Diskrete Strukturen
Skript zur Vorlesung

Manuel Bodirsky,
Institut für Algebra, TU Dresden,
Manuel.Bodirsky@tu-dresden.de

7. Februar 2025

Es handelt sich hier um eine Vorlesung, die vom Institut für Algebra der TU Dresden für Student:innen der Informatik angeboten wird. Es werden Themen der Mathematik behandelt, die Student:innen der Informatik kennen sollten. Die hier vorgestellten Konzepte sind wichtige Werkzeuge in der Informatik. Vor allem aber führt diese Vorlesung international und fächerübergreifend gültige mathematische Sprache und Notation ein, und vermittelt formales Argumentieren und Beweistechniken.

Dieses Skript und die Vorlesung basieren auf Vorlesungen, die von Kolleg:innen am Institut für Algebra in den letzten Jahren gehalten wurden, und denen ich an dieser Stelle danken möchte. Für die vielen Rückmeldungen zum Skript danke ich Antje Noack, Ellen Henke, Waltraud Lederle, und den Student:innen. Wir folgen keinem Lehrbuch; wer aber nach ergänzender Literatur sucht, sei auf das Buch *Invitation to discrete mathematics* von Matoušek und Nešetřil verwiesen [6], welches auch auf Deutsch unter dem Titel *Diskrete Mathematik: Eine Entdeckungsreise* erhältlich ist. Die 56 Übungen im Text sind zum Vertiefen gedacht, und ersetzen nicht die Übungen, die im Übungsbetrieb der Vorlesung bearbeitet werden. Übungen mit einem Stern ($*$) sind schwerer und zum Knobeln. Fehler, Tippfehler, Lücken, Kommentare, Anregungen, usw. bitte an den Autor richten, sie sind herzlich willkommen!

„Si bien doué que l'on soit, on ne fait rien de grand sans travail"
Henri Poincaré[1]

[1] Henri Poincaré, geboren am 29 April 1854 in Nancy; gestorben am 17 Juli 1912 in Paris.

Inhaltsverzeichnis

1 Die Sprache der modernen Mathematik **5**
1.1 Die Symbole der Mengensprache . . . 5
1.2 Mengenangaben durch Aussondern . . . 5
1.3 Mengenoperationen . . . 6
1.4 Mengenkomplement . . . 7
1.5 Kodieren mit Mengen . . . 8
1.6 Doppeltes Abzählen . . . 9
1.7 Binomialkoeffizienten . . . 9
1.8 Die Russellsche Antinomie . . . 11
1.9 Die Axiome von Zermelo-Fraenkel . . . 12

2 Abbildungen **14**
2.1 Notation . . . 14
2.2 Größenvergleich von Mengen . . . 15
2.3 Der Satz von Cantor-Schröder-Bernstein . . . 15
2.4 Das Auswahlaxiom . . . 16
2.5 Die Kontinuumshypothese . . . 17
2.6 Permutationen . . . 18

3 Boolsche Funktionen und Aussagenