ähnlich wie bei der Knotenfärbung, nehmen wir einen ungerichteten Graphen G und eine ganze Zahl $k \geq 1$. Eine *Kantenfärbung* von G mit k Farben ordnet jeder Kante von G eine von k Farben zu, so dass Kanten mit einem gemeinsamen Endknoten verschiedene Farben bekommen müssen. G ist mit k Farben *kantenfärbbar*, wenn es eine solche Kantenfärbung gibt. Die Kanten von G, denen bei einer gegebenen Färbung dieselbe Farbe zugeordnet wird, bilden eine *Farbenklasse*.

Bemerkung 2.225. Im Gegensatz zur Knotenfärbung sind bei der Kantenfärbung Schlingen und Parallelkanten erlaubt.

Definition 2.226. Die *kantenchromatische Zahl* (oder *chromatischer Index)* eines Graphen G ist die kleinste Zahl k, so dass G mit k Farben kantenfärbbar ist. Die kantenchromatische Zahl von G wird mit $\chi'(G)$ oder $\chi_e(G)$ bezeichnet.

Bemerkung 2.227. Kein effizienter Algorithmus ist bekannt für die Bestimmung von χ' in einem allgemeinen Graphen.

Satz 2.228. In jedem Graphen G gilt $\chi'(G) \geq \Delta(G)$.

Beweis. Nehmen wir einen Graphen G, und sei v der Knoten mit der höchsten Gradzahl. Zu den Δ Kanten, die zu v inzident sind, müssen verschiedene Farben zugeordnet werden. Daraus folgt, dass man mindestens Δ Farben braucht, um die Kanten von G zu farben. □

Satz 2.229. In jedem Graphen G gilt $\chi'(G) \geq \frac{m}{\nu}$.

Beweis. Betrachten wir eine Kantenfärbung des Graphen mit χ' Farben. Die Anzahl der Kanten in der i. Farbenklasse bezeichnen wir mit k_i. Es ist klar, dass jede Farbenklasse eine unabhängige Kantennmenge ist, und damit gilt $k_i \leq \nu$. Daraus folgt $m = \sum_{i=1}^{\chi'} k_i \leq \chi'\nu$, was eben zur gewünschten Formel führt. □

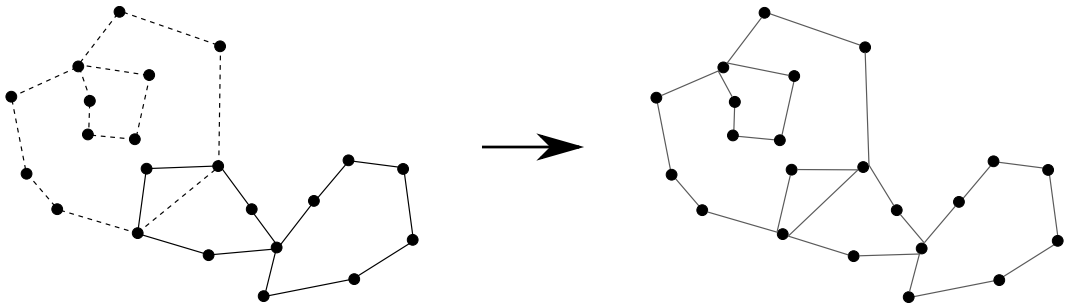

Abbildung 2.15: Algorithmus für das Suchen eines Eulerschen „Kreis"-es

Satz 2.230 (Vizing, 1964)**.** In jedem einfachen Graphen gilt $\chi'(G) \leq \Delta(G) + 1$. ⊠

Satz 2.231 (Kőnig)**.** In jedem bipartiten Graphen gilt $\chi'(G) = \Delta(G)$. ⊠

2.14 Eulersche „Kreise" und „Wege"

Definition 2.232 (Eulerscher „Kreis")**.** In einem ungerichteten – nicht unbedingt einfachen – Graphen G wird eine geschlossene Kantenfolge, die jede Kante des Graphen genau einmal durchläuft, *Eulerscher „Kreis"* genannt.

Bemerkung 2.233. Ein Eulerscher „Kreis" ist nicht unbedingt ein Kreis, weil er einen Knoten auch mehrfach durchlaufen kann.

Satz 2.234 (Euler)**.** Sei G ein Graph ohne isolierte Knoten. G hat genau dann einen Eulerschen „Kreis", falls er zusammenhängend ist und $\forall v \in V(G)\ \ d(v)$ gerade ist.

Beweis. Falls ein Graph G einen Eulerschen „Kreis" hat, ist es trivial, dass G zusammenhängend ist und alle Knoten in G gerade Gradzahl haben.

Falls der Graph zusammenhängend ist und $\forall v \in V(G)\ \ d(v)$ gerade ist, kann man mit dem folgenden Algorithmus einen Eulerschen „Kreis" finden. Man startet in einem beliebigen Knoten v_0. Da es keinen isolierten Knoten gibt, existiert eine Kante $e_1 = v_0v_1$. Da die Gradzahl von v_1 gerade ist, kann man aus v_1 wieder über eine neue, bisher unbesuchte Kante $e_2 = v_1v_2$ weitergehen, usw. Im Allgemeinen, wenn wir gerade in einem Knoten v_i sind, den die Kantenfolge schon k-mal erreicht und $k-1$-mal verlassen hat, dann gibt es außer diesen $2k-1$ Kanten mindestens noch eine weitere zu v_i inzidente Kante, über die die Kantenfolge fortgesetzt werden kann, da die Gradzahl von v_i gerade ist. Dieses Verfahren kann nur dann terminieren, wenn wieder der Anfangsknoten v_0 erreicht wird. Somit wurde eine geschlossene Kantenfolge gefunden, die jede Kante höchstens einmal durchläuft. Wenn es keine weiteren Kanten gibt, ist das ein Eulerscher „Kreis". Sonst – da G zusammenhängend ist – gibt es einen schon besuchten Knoten v_i, zu dem eine bisher noch unbesuchte Kante inzident ist. Dann startet das gleiche Verfahren erneut, diesmal aus v_i. Da die früher schon besuchten und damit nicht mehr nutzbaren Kanten die Gradzahl jedes Knotens um eine gerade Zahl verringern, ist es weiterhin wahr, dass die Kantenfolge aus jedem Knoten weitergeführt werden kann, bis letztendlich der Anfangsknoten wieder erreicht wird. Somit erhält man also eine zweite geschlossene Kantenfolge, die mit der ersten vereinigt werden kann (siehe Abbildung 2.15). Damit ist also eine größere geschlossene Kantenfolge entstanden, die jede Kante höchstens einmal durchläuft. Dieses Verfahren wird wiederholt bis jede Kante in der Kantenfolge enthalten ist und damit ein Eulerscher „Kreis" gefunden wurde. □

Definition 2.235 (Eulerscher „Weg")**.** In einem ungerichteten – nicht unbedingt einfachen – Graphen G wird eine offene Kantenfolge, die jede Kante des Graphen genau einmal durchläuft, *Eulerscher „Weg"* genannt.

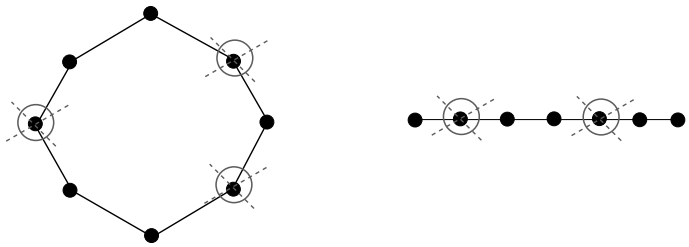

Abbildung 2.16: Notwendige Bedingungen für Hamiltonschen Kreis/Weg

Satz 2.236. Sei G ein Graph ohne isolierte Knoten. G hat genau dann einen Eulerschen „Weg", falls er zusammenhängend ist und folgende Eingenschaft besitzt: es gibt 2 Knoten mit ungerader Gradzahl und $n-2$ Knoten mit gerader Gradzahl.

Beweis. Falls ein Graph G einen Eulerschen „Weg" hat, ist es trivial, dass G zusammenhängend ist, die beiden Endknoten des Eulerschen „Weg"-es ungerade Gradzahl haben und die anderen Gradzahlen gerade sind.

Falls der Graph zusammenhängend ist und alle Knoten außer v_1 und v_2 gerade Gradzahl haben, dann haben alle Knoten in $G' = G \cup \{v_1v_2\}$ gerade Gradzahl, also kann man Satz 2.234 anwenden. Daraus folgt, dass es in G' einen Eulerschen „Kreis" gibt. Ohne v_1v_2 ergibt sich ein Eulerscher „Weg" in G. □

2.15 Hamiltonsche Kreise und Wege

Definition 2.237 (Hamiltonscher Kreis)**.** In einem Graphen G wird ein Kreis, der jeden Knoten des Graphen genau einmal durchläuft, *Hamiltonscher Kreis* genannt.

Definition 2.238 (Hamiltonscher Weg)**.** In einem Graphen G wird ein Weg, der jeden Knoten des Graphen genau einmal durchläuft, *Hamiltonscher Weg* genannt.

Bemerkung 2.239. Im Gegensatz zu Eulerschen „Kreis"-en und „Weg"-en sind Hamiltonsche Kreise und Wege echte Kreise bzw. Wege.

Bemerkung 2.240. Kein effizienter Algorithmus ist bekannt, der entscheiden würde, ob ein allgemeiner Graph einen Hamiltonschen Kreis/Weg enthält. Keine einfach überprüfbare Bedingung ist bekannt, die gleichzeitig notwendig und hinreichend für die Existenz eines Hamiltonschen Kreises/Weges wäre.

Notwendige (aber nicht hinreichende) Bedingung für die Existenz eines Hamiltonschen Kreises/Weges:

Satz 2.241. Wenn ein Graph G einen Hamiltonschen Kreis enthält, ist es für $\forall X \subseteq V(G)$ wahr, dass $c(G\backslash X) \leq |X|$, wobei c die Anzahl der Komponenten bezeichnet.

Beweis. Falls man von einem Kreis k Knoten entfernt, zerfällt der Kreis in k Komponenten (siehe Abbildung 2.16), oder weniger, wenn benachbarte Knoten weggelassen werden. Ein Graph, der einen Hamiltonschen Kreis enthält, ist ein Kreis und eventuell hinzugefügte weitere Kanten. Die Anzahl der Komponenten in $G \setminus X$ kann sich durch diese zusätzliche Kanten nur verringen. □

Satz 2.242. Wenn ein Graph G einen Hamiltonschen Weg enthält, ist es für $\forall X \subseteq V(G)$ wahr, dass $c(G\backslash X) \leq |X| + 1$, wobei c die Anzahl der Komponenten bezeichnet.

Beweis. Falls man von einem Weg k Knoten entfernt, zerfällt der Weg in $k+1$ Komponenten (siehe Abbildung 2.16), oder weniger, wenn benachbarte Knoten weggelassen werden. Ein Graph, der einen Hamiltonschen Weg enthält, ist ein Weg und eventuell hinzugefügte weitere Kanten. Die Anzahl der Komponenten in $G \setminus X$ kann sich durch diese zusätzliche Kanten nur verringen. □

Hinreichende (aber nicht notwendige) Bedingungen für die Existenz eines Hamiltonschen Kreises:

Satz 2.243 (Ore)**.** Sei G ein einfacher Graph, in dem für alle nicht benachbarte Knotenpaare x, y gilt, dass $d(x) + d(y) \geq n$. Dann existiert in G ein Hamiltonscher Kreis.

Beweis. Indirekt, sei G ein Gegenbeispiel mit einer maximalen Anzahl von Kanten. Gegenbeispiel bedeutet, dass die Bedingung bzgl. der Gradzahlen in G erfüllt ist, aber er hat keinen Hamiltonschen Kreis. G hat maximale Anzahl von Kanten, also falls $xy \notin E(G)$, dann ist $G' = G \cup \{xy\}$ kein Gegenbeispiel mehr. Die Bedingung des Satzes ist auch für G' erfüllt, also bedeutet die Tatsache, dass G' kein Gegenbeispiel ist, dass G' einen Hamiltonschen Kreis enthält. Daraus ergibt sich ein Hamiltonscher Weg in G zwischen x und y.

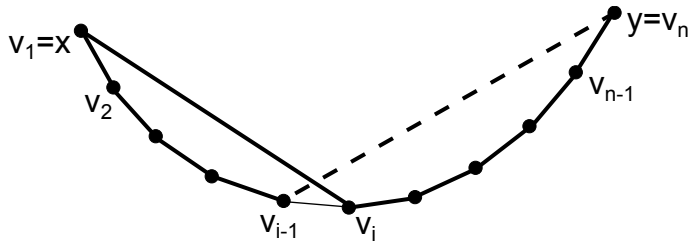

Abbildung 2.17: Zum Beweis vom Satz von Ore

Seien die Knoten entlang dieses Hamiltonschen Weges nummeriert (siehe Abbildung 2.17). Nehmen wir eine Kante $xv_i \in E$. Wie man in der Abbildung sieht, muss $yv_{i-1} \notin E$ gelten, sonst gäbe es einen Hamiltonschen Kreis in G. Somit werden aus den $n-1$ möglichen Nachbarn von y (y ist mit sich selbst nicht benachbart) $d(x)$ ausgeschlossen. Daraus folgt $d(y) \leq n-1-d(x)$, oder umgeformt: $d(x)+d(y) \leq n-1$. Dies widerspricht der Bedingung des Satzes. □

Satz 2.244 (Dirac)**.** Sei G ein einfacher Graph, in dem für $\forall v \in V(G)$ gilt, dass $d(v) \geq \frac{n}{2}$. Dann existiert in G ein Hamiltonscher Kreis.

Beweis. Folgt aus dem Satz von Ore. □

Bemerkung 2.245. Bei Eulerschen „Kreis"-en/„Weg"-en sind Schlingen und Parallelkanten erlaubt. Bei Hamiltonschen Kreisen/Wegen spielen Schlingen und Parallelkanten keine Rolle; in den Sätzen von Ore und Dirac ist es auch wichtig, dass es sich um einfache Graphen handelt.

Kapitel 3

Komplexitätstheorie (informeller Aufbau)

Definition 3.1 (Entscheidungsproblem)**.** Ein *Entscheidungsproblem* hat einen gewissen Input, und die Aufgabe ist zu entscheiden, ob der Input eine gewisse Eigenschaft erfüllt. D.h. ein Entscheidungsproblem ist ein spezielles Problem, in dem der Output nur zwei mögliche Werte hat: JA oder NEIN.

Bemerkung 3.2. Andere Arten von Problemen, beispielsweise Optimierungsprobleme, können auch einfach auf Entscheidungsprobleme zurückgeführt werden. Z.B. ist es ein Optimierungsproblem, in einem Graphen den längsten Weg zu finden (Input: der Graph). Das zugehörige Entscheidungsproblem ist die Entscheidung ob es im Graphen einen mindestens k langen Weg gibt (Input: der Graph sowie k).

Definition 3.3 (Polynomialzeit)**.** Ein Algorithmus arbeitet in *Polynomialzeit*, wenn es Konstanten a und b gibt, so dass die Anzahl der Schritte, die der Algorithmus für Inputs der Länge n ausführt, höchstens an^b beträgt.

Definition 3.4 ($\mathcal{P}$)**.** $\mathcal{P}$ bezeichnet die Menge jener Entscheidungsprobleme, die durch einen Algorithmus in Polynomialzeit gelöst werden können.

Definition 3.5 ($\mathcal{NP}$)**.** $\mathcal{NP}$ bezeichnet die Menge jener Entscheidungsprobleme, bei denen für Inputs, für die die Antwort JA ist, diese positive Antwort mit Hilfe geeigneter Zusatzinformation (*Zeuge* oder *Beweis* genannt) in Polynomialzeit verifiziert werden kann. Dabei muss die Größe der Zusatzinformation auch durch ein Polynom des Inputs beschränkt sein.

Definition 3.6 ($co\mathcal{NP}$)**.** $co\mathcal{NP}$ bezeichnet die Menge jener Entscheidungsprobleme, bei denen für Inputs, für die die Antwort NEIN ist, diese negative Antwort mit Hilfe geeigneter Zusatzinformation (*Zeuge* oder *Beweis* genannt) in Polynomialzeit verifiziert werden kann. Dabei muss die Größe der Zusatzinformation auch durch ein Polynom des Inputs beschränkt sein.

Bemerkung 3.7. Z.B. das Problem, ob ein Graph einen Eulerschen „Kreis" enthält, ist sowohl in $\mathcal{NP}$ als auch in $co\mathcal{NP}$. Ein möglicher Zeuge für die positive Antwort ist die Reihenfolge der Kanten in einem Eulerschen „Kreis" des Graphen. Ein möglicher Zeuge für die negative Antwort ist ein Knoten ungeraden Grades.

Bemerkung 3.8. Z.B. das Problem, ob ein Graph einen Hamiltonschen Kreis enthält, ist in $\mathcal{NP}$. Ein möglicher Zeuge für die positive Antwort ist die Reihenfolge der Knoten in einem Hamiltonschen Kreis des Graphen. Für die negative Antwort ist kein geeigneter Zeuge bekannt; daher ist es unbekannt, ob das Problem in $co\mathcal{NP}$ liegt.

Satz 3.9. $\mathcal{P} \subseteq \mathcal{NP} \cap co\mathcal{NP}$.

Bemerkung 3.10. Die wichtigsten offenen Punkte in der Komplexitätstheorie sind:

- Ist $\mathcal{P} = \mathcal{NP}$? Vermutung: Nein.
- Ist $\mathcal{P} = \mathcal{NP} \cap co\mathcal{NP}$? Vermutung: Ja.

Definition 3.11 (Cook-Reduktion)**.** Seien P_1 und P_2 zwei Entscheidungsprobleme. P_1 ist *zurückführbar* auf P_2 (Notation: $P_1 \prec P_2$), wenn es einen Algorithmus A gibt, so dass die Anzahl der Schritte von A durch ein Polynom der Länge des Inputs beschränkt ist und jeder Schritt von A entweder ein elementarer Schritt oder die Lösung einer Instanz des P_2 Problems ist, wobei die Größe dieser P_2-Instanzen durch ein Polynom der Länge des ursprünglichen Inputs beschränkt sein muss.

Definition 3.12 (Karp-Reduktion)**.** Seien P_1 und P_2 zwei Entscheidungsprobleme. P_1 ist *zurückführbar* auf P_2 (Notation: $P_1 \prec_{Karp} P_2$), wenn es eine Funktion f gibt, so dass f in Polynomialzeit berechnet werden kann und der Output des Problems P_1 beim Input x genau dann JA ist, wenn der Output des Problems P_2 beim Input $f(x)$ JA ist.

Bemerkung 3.13. Die Karp-Reduktion ist eine spezielle Cook-Reduktion, in der genau eine Instanz des P_2 Problems gelöst wird und das findet am Ende des Algorithmus statt, so dass der Output dieses Aufrufs gleich als Output des Algorithmus genutzt wird. Es wird vermutet, dass aus $P_1 \prec P_2$ auch $P_1 \prec_{Karp} P_2$ folgt (mit Ausnahme einiger trivialer Fälle).

Satz 3.14. $P_1 \prec P_2, P_2 \in \mathcal{P} \Rightarrow P_1 \in \mathcal{P}$.

Satz 3.15. $P_1 \prec_{Karp} P_2, P_2 \in \mathcal{NP} \Rightarrow P_1 \in \mathcal{NP}$.

Satz 3.16. $P_1 \prec P_2, P_2 \prec P_3 \Rightarrow P_1 \prec P_3$.

Definition 3.17 ($\mathcal{NP}$-vollständig, $\mathcal{NP}$-schwer, $\mathcal{NPC}$)**.** Das Problem P_0 ist *$\mathcal{NP}$-schwer*, wenn alle Probleme in $\mathcal{NP}$ auf P_0 zurückgeführt werden können. Das Entscheidungsproblem P_0 ist *$\mathcal{NP}$-vollständig*, wenn es in $\mathcal{NP}$ liegt und $\mathcal{NP}$-schwer ist. Die Menge der $\mathcal{NP}$-vollständigen Probleme wird mit $\mathcal{NPC}$ bezeichnet.

Satz 3.18. Wenn $\mathcal{P} \neq \mathcal{NP}$, dann $\mathcal{P} \cap \mathcal{NPC} = \emptyset$.

Bemerkung 3.19. Folglich, wenn man *ein* $\mathcal{NP}$-vollständiges Problem in Polynomialzeit lösen kann, dann kann man *alle* Probleme in $\mathcal{NP}$ in Polynomialzeit lösen.

Satz 3.20 (Cook–Levin)**.** Es gibt $\mathcal{NP}$-vollständige Probleme.

Satz 3.21 (Ladner)**.** Wenn $\mathcal{P} \neq \mathcal{NP}$, dann $\mathcal{NP} \setminus (\mathcal{P} \cup \mathcal{NPC}) \neq \emptyset$.

Satz 3.22. Wenn $P_0 \in \mathcal{NP}$ und $\exists P_1 \in \mathcal{NPC}$ so dass $P_1 \prec P_0$, dann gilt $P_0 \in \mathcal{NPC}$.

Bemerkung 3.23. Das ist die Art und Weise, wie man relativ einfach beweisen kann, dass ein Problem $\mathcal{NP}$-vollständig ist.

Satz 3.24. Die folgenden Probleme sind $\mathcal{NP}$-vollständig:

- Input: (Gerichteter) Graph G. Frage: Hat G einen (gerichteten) Hamiltonschen Weg?
- Input: (Gerichteter) Graph G. Frage: Hat G einen (gerichteten) Hamiltonschen Kreis?
- Input: (Gerichteter) Graph G und eine Nummer k. Frage: Hat G einen mindestens k langen (gerichteten) Weg?
- Input: (Gerichteter) Graph G und eine Nummer k. Frage: Hat G einen mindestens k langen (gerichteten) Kreis?
- Input: Zwei Graphen G und H. Frage: Hat G einen Teilgraphen G', der zu H isomorph ist?
- Input: Graph G und eine Nummer k. Frage: Kann G mit k Farben gefärbt werden?

- Sei $k \geq 3$ gegeben. Input: Graph G. Frage: Kann G mit k Farben gefärbt werden?
- Input: Graph G. Frage: Können die Kanten von G mit $\Delta(G)$ Farben gefärbt werden?
- Input: Graph G und eine Nummer k. Frage: Enthält G mindestens k unabhängige Knoten?
- Input: Graph G und eine Nummer k. Frage: Enthält G eine Menge von k Knoten, die alle Kanten überdecken?
- Input: Graph G und eine Nummer k. Frage: Enthält G eine Clique der Größe mindestens k?
- Input: Netzwerk (G, c, s, t) und eine Nummer k. Frage: Gibt es einen $s\bar{t}$-Schnitt mit Kapazität mindestens k?
- Input: Aussagenlogische Formel (bestehend aus Booleschen Variablen, Negation, Disjunktion, Konjunktion) in konjunktiver Normalform. Frage: Ist die Formel erfüllbar?
- Sei $k \geq 3$ gegeben. Input: Aussagenlogische Formel (bestehend aus Booleschen Variablen, Negation, Disjunktion, Konjunktion) in konjunktiver Normalform, wobei jeder Term aus höchstens k Literalen besteht. Frage: Ist die Formel erfüllbar?

Satz 3.25. Die folgenden Probleme sind in $\mathcal{P}$:

- Input: (Gerichteter) Graph G. Frage: Hat G einen (gerichteten) Eulerschen „Weg"?
- Input: (Gerichteter) Graph G. Frage: Hat G einen (gerichteten) Eulerschen „Kreis"?
- Input: (Gerichteter) Graph G, zwei ausgewählte Knoten (s, t) und eine Nummer k. Frage: Gibt es in G einen höchstens k langen (gerichteten) Weg von s zu t?
- Input: (Gerichteter) Graph G und eine Nummer k. Frage: Enthält G einen höchstens k langen (gerichteten) Kreis?
- Input: Graph G. Frage: Kann G mit 2 Farben gefärbt werden?
- Input: Graph G und eine Nummer k. Frage: Enthält G mindestens k unabhängige Kanten?
- Input: Graph G und eine Nummer k. Frage: Enthält G eine Menge von k Kanten, die alle Knoten überdecken?
- Input: Netzwerk (G, c, s, t) und eine Nummer k. Frage: Gibt es einen $s\bar{t}$-Schnitt mit Kapazität höchstens k?
- Input: Aussagenlogische Formel (bestehend aus Booleschen Variablen, Negation, Disjunktion, Konjunktion) in konjunktiver Normalform, wobei jeder Term aus höchstens 2 Literalen besteht. Frage: Ist die Formel erfüllbar?

Bemerkung 3.26. Das folgende Problem ist in $\mathcal{NP}$, aber es ist nicht bekannt, dass es in $\mathcal{P}$ oder in $\mathcal{NPC}$ wäre:
Input: Zwei Graphen G und H; Frage: Sind G und H isomorph?

Bemerkung 3.27. Für die möglichst effiziente Lösung von $\mathcal{NP}$-vollständigen Problemen gibt es eine Reihe von Techniken:

- Spezialfälle von $\mathcal{NP}$-vollständigen Problemen können manchmal in Polynomialzeit gelöst werden.
- Wenn die in der Praxis vorkommenden Eingaben nicht allzu groß sind, können auch Algorithmen mit exponentieller Laufzeit geeignet sein.
- Reduktion der Laufzeit eines exponentiellen Algorithmus, z.B. von $O(2^n)$ zu $O\left(\left(\frac{4}{3}\right)^n\right)$

- Approximationsalgorithmen
- Heuristiken
- Randomisierte Algorithmen, die in Polynomialzeit arbeiten und mit hinreichend hoher Wahrscheinlichkeit das korrekte Ergebnis liefern

Kapitel 4

Zahlentheorie

4.1 Grundbegriffe

Satz 4.1. Seien $a, b \in \mathbb{Z}^+$. $a + b$ und $a - b$ können in linearer Zeit berechnet werden. ab und $[a/b]$ können in Polynomialzeit berechnet werden. a^b kann nur in exponentiell vielen Schritten berechnet werden.

Definition 4.2 (Teiler)**.** $a \in \mathbb{Z}$ ist ein *Teiler* von $b \in \mathbb{Z}$ (Notation: $a|b$), wenn $\exists c \in \mathbb{Z}$ so dass $ac = b$.

Definition 4.3 (ggT)**.** Für $a, b \in \mathbb{Z}^+$ ist $c \in \mathbb{Z}^+$ der *größte gemeinsame Teiler* (Notation: $ggT(a, b)$), wenn $c|a$, $c|b$ und für jedes $d \in \mathbb{Z}^+$ mit $d|a$ und $d|b$ gilt, dass $d|c$.

Definition 4.4 (kgV)**.** Für $a, b \in \mathbb{Z}^+$ ist $c \in \mathbb{Z}^+$ das *kleinste gemeinsame Vielfache* (Notation: $kgV(a, b)$), wenn $a|c$, $b|c$ und für jedes $d \in \mathbb{Z}^+$ mit $a|d$ und $b|d$ gilt, dass $c|d$.

Definition 4.5 (irreduzibel)**.** $a \in \mathbb{Z}$, $a > 1$ ist *irreduzibel*, wenn es keine positiven Teiler außer 1 und a hat.

Definition 4.6 (Primzahl)**.** $a \in \mathbb{Z}$, $a > 1$ ist eine *Primzahl*, wenn aus $a|bc$ entweder $a|b$ oder $a|c$ folgt. (Inklusives Oder)

Definition 4.7 (teilerfremd)**.** $a, b \in \mathbb{Z}^+$ sind *teilerfremd*, wenn $ggT(a, b) = 1$.

Satz 4.8 (Division mit Rest)**.** Für alle $a, b \in \mathbb{Z}^+$ existieren $q, r \in \mathbb{N}$ eindeutig so dass $a = qb + r$ und $0 \leq r \leq b - 1$. q, r können in Polynomialzeit bestimmt werden.

Lemma 4.9. Wenn $a = qb + r$, dann gilt $ggT(a, b) = ggT(b, r)$

Algorithmus 4.10 (Euklidischer Algorithmus).
`Input:`
$a, b \in \mathbb{Z}^+$, $a > b$.
`Output:`
$ggT(a, b)$.
`Ablauf:`

$$\begin{array}{rcrcrcl}
a & = & q_1 b & + & r_1 & , & 0 < r_1 < b \\
b & = & q_2 r_1 & + & r_2 & , & 0 < r_2 < r_1 \\
r_1 & = & q_3 r_2 & + & r_3 & , & 0 < r_3 < r_2 \\
 & \cdots & & & & & \\
r_{n-1} & = & q_{n+1} r_n & + & r_{n+1} & , & 0 < r_{n+1} < r_n \\
r_n & = & q_{n+2} r_{n+1} & + & 0 & &
\end{array}$$

$ggT(a, b) = r_{n+1}$.

Satz 4.11. Der euklidische Algorithmus terminiert in Polinomzeit und liefert den größten gemeinsamen Teiler.

Satz 4.12 (Bézout)**.** (1) Für alle $a, b \in \mathbb{Z}^+$ existieren $x, y \in \mathbb{Z}$ so dass $ggT(a, b) = ax + by$.
(2) Seien $a, b, c \in \mathbb{Z}^+$ gegeben. Es existiert ein Paar $x, y \in \mathbb{Z}$ mit $c = ax+by$ genau dann wenn $ggT(a, b)|c$.
(3) Seien $a, b \in \mathbb{Z}^+$ gegeben. Es existiert zu jedem $c \in \mathbb{Z}$ ein Paar $x, y \in \mathbb{Z}$ mit $c = ax + by$ genau dann wenn $ggT(a, b) = 1$.

Lemma 4.13. Sei $a|bc$, $ggT(a, b) = 1$. Dann gilt $a|c$.

Satz 4.14. $a \in \mathbb{Z}^+$ ist irreduzibel genau dann wenn es eine Primzahl ist.

Satz 4.15 (Fundamentalsatz der Arithmetik)**.** Jede ganze Zahl ≥ 2 kann als das Produkt von Primzahlen aufgeschrieben werden. Welche Primzahlen darin vorkommen und wie oft eine gegebene Primzahl darin vorkommt ist eindeutig, die Reihenfolge der Primzahlen nicht.

Definition 4.16 (kanonische Form, Primfaktorzerlegung)**.** $n \in \mathbb{Z}, n \geq 2 \Rightarrow n = p_1^{\alpha_1} \cdot \ldots \cdot p_k^{\alpha_k}$, wobei $p_1, \ldots, p_k$ verschiedene Primzahlen sind und $\alpha_i > 0$ für jedes $i = 1, \ldots, k$.

Bemerkung 4.17. Wir kennen keinen effizienten Algorithmus für die Bestimmung der Primfaktoren einer Zahl.

Definition 4.18. Sei $n \in \mathbb{Z}^+$. $d(n)$ bezeichnet die Anzahl der positiven Teiler von n. $\sigma(n)$ bezeichnet die Summe der positiven Teiler von n.

Satz 4.19. Sei die kanonische Form von a: $a = p_1^{\alpha_1} \cdot \ldots \cdot p_k^{\alpha_k}$. Sei $b|a$. Dann ist $b = p_1^{\beta_1} \cdot \ldots \cdot p_k^{\beta_k}$, wobei $0 \leq \beta_i \leq \alpha_i$ für jedes $i = 1, \ldots, k$.

Satz 4.20. Sei die kanonische Form von n: $n = p_1^{\alpha_1} \cdot \ldots \cdot p_k^{\alpha_k}$. Dann ist $d(n) = (\alpha_1 + 1) \cdot \ldots \cdot (\alpha_k + 1)$.

Satz 4.21. Seien $a = p_1^{\alpha_1} \cdot \ldots \cdot p_k^{\alpha_k}$ und $b = p_1^{\beta_1} \cdot \ldots \cdot p_k^{\beta_k}$, wobei einige α_is oder β_is auch 0 sein können. Dann ist $ggT(a, b) = p_1^{\min(\alpha_1,\beta_1)} \cdot \ldots \cdot p_k^{\min(\alpha_k,\beta_k)}$, $kgV(a, b) = p_1^{\max(\alpha_1,\beta_1)} \cdot \ldots \cdot p_k^{\max(\alpha_k,\beta_k)}$.

4.2 Sätze und Vermutungen über Primzahlen

Satz 4.22 (Euklid)**.** Es gibt unendlich viele Primzahlen.

Satz 4.23 (Dirichlet)**.** Wenn $ggT(a, b) = 1$, dann enthält die Zahlenfolge $a, a + b, a + 2b, a + 3b, \ldots$ unendlich viele Primzahlen.

Satz 4.24 (Primzahlsatz)**.** Sei $\pi(x)$ die Anzahl der Primzahlen von 2 bis x. Dann gilt: $\pi(x) \approx \frac{x}{\ln x}$.

Satz 4.25. Es gibt beliebig große Lücken zwischen nacheinanderfolgenden Primzahlen.

Definition 4.26 (Primzahlenzwillinge)**.** p und $p + 2$ sind *Primzahlenzwillinge*, wenn sie beide Primzahlen sind.

Bemerkung 4.27. Es ist unklar, ob es unendlich viele Primzahlenzwillinge gibt.

Bemerkung 4.28. Goldbachsche Vermutung: Jede gerade Zahl ≥ 2 kann als Summe zweier Primzahlen aufgeschrieben werden. Schwächere Form: Jede ungerade Zahl ≥ 7 kann als Summe dreier Primzahlen aufgeschrieben werden.

4.3 Kongruenz

Definition 4.29 (kongruent)**.** Seien $a, b, m \in \mathbb{Z}$, $m > 1$. a und b sind *kongruent modulo* m (Notation: $a \equiv b \pmod{m}$), wenn $m|a - b$.

Satz 4.30. Sei $m \in \mathbb{Z}$, $m > 1$ gegeben. Dann ist die Kongruenz modulo m eine Äquivalenzrelation.

Definition 4.31 (Restklasse)**.** Sei $m \in \mathbb{Z}$, $m > 1$ gegeben. Die Äquivalenzklassen der Kongruenzrelation sind die *Restklassen modulo* m.

Satz 4.32. Wenn $a \equiv b \pmod m$ und $c \equiv d \pmod m$, dann gilt: $a + c \equiv b + d \pmod m$, $a - c \equiv b - d \pmod m$ und $ac \equiv bd \pmod m$.

Satz 4.33. Sei $ac \equiv bc \pmod m$, $d = ggT(c, m)$. Dann gilt: $a \equiv b \pmod{\frac{m}{d}}$.

Satz 4.34. $ax \equiv b \pmod m$ ist lösbar genau dann wenn $ggT(a, m)|b$. In diesem Fall bilden die Lösungen eine Restklasse modulo $\frac{m}{ggT(a,m)}$. Wenn das x_0 ist, dann sind die Lösungen modulo m: $x_0, x_0 + \frac{m}{ggT(a,m)}, x_0 + 2\frac{m}{ggT(a,m)}, \ldots, x_0 + (ggT(a, m) - 1)\frac{m}{ggT(a,m)}$, also $ggT(a, m)$ Restklassen.

Definition 4.35 (lineare diophantische Gleichung)**.** Seien $a, b, c \in \mathbb{Z}$ gegeben. Die Gleichung $ax + by = c$, in der die Unbekannten x, y ganze Zahlen sein müssen, ist eine *lineare diophantische Gleichung*.

Satz 4.36. Die lineare diophantische Gleichung $ax + by = c$ ist lösbar genau dann wenn $ggT(a, b)|c$.

Satz 4.37. Bei der linearen diophantischen Gleichung $ax + by = c$ sei $ggT(a, b)|c$. Die Lösung der Kongruenz $ax \equiv c \pmod b$ sei $x \equiv x_0 \pmod{b'}$, wobei $b' = \frac{b}{ggT(a,b)}$. Dann sind die Lösungen der diophantischen Gleichung: $\{(tb' + x_0, -ta' + \frac{c-ax_0}{b}) : t \in \mathbb{Z}\}$, wobei $a' = \frac{a}{ggT(a,b)}$.

Satz 4.38. Wenn p eine Primzahl ist, dann folgt aus $x^2 \equiv 1 \pmod p$ entweder $x \equiv 1 \pmod p$ oder $x \equiv -1 \pmod p$.

Satz 4.39 (Wilson)**.**

$$(m-1)! \equiv \begin{cases} -1 \pmod m & \text{wenn } m \text{ eine Primzahl ist} \\ 0 \pmod m & \text{wenn } m \text{ eine zusammengesetzte Zahl} > 4 \text{ ist} \\ 2 \pmod m & \text{wenn } m = 4 \end{cases}$$

Satz 4.40. Sei $a \equiv b \pmod m$. Dann ist $ggT(a, m) = 1$ genau dann wenn $ggT(b, m) = 1$. D.h. entweder sind alle Elemente einer Restklasse teilerfremd zu m oder sie sind alle nicht teilerfremd zu m.

Definition 4.41 (Eulersche φ-Funktion)**.** Für $m \in \mathbb{Z}$, $m > 1$ bezeichne $\varphi(m)$ die Anzahl ganzer Zahlen in [1,m], die zu m teilerfremd sind. Äquivalent: $\varphi(m)$ ist die Anzahl der zu m teilerfremden Restklassen.

Satz 4.42. Wenn a und b zu m teilerfremd sind, dann ist auch ab teilerfremd zu m.

Satz 4.43 (Euler-Fermat)**.** Seien $a, m \in \mathbb{Z}$, $m > 1$, $ggT(a, m) = 1$. Dann gilt: $a^{\varphi(m)} \equiv 1 \pmod m$.

Satz 4.44 (kleiner Satz von Fermat)**.** Sei p eine Primzahl, $a \in \mathbb{Z}$ beliebig. Dann gilt: $a^p \equiv a \pmod p$.

Satz 4.45 (Multiplikativität von φ)**.** Wenn $ggT(a, b) = 1$, dann ist $\varphi(ab) = \varphi(a)\varphi(b)$.

Satz 4.46. Sei die kanonische Form von n: $n = p_1^{\alpha_1} \cdot \ldots \cdot p_k^{\alpha_k}$. Dann ist $\varphi(n) = (p_1^{\alpha_1} - p_1^{\alpha_1 - 1}) \cdot \ldots \cdot (p_k^{\alpha_k} - p_k^{\alpha_k - 1}) = n \cdot (1 - \frac{1}{p_1}) \cdot \ldots \cdot (1 - \frac{1}{p_k})$.

Satz 4.47. Wenn $ggT(a, m) = 1$, dann ist die Lösung von $ax \equiv b \pmod m$ die Restklasse $x \equiv a^{\varphi(m)-1}b \pmod m$.

Satz 4.48. Die Grundrechenarten können auch modulo m in Polynomialzeit berechnet werden.

Algorithmus 4.49 (Binäre Modulo-Exponentiation).
`Input:`
$a, b, m \in \mathbb{Z}^+$, $m > 1$.
`Output:`
$a^b \pmod m$.

`Ablauf`:
(1) Sei $K = \lfloor \log_2 b \rfloor$.
(2) Man bestimmt die Binärdarstellung von b. Seien die Positionen, in der eine 1 steht: $0 \leq i_1 < i_2 < \ldots < i_k \leq K$. (Die Position 0 ist das LSB, die Position K das MSB.)
(3) Man bestimmt durch sukzessives Quadrieren und Restbildung $a^2 \pmod m, a^4 \pmod m, a^8 \pmod m, \ldots, a^{2^K} \pmod m$.
(4) Man multipliziert $a^{2^{i_1}}, a^{2^{i_2}}, \ldots, a^{2^{i_k}}$ modulo m.

Satz 4.50. Die binäre Modulo-Exponentiation liefert $a^b \pmod m$ in Polynomialzeit.

4.4 Primzahltest-Verfahren

Algorithmus 4.51 (Probedivision).
`Input`:
$n \in \mathbb{Z}$, $n > 1$.
`Output`:
Ob n eine Primzahl oder eine zusammengesetzte Zahl ist.
`Ablauf`:
Überprüfe nacheinander, ob n mit $2, 3, 4, , \ldots, \sqrt{n}$ teilbar ist. Wenn es mit keiner dieser Zahlen teilbar ist, dann ist es eine Primzahl, sonst eine zusammengesetzte Zahl.

Algorithmus 4.52 (Sieb des Eratosthenes).
`Input`:
$N \in \mathbb{Z}$, $N > 1$.
`Output`:
Liste der Primzahlen in $[2, N]$.
`Ablauf`:
(1) Man schreibt alle ganze Zahlen von 2 bis N auf.
(2) Solange es noch unmarkierte Zahlen gibt, wiederhole:
  (2.1) Markiere die kleinste unmarkierte Zahl als Primzahl.
  (2.1) Markiere alle Vielfachen dieser Zahl als zusammengesetzt.

Algorithmus 4.53 (Fermatscher Primzahltest).
`Input`:
$n, a \in \mathbb{Z}$, $n \geq 5$, $1 < a < n - 1$.
`Output`:
Indikation darüber ob n eine Primzahl ist.
`Ablauf`:
Überprüfe, ob $a^{n-1} \equiv 1 \pmod n$ gilt. Wenn ja, ist das Ergebnis: „n kann eine Primzahl sein". Sonst ist das Ergebnis: „n ist keine Primzahl".

Definition 4.54 (Carmichael-Zahl)**.** $n \in \mathbb{Z}^+$ ist eine *Carmichael-Zahl*, wenn $n = p_1 \cdot \ldots \cdot p_k$, wobei $k \geq 3$, die p_is verschiedene Primzahlen sind und $p_i - 1 | n - 1$ für jedes $1 \leq i \leq k$.

Satz 4.55. Wenn das Ergebnis des Fermatschen Primzahltests „n ist keine Primzahl" lautet, dann ist n wirklich keine Primzahl. Wenn n keine Carmichael-Zahl ist, a zufällig gewählt wird und das Ergebnis des Fermatschen Primzahltests „n kann eine Primzahl sein" lautet, dann ist die Wahrscheinlichkeit, dass n wirklich eine Primzahl ist, mindestens 50%.

Algorithmus 4.56 (Miller-Rabin-Test).
`Input`:
$n, a \in \mathbb{Z}$, $n \geq 5$, $1 < a < n - 1$.
`Output`:
Indikation darüber ob n eine Primzahl ist.

Ablauf:
(1) Sei $n-1 = d \cdot 2^s$, wobei d ungerade ist.
(2) Man berechnet $a^d \pmod n$.
(3) Man berechnet durch sukzessives Quadrieren und Restbildung $a^{d \cdot 2^r} \pmod n$, $r = 1, 2, \ldots, s$.
(4) Das Ergebnis ist: „n kann eine Primzahl sein", wenn $a^{d \cdot 2^s} \equiv 1 \pmod n$ und entweder $a^d \equiv \pm 1 \pmod n$ oder $a^{d \cdot 2^r} \equiv -1 \pmod n$ für ein $1 \leq r \leq s-1$. Sonst ist das Ergebnis: „n ist keine Primzahl".

Satz 4.57. Wenn das Ergebnis des Miller-Rabin-Tests „n ist keine Primzahl" lautet, dann ist n wirklich keine Primzahl. Wenn a zufällig gewählt wird und das Ergebnis des Miller-Rabin-Tests „n kann eine Primzahl sein" lautet, dann ist die Wahrscheinlichkeit, dass n wirklich eine Primzahl ist, mindestens 75%.

Bemerkung 4.58. Der so genannte AKS-Test entscheidet in polynomieller Zeit mit vollständiger Sicherheit, ob eine gegebene Zahl eine Primzahl ist. Er ist jedoch zu kompliziert für die praktische Anwendung.

4.5 Kryptographie

Algorithmus 4.59 (RSA-Algorithmus: Vorbereitung).
Input:
–.
Output:
Privater und öffentlicher Schlüssel.
Ablauf:
(1) Seien p und q zwei große Primzahlen, $m = pq$, $\varphi(m) = (p-1)(q-1)$.
(2) Sei e eine Zahl mit $ggT(e, \varphi(m)) = 1$.
(3) Die Lösung von $ex \equiv 1 \pmod{\varphi(m)}$ sei d.
(4) Der öffentliche Schlüssel ist (m, e), der private Schlüssel: $(p, q, \varphi(m), d)$.

Algorithmus 4.60 (RSA-Algorithmus: Verschlüsselung).
Input:
Zu verschlüsselnde Nachricht $x \pmod m$.
Output:
Verschlüsselte Nachricht y.
Ablauf:
Sei $y = x^e \pmod m$

Algorithmus 4.61 (RSA-Algorithmus: Entschlüsselung).
Input:
Verschlüsselte Nachricht $x^e \pmod m$.
Output:
Unverschlüsselte Nachricht x.
Ablauf:
$(x^e)^d \equiv x \pmod m$

Algorithmus 4.62 (Diffie-Hellman-Schlüsselaustausch).
Input:
Eine Primzahl p und eine Zahl $g > 1$.
Output:
Gemeinsamer geheimer Schlüssel.
Ablauf:
(1) Alice wählt eine Zahl a und schickt $g^a \pmod p$ an Bob.
(2) Bob wählt eine Zahl b und schickt $g^b \pmod p$ an Alice.
(3) Alice berechnet $K = \left(g^b\right)^a \equiv g^{ab} \pmod p$.
(4) Bob berechnet $K = \left(g^a\right)^b \equiv g^{ab} \pmod p$.

Kapitel 5

Abstrakte Algebra

Definition 5.1 (algebraische Struktur, Operation)**.** Eine *algebraische Struktur* ist eine Menge M mit einer oder mehreren Operationen. Eine *Operation* über M mit k Argumenten ordnet zu jedem Tupel von k Elementen aus M ein Element aus M zu, d.h. die Operationen führen nicht aus M hinaus.

Definition 5.2 (Assoziativität)**.** Die Operation $*$ mit zwei Argumenten ist *assoziativ*, wenn $(a * b) * c = a * (b * c)$ für alle mögliche a, b, c gilt.

Definition 5.3 (Kommutativität)**.** Die Operation $*$ mit zwei Argumenten ist *kommutativ*, wenn $a * b = b * a$ für alle mögliche a, b gilt.

5.1 Gruppentheorie

5.1.1 Grundbegriffe

Definition 5.4 (Halbgruppe)**.** $(M, *)$ ist eine *Halbgruppe*, wenn $*$ eine assoziative Operation über M mit zwei Argumenten ist.

Definition 5.5 (neutrales Element)**.** Sei $(M, *)$ eine Halbgruppe. $e \in M$ ist ein *neutrales Element*, wenn $a * e = e * a = a$ für alle $a \in M$.

Satz 5.6. Seien e_1 und e_2 neutrale Elemente in einer Halbgruppe. Dann ist $e_1 = e_2$.

Definition 5.7 (Inverse)**.** Sei $(M, *)$ eine Halbgruppe, in der es ein neutrales Element e gibt. Für $a \in M$ ist $a' \in M$ eine *Inverse*, wenn $a * a' = a' * a = e$.

Satz 5.8. Sei $(M, *)$ eine Halbgruppe, in der es ein neutrales Element gibt und seien a' und a'' Inversen von a. Dann gilt $a' = a''$.

Bemerkung 5.9. Gemäß Satz 5.8 ist die Inverse von a eindeutig durch a gegeben. Daher ist es berechtigt, die Inverse von a mit a^{-1} zu bezeichnen.

Definition 5.10 (Gruppe)**.** Die Halbgruppe $(M, *)$ ist eine *Gruppe*, wenn es ein neutrales Element gibt und jedes Element eine Inverse hat.

Definition 5.11 (Abelsche Gruppe)**.** Die Gruppe $(M, *)$ ist eine *kommutative Gruppe* oder *Abelsche Gruppe*, wenn $*$ kommutativ ist.

Definition 5.12 (Ordnung einer Gruppe)**.** Die *Ordnung* der Gruppe $(M, *)$ ist $|M|$.

Definition 5.13 (Diedergruppe)**.** Sei $n \in \mathbb{Z}$, $n \geq 3$ und man betrachte ein regelmäßiges n-Eck in der Ebene. Die *Diedergruppe* D_n besteht aus den Symmetrien der Ebene, die das n-Eck auf sich selbst abbilden. Auf dieser Menge ist eine Operation definiert: die Komposition von Symmetrien.

Definition 5.14 (Symmetrische Gruppe)**.** Sei $n \in \mathbb{Z}^+$ und man betrachte eine n-elementige Menge M. Die *symmetrische Gruppe* S_n besteht aus allen Permutationen von M. Auf S_n ist eine Operation definiert: die Komposition von Permutationen.

Definition 5.15 ($\mathbb{Z}_m$, $\mathbb{Z}_m^*$, $m\mathbb{Z}$)**.** Sei $m \in \mathbb{Z}$, $m > 1$. $\mathbb{Z}_m$ bezeichnet die Menge der Restklassen modulo m. $\mathbb{Z}_m^*$ bezeichnet die Menge der zu m teilerfremden Restklassen modulo m. $m\mathbb{Z}$ bezeichnet die Menge der mit m teilbaren ganzen Zahlen.

Satz 5.16. Die folgenden sind kommutative Gruppen:

- $(\mathbb{Z}, +)$
- $(\mathbb{Q}, +)$
- $(\mathbb{R}, +)$
- $(\mathbb{C}, +)$
- $(\mathbb{Q}^+, \cdot)$
- $(\mathbb{R}^+, \cdot)$
- $(\mathbb{Q} \setminus \{0\}, \cdot)$
- $(\mathbb{R} \setminus \{0\}, \cdot)$
- $(\mathbb{C} \setminus \{0\}, \cdot)$
- $(\mathbb{Z}_m, +)$
- $(m\mathbb{Z}, +)$
- $(\mathbb{Z}_m^*, \cdot)$
- $(n \times m$-Matrizen$, +)$

Die folgenden sind nicht-kommutative Gruppen:

- Diedergruppe D_n
- symmetrische Gruppe S_n
- (invertierbare $n \times n$-Matrizen$, \cdot)$

Die folgenden sind Halbgruppen, jedoch keine Gruppen:

- $(\mathbb{N}, +)$
- $(\mathbb{N}, \cdot)$
- $(\mathbb{Z}, \cdot)$
- $(\mathbb{Q}^+, +)$
- $(\mathbb{R}^+, +)$
- $(\mathbb{Z}_m, \cdot)$

- $(n \times n$-Matrizen, $\cdot)$

Satz 5.17. Nehmen wir an, in der Halbgruppe $(M, *)$ gibt es ein rechtsneutrales Element e, d.h. $\forall a \in M : a * e = a$. Nehmen wir des Weiteren an, dass jedes Element eine Rechtsinverse hat, d.h. $\forall a \in M\ \exists a' \in M : a * a' = e$. Dann ist $(M, *)$ eine Gruppe.

Satz 5.18. Nehmen wir an, in der Halbgruppe $(M, *)$ gibt es zu jedem Paar $a, b \in M$ genau ein $x \in M$ mit $a * x = b$ und genau ein $y \in M$ mit $y * a = b$. Dann ist $(M, *)$ eine Gruppe.

Satz 5.19. Sei G eine Gruppe, $x, y, z \in G$ mit $xy = xz$. Dann gilt $y = z$.

Definition 5.20 (Cayley-Tabelle)**.** Sei G eine endliche Gruppe mit n Elementen. Die *Cayley-Tabelle* von G ist eine $n \times n$-Tabelle, in der jede Zeile und jede Spalte genau einem Element von G entspricht. Wenn $x, y \in G$, dann enthält die Zelle der Tabelle in Zeile x, Spalte y das Element $xy \in G$.

Satz 5.21. Sei G eine endliche Gruppe. Dann enthält jede Zeile und jede Spalte in der Cayley-Tabelle von G jedes Element von G genau einmal.

5.1.2 Untergruppen und Homomorphismen

Definition 5.22 (Untergruppe)**.** Sei $(G, *)$ eine Gruppe, H eine nichtleere Teilmenge von G. Wenn $(H, *)$ auch eine Gruppe ist, dann ist es eine *Untergruppe* von G, Notation: $H \leq G$.

Satz 5.23. Sei $(G, *)$ eine Gruppe, H eine nichtleere Teilmenge von G. Die Folgenden sind äquivalent:
(1) $H \leq G$
(2) $a, b \in H \Rightarrow a * b \in H,\ a^{-1} \in H$
(3) $a, b \in H \Rightarrow a * b^{-1} \in H$

Definition 5.24 (triviale Untergruppe)**.** Sei $(G, *)$ eine Gruppe, $e \in G$ das neutrale Element. Die Untergruppen $(\{e\}, *)$ und $(G, *)$ sind die *trivialen Untergruppen* von G.

Definition 5.25 (Zentrum)**.** Sei $(G, *)$ eine Gruppe. Das *Zentrum* von G ist: $Z(G) = \{a \in G \mid \forall b \in G : a * b = b * a\}$.

Satz 5.26. Sei G eine Gruppe. Dann ist $Z(G) \leq G$.

Satz 5.27. $(\mathbb{Z}, +) \leq (\mathbb{Q}, +) \leq (\mathbb{R}, +) \leq (\mathbb{C}, +)$.
$(\{$Drehungen in $D_m\}$, Komposition$) \leq D_m$.

Definition 5.28 (Gruppenhomomorphismus)**.** Eine Abbildung $f : A \to B$ heißt *Gruppenhomomorphismus* zwischen den Gruppen $(A, *)$ und $(B, \circ)$, wenn für alle $x, y \in A$ gilt, dass $f(x * y) = f(x) \circ f(y)$.

Satz 5.29. Sei f ein Gruppenhomomorphismus zwischen den Gruppen $(A, *)$ und $(B, \circ)$. Sei e_A das neutrale Element in A, e_B das neutrale Element in B. Dann gilt:
(1) $f(e_A) = e_B$
(2) $\forall a \in A : f(a^{-1}) = (f(a))^{-1}$

Definition 5.30 (Gruppenisomorphismus)**.** Ist ein Gruppenhomomorphismus $f : A \to B$ bijektiv, dann ist es ein *Gruppenisomorphismus* und die Gruppen A und B heißen zueinander isomorph.

Definition 5.31 (Bild, Kern)**.** Sei $f : A \to B$ ein Gruppenhomomorphismus. Dann ist das *Bild* von f: $Imf = \{f(a) \mid a \in A\}$ und der *Kern* von f ist: $Kerf = \{a \in A \mid f(a) = e_B\}$.

Satz 5.32. Sei $f : A \to B$ ein Gruppenhomomorphismus. Dann ist $Imf \leq B$ und $Kerf \leq A$.

Satz 5.33. Sei $m \in \mathbb{Z}$, $m > 1$. Dann ist $f : (\mathbb{Z}, +) \to (\mathbb{Z}, +)$, $f(x) = mx$ ein Gruppenhomomorphismus, $Imf = m\mathbb{Z}$ und $Kerf = \{0\}$.

Satz 5.34. Sei $m \in \mathbb{Z}$, $m > 1$. Dann ist $f : (\mathbb{Z}, +) \to (m\mathbb{Z}, +)$, $f(x) = mx$ ein Gruppenisomorphismus, $Imf = m\mathbb{Z}$ und $Kerf = \{0\}$.

Satz 5.35. Sei $f : A \to B$ ein Gruppenhomomorphismus. f ist ein Isomorphismus genau dann wenn $Kerf = \{0\}$ und $Imf = B$.

Satz 5.36. Sei $m \in \mathbb{Z}$, $m > 1$. Dann ist $f : (\mathbb{Z}, +) \to (\mathbb{Z}_m, +)$, $f(x) = x \pmod m$ ein Gruppenhomomorphismus, $Imf = \mathbb{Z}_m$ und $Kerf = m\mathbb{Z}$.

Satz 5.37. $f : (\mathbb{R}^+, \cdot) \to (\mathbb{R}, +)$, $f(x) = \log x$ ist ein Gruppenisomorphismus.

Satz 5.38. Sei $f : (\mathbb{Z}_m, +) \to (\{\text{Drehungen in } D_m\}, \text{Komposition})$, $f(k) =$ Drehung um $\frac{2\pi}{n}k$. Dann ist f ein Gruppenisomorphismus.

5.1.3 Zyklische Gruppen

Satz 5.39. Sei G eine Gruppe. Dann ist die Schnittmenge von beliebig vielen Untergruppen von G auch eine Untergruppe von G.

Definition 5.40 (generierte Untergruppe)**.** Sei $G = (M, *)$ eine Gruppe, T eine Teilmenge von M. Die von T *generierte Untergruppe* (Notation: $\langle T \rangle$) ist der Durchschnitt aller Untergruppen von G, die T enthalten. Diese ist gleichzeitig die kleinste Untergruppe von G im Sinne der Inklusion, die die Eigenschaft besitzt T als Teilmenge zu enthalten.

Satz 5.41. Sei G eine Gruppe, $a \in G$. Dann ist $\langle \{a\} \rangle = \{a^n \mid n \in \mathbb{Z}\}$.

Definition 5.42 (Ordnung eines Elements)**.** Sei G eine Gruppe, $e \in G$ das neutrale Element, $a \in G$. Die *Ordnung* von a (Notation: $o(a)$) ist die kleinste positive ganze Zahl m so dass $a^m = e$ oder ∞ wenn es kein solches m gibt.

Satz 5.43. Sei G eine Gruppe, $a \in G$, $o(a) = m < \infty$. Dann ist $\langle \{a\} \rangle = \{a^n \mid 0 \leq n \leq m-1\}$.

Definition 5.44 (zyklische Gruppe)**.** Die Gruppe G ist *zyklisch*, wenn $G = \langle \{a\} \rangle$ für ein $a \in G$.

Satz 5.45. In einer zyklischen Gruppe ist die Ordnung der Gruppe gleich der Ordnung des generierenden Elements. Allgemeiner, wenn $a \in G$, dann gilt $o(a) = | < a > |$.

Satz 5.46. Sei G eine zyklische Gruppe. Wenn G unendlich viele Elemente hat, dann ist G isomorph zu $(\mathbb{Z}, +)$. Wenn G aus n Elementen besteht, dann ist G isomorph zu $(\mathbb{Z}_n, +)$.

Definition 5.47 (C_n)**.** Die zyklische Gruppe mit n Elementen wird mit C_n, die zyklische Gruppe mit unendlich vielen Elementen mit C_∞ bezeichnet.

5.1.4 Nebenklassen

Definition 5.48 (Nebenklasse)**.** Sei G eine Gruppe, $H \leq G$, $g \in G$. Dann ist $gH := \{gh : h \in H\}$ eine *linksseitige Nebenklasse*, $Hg := \{hg : h \in H\}$ eine *rechtsseitige Nebenklasse*.

Satz 5.49. Sei G eine Gruppe, $H \leq G$, $g \in G$. Dann gilt $g \in Hg$.

Satz 5.50. Sei G eine Gruppe, $H \leq G$, $g_1, g_2 \in G$. Dann ist entweder $Hg_1 = Hg_2$ oder $Hg_1 \cap Hg_2 = \emptyset$.

Satz 5.51. Sei G eine Gruppe, $H \leq G$ eine endliche Untergruppe, $g \in G$. Dann gilt $|Hg| = |H|$.

Definition 5.52 (Index)**.** Sei G eine endliche Gruppe, $H \leq G$. Die Anzahl der zu H gehörenden Nebenklassen ist der *Index* von H. Bezeichnung: $|G : H|$.

Satz 5.53 (Lagrange)**.** Sei G eine endliche Gruppe, $H \leq G$. Dann gilt $|G| = |H| \cdot |G : H|$.

Korollar 5.54. Sei G eine endliche Gruppe, $a \in G$. Dann ist $o(a)$ ein Teiler von $|G|$.

Korollar 5.55. Sei G eine endliche Gruppe mit neutralem Element e, $a \in G$. Dann ist $a^{|G|} = e$.

5.2 Ringe und Körper

5.2.1 Grundbegriffe

Definition 5.56 (Ring)**.** $(M, +, *)$ ist ein *Ring*, wenn die folgenden Eigenschaften erfüllt sind:

- $(M, +)$ ist eine kommutative Gruppe
- $(M, *)$ ist eine Halbgruppe
- $\forall a, b, c \in M : a * (b + c) = a * b + a * c$ und $(a + b) * c = a * c + b * c$ (Distributivität)

Satz 5.57. Sei $(M, +, *)$ ein Ring und sei $0 \in M$ das neutrale Element der Gruppe $(M, +)$. Dann gilt $\forall a \in M : a * 0 = 0 * a = 0$.

Definition 5.58 (kommutativer Ring; Ring mit Einselement)**.** Sei $R = (M, +, *)$ ein Ring. Wenn $(M, *)$ eine kommutative Halbgruppe ist, dann ist R ein *kommutativer Ring*. Wenn es ein neutrales Element in der Halbgruppe $(M, *)$ gibt, dann ist R ein *Ring mit Einselement*.

Definition 5.59 (Nullteiler)**.** Sei $R = (M, +, *)$ ein Ring und sei $0 \in M$ das neutrale Element der Gruppe $(M, +)$. Wenn $a, b \neq 0$ aber $a * b = 0$, dann ist a ein *linksseitiger Nullteiler* und b ein *rechtsseitiger Nullteiler*.

Definition 5.60 (nullteilerfrei, Integritätsbereich)**.** Ein Ring, in dem es keine Nullteiler gibt, ist *nullteilerfrei*. Ein kommutativer nullteilerfreier Ring ist ein *Integritätsbereich*.

Definition 5.61 (Körper)**.** $(M, +, *)$ ist ein *Körper*, wenn die folgenden Eigenschaften erfüllt sind:

- $(M, +)$ ist eine kommutative Gruppe mit neutralem Element 0
- $(M \setminus \{0\}, *)$ ist eine kommutative Gruppe
- $\forall a, b, c \in M : a * (b + c) = a * b + a * c$ (Distributivität)

Satz 5.62. Jeder Körper ist ein Integritätsbereich.

Definition 5.63 (Polynomring)**.** Sei K ein Körper. Der *Polynomring* $K[x]$ besteht aus den Polynomen mit einem Variablen x und Koeffizienten aus K.

Satz 5.64. Die folgenden Strukturen sind Ringe:

- $(\mathbb{Z}, +, *)$, $(\mathbb{N}, +, *)$: Integritätsbereich mit Einselement
- $(\mathbb{Q}, +, *)$, $(\mathbb{R}, +, *)$, $(\mathbb{C}, +, *)$: Körper
- $(\mathbb{Z}_m, +, *)$: Körper, falls m eine Primzahl ist; sonst kommutativer Ring mit Einselement
- $(m\mathbb{Z}, +, *)$: Integritätsbereich
- $(\mathbb{R} \to \mathbb{R}$-Funktionen$, +, *)$: kommutativer Ring mit Einselement
- (Teilmengen einer gegebenen Menge$, \triangle, \cap)$: kommutativer Ring mit Einselement
- $(n \times n$-Matrizen$, +, *)$: Ring mit Einselement
- $(K[x], +, *)$: Integritätsbereich mit Einselement

Bemerkung 5.65. Viele Begriffe der Zahlentheorie lassen sich ohne Weiteres für beliebige Ringe verallgemeinern, z.B. Teilbarkeit, größter gemeinsamer Teiler, irreduzible und Primelemente.

Satz 5.66. Wenn es in einem Integritätsbereich eine Ordnungsrelation gibt, so dass Satz 4.8 (Division mit Rest) erfüllt ist, dann gilt in diesem Ring der Fundamentalsatz der Arithmetik.

Korollar 5.67. Sei K ein Körper. Dann gilt im Polynomring $K[x]$ der Fundamentalsatz der Arithmetik.

5.2.2 Körpererweiterungen

Definition 5.68 (Körpererweiterung)**.** Sei $(L, +, *)$ ein Körper, $(K, +, *)$ ein Teilkörper von L, d.h. eine Teilmenge, die mit denselben Operationen einen Körper bildet. Dann ist das Paar $L|K$ eine *Körpererweiterung*.

Satz 5.69. Sei $L|K$ eine Körpererweiterung. Dann ist L ein Vektorraum über K.

Definition 5.70 (Grad der Körpererweiterung)**.** Sei $L|K$ eine Körpererweiterung. Der *Grad* der Körpererweiterung ist die Dimension von L als Vektorraum über K (Notation: $|L : K|$).

Satz 5.71. Jeder Körper enthält als Teilkörper entweder $\mathbb{Q}$ oder $\mathbb{Z}_p$ für eine Primzahl p. Im ersten Fall gibt es kein $a \neq 0$ und $k \geq 1$ mit

$$\underbrace{a + \ldots + a}_{k} = 0,$$

im zweiten Fall ist p das kleinste solche k für jedes Element $a \neq 0$.

Definition 5.72 (Charakteristik)**.** Die *Charakteristik* eines Körpers K (Notation: $char(K)$) ist 0, wenn $\mathbb{Q} \leq K$ und p, wenn $\mathbb{Z}_p \leq K$.

Satz 5.73. Sei K ein endlicher Körper. Dann ist $char(K) = p$ für eine Primzahl p und die Anzahl der Elemente in K ist eine Potenz von p.

Definition 5.74 (GF)**.** Ein endlicher Körper mit q Elementen wird mit $GF(q)$ bezeichnet.

Definition 5.75 (Adjunktion, einfache Erweiterung, algebraisch, transzendent)**.** Sei $L|K$ eine Körpererweiterung, $a \in L \setminus K$. Dann bezeichne $K(a)$ den kleinsten Körper (im Sinne der Inklusion), der sowohl K als auch a enthält. $K(a)$ wird als *Adjunktion* von a zu K bezeichnet. Körper, die durch die Adjunktion eines Elementes aus K gewonnen werden können, sind die *einfachen Erweiterungen* von K. Wenn es ein Polynom $f \in K[x]$ gibt, so dass $f(a) = 0$, dann ist a ein *algebraisches Element* über K und $K(a)|K$ ist eine *algebraische Erweiterung*. Wenn es kein solches Polynom gibt, dann ist a ein *transzendentes Element* über K und $K(a)|K$ ist eine *transzendente Erweiterung*.

Satz 5.76. $\mathbb{Q}(\sqrt{2})$ ist eine einfache algebraische Erweiterung zweiten Grades von $\mathbb{Q}$. $\mathbb{Q}(\sqrt{2}) = \{a+b\sqrt{2} : a, b \in \mathbb{Q}\}$.

Satz 5.77. $\mathbb{R}(i)$ ist eine einfache algebraische Erweiterung zweiten Grades von $\mathbb{R}$. $\mathbb{R}(i) = \mathbb{C} = \{a + bi : a, b \in \mathbb{R}\}$.

Satz 5.78. $\mathbb{Q}(\pi)$ ist eine einfache transzendente Erweiterung von $\mathbb{Q}$. $\mathbb{Q}(\pi) = \{\sum_{i=0}^{n} a_i \pi^i : n \in \mathbb{N}, a_i \in \mathbb{Q}\}$.

Satz 5.79. Sei $K(a)|K$ eine algebraische Erweiterung, $f \in K[x]$ irreduzibel und $f(a) = 0$. Dann ist grf gleich dem Grad der Erweiterung.

Satz 5.80. Sei $K \leq L \leq M$. Dann gilt $|M : K| = |M : L| \cdot |L : K|$.

Satz 5.81 (Fundamentalsatz der Algebra)**.** $\mathbb{C}$ ist algebraisch geschlossen, d.h. für jedes Polynom $f \in \mathbb{C}[x]$ mit $grf \geq 1$ gibt es ein $a \in \mathbb{C}$, so dass $f(a) = 0$.

5.2.3 Anwendung: Konstruktion mit Lineal und Zirkel

Satz 5.82 (Galois)**.** Eine Strecke der Länge $x \in \mathbb{R}$ ist konstruierbar mit Lineal und Zirkel dann und nur dann, wenn es eine Folge $\mathbb{Q} \leq K_1 \leq K_2 \leq \ldots \leq K_n$ von algebraischen Erweiterungen zweiten Grades gibt, so dass $x \in K_n$.

Korollar 5.83. $\sqrt{\pi}$ ist nicht konstruierbar, und damit ist die Quadratur des Kreises nicht machbar.

Korollar 5.84. $\sqrt[3]{2}$ ist nicht konstruierbar, und damit ist die Würfeldopplung nicht machbar.

Korollar 5.85. $\cos(20°)$ ist nicht konstruierbar, und damit ist die Winkeldrittelung nicht machbar.

5.2.4 Anwendung: Fehlererkennende und -korrigierende Codes

Definition 5.86 (Reed-Solomon-Code)**.** Sei K ein endlicher Körper, $u_1, \ldots, u_n$ seien paarweise verschiedene, fixierte Elemente von K.

Algorithmus 5.87 (Reed-Solomon-Code: Vorbereitung).
`Input`:
n: gewünschte Codelänge.
`Output`:
K: endlicher Körper mit mindestens n Elementen
$u_1, \ldots, u_n$: paarweise verschiedene, fixierte Elemente von K.
`Ablauf`:

Algorithmus 5.88 (Reed-Solomon-Code: Codieren einer Nachricht).
`Input`:
$a_1 \ldots a_k$: zu codierende Nachricht ($k < n$).
`Output`:
$a_1 \ldots a_k a_{k+1} \ldots a_n$: codierte Nachricht.
`Ablauf`:
1. Durch Interpolation wird ein Polynom $f \in K[x]$ bestimmt, für das $f(u_1) = a_1, \ldots, f(u_k) = a_k$ und $grf \leq k-1$
2. $a_{k+1} := f(u_{k+1}), \ldots, a_n := f(u_n)$

Algorithmus 5.89 (Reed-Solomon-Code: Überprüfen ob eine Nachricht fehlerfrei ist).
`Input`:
$a_1 \ldots a_n$: Nachricht.
`Output`:
Ob die Nachricht fehlerfrei übertragen wurde.
`Ablauf`:
1. Durch Interpolation wird ein Polynom $f \in K[x]$ bestimmt, für das $f(u_1) = a_1, \ldots, f(u_k) = a_k$ und $grf \leq k-1$
2. Es wird überprüft, dass $a_{k+1} = f(u_{k+1}), \ldots, a_n = f(u_n)$

Algorithmus 5.90 (Reed-Solomon-Code: Decodieren einer Nachricht mit unlesbar gewordenen Stellen).
`Input`:
$a_1 \ldots a_n$: Nachricht mit bis zu $n-k$ unlesbar gewordenen Stellen.
`Output`:
$a_1 \ldots a_k$: decodierte Nachricht.
`Ablauf`:
1. Es werden k lesbare Stellen $i_1, \ldots, i_k$ der Nachricht identifiziert
2. Durch Interpolation wird ein Polynom $f \in K[x]$ bestimmt, für das $f(u_{i_1}) = a_{i_1}, \ldots, f(u_{i_k}) = a_{i_k}$ und $grf \leq k-1$
3. $a_1 := f(u_1), \ldots, a_k := f(u_k)$

Kapitel 6

Kardinalität unendlicher Mengen

6.1 Grundbegriffe

Definition 6.1 ($|A| = |B|$)**.** Zwei Mengen A und B haben die gleiche Kardinalität (Notation: $|A| = |B|$), wenn es eine Bijektion zwischen den beiden Mengen gibt.

Definition 6.2 ($|A| \leq |B|$)**.** Die Kardinalität der Menge A ist kleiner oder gleich der Kardinalität der Menge B (Notation: $|A| \leq |B|$), wenn es eine Bijektion zwischen A und einer Teilmenge von B gibt.

Definition 6.3 ($|A| < |B|$)**.** Die Kardinalität der Menge A ist kleiner als die Kardinalität der Menge B (Notation: $|A| < |B|$), wenn $|A| \leq |B|$ richtig und $|A| = |B|$ falsch ist.

Satz 6.4 (Cantor-Bernstein)**.** Wenn $|A| \leq |B|$ und $|B| \leq |A|$, dann ist $|A| = |B|$.

6.2 Abzählbar unendliche Mengen

Definition 6.5 (abzählbar unendlich)**.** Die Menge A ist *abzählbar unendlich* (Notation: $|A| = \aleph_0$), wenn $|A| = |\mathbb{N}|$.

Satz 6.6. Die Menge A ist abzählbar unendlich genau dann wenn $A = \{a_0, a_1, a_2, \ldots\}$, d.h. wenn man aus den Elementen von A eine Folge bilden kann.

Satz 6.7. Die folgenden Mengen sind abzählbar unendlich:

- $\mathbb{Z}^+$
- $\mathbb{Z}$
- Die Menge der geraden Zahlen

Satz 6.8. Seien A und B zwei disjunkte Mengen, A abzählbar unendlich, B endlich oder abzählbar unendlich. Dann ist $A \cup B$ abzählbar unendlich (Notation: $\aleph_0 + k = \aleph_0$, $\aleph_0 + \aleph_0 = \aleph_0$).

Satz 6.9. Sei A abzählbar unendlich, $B \subseteq A$ endlich. Dann ist $A \setminus B$ abzählbar unendlich (Notation: $\aleph_0 - k = \aleph_0$,).

Satz 6.10. Sei A abzählbar unendlich, $B \subseteq A$. Dann ist B entweder endlich oder abzählbar unendlich.

Satz 6.11. Sei A eine unendliche Menge. Dann gibt es $B \subseteq A$ mit $|B| = \aleph_0$.

Satz 6.12. Seien A und B zwei disjunkte Mengen, A unendlich, B endlich oder abzählbar unendlich. Dann ist $|A \cup B| = |A|$.

Satz 6.13. Sei A eine unendliche Menge, $B \subseteq A$ endlich. Dann ist $|A \setminus B| = |A|$.

Korollar 6.14. Eine Menge ist genau dann unendlich, wenn sie eine echte Teilmenge mit gleicher Kardinalität hat.

Satz 6.15. Seien $A_0, A_1, A_2, \ldots$ paarweise disjunkte, abzählbar unendliche Mengen. Dann ist $\cup A_i$ abzählbar unendlich (Notation: $\aleph_0 \cdot \aleph_0 = \aleph_0$).

Satz 6.16. Sei A abzählbar unendlich. Dann ist $A \times A$ auch abzählbar unendlich.

Satz 6.17. $\mathbb{Q}$ ist abzählbar unendlich.

6.3 Continuum

Satz 6.18 (Cantor)**.** Sei $I = (0, 1)$ die Menge der reellen Zahlen zwischen 0 und 1. Dann ist $|I| > \aleph_0$.

Definition 6.19 (Continuum)**.** Die Kardinalität von $(0, 1)$ wird *Continuum* genannt.

Satz 6.20. Für beliebige reelle Zahlen $a < b$ ist $|(0, 1)| = |(a, b)| = |\mathbb{R}|$.

Satz 6.21. Sei $k \in \mathbb{Z}^+$. Dann ist $|\mathbb{R}^k| = |\mathbb{R}| = |\mathbb{C}|$.

Definition 6.22 (Continuum-Hypothese)**.** Die *Continuum-Hypothese* besagt, dass es keine Kardinalität zwischen $\aleph_0$ und continuum gibt, d.h. wenn $A \subseteq \mathbb{R}$, dann ist die Kardinalität von A entweder endlich, oder abzählbar unendlich, oder continuum.

Bemerkung 6.23. Es ist bewiesen, dass die Continuum-Hypothese unentscheidbar ist, d.h. aus den üblichen Axiomen der Mengenlehre weder die Continuum-Hypothese noch die Negation der Continuum-Hypothese bewiesen werden kann.

6.4 Potenzmengen

Definition 6.24 (Potenzmenge)**.** Die Potenzmenge $P(A)$ einer Menge A ist die Menge aller Teilmengen von A, d.h. $x \in P(A)$ genau dann wenn $x \subseteq A$.

Satz 6.25 (Cantor)**.** Für eine beliebige Menge A gilt: $|P(A)| > |A|$.

Korollar 6.26. Es gibt keine größte Kardinalität.

Definition 6.27 (verallgemeinerte Continuum-Hypothese)**.** Die *verallgemeinerte Continuum-Hypothese* besagt, dass es keine Kardinalität zwischen $|A|$ und $|P(A)|$ gibt, d.h. wenn $B \subseteq P(A)$ und $|B| \geq |A|$, dann gilt entweder $|B| = |A|$ oder $|B| = |P(A)|$.

Bemerkung 6.28. Es ist bewiesen, dass die verallgemeinerte Continuum-Hypothese unentscheidbar ist, d.h. aus den üblichen Axiomen der Mengenlehre weder die Hypothese noch die Negation der Hypothese bewiesen werden kann.

Satz 6.29. $|P(\mathbb{N})| = |\{\text{unendliche } 0-1\text{-Folgen}\}| =$continuum.

6.5 Anwendung: algebraische Zahlen

Definition 6.30. Sei $\mathbb{A}$ die Menge der algebraischen Elemente in $\mathbb{R}|\mathbb{Q}$, d.h. der reellen Lösungen von Polynomen mit rationalen Koeffizienten. Die Elemente von $\mathbb{A}$ sind die *algebraischen Zahlen*, die Elemente von $\mathbb{R} \setminus \mathbb{A}$ die *transzendenten Zahlen*.

Satz 6.31. $|\mathbb{A}| = \aleph_0$.

Korollar 6.32. Es gibt transzendente Zahlen.

Literaturverzeichnis

[1] CORMEN, TH. H., LEISERSON, CH. E., RIVEST, R. L.: *Algorithmen – eine Einführung*. Oldenbourg, 2. Auflage, 2007.

[2] DIESTEL, R.: *Graphentheorie*. Springer-Verlag, Heidelberg, 3. Auflage, 2006.

[3] KATONA GY. Y, RECSKI A., SZABÓ CS.: *A számítástudomány alapjai*. Typotex Kiadó, Budapest, 2002.

[4] RÓNYAI L., IVANYOS G., SZABÓ R.: *Algoritmusok*. Typotex Kiadó, Budapest, 1999.

[5] VOLKMANN, L.: *Graphen an allen Ecken und Kanten*. RWTH Aachen, `http://www.math2.rwth-aachen.de/~uebung/GT/volkm_gt.pdf`, 2006.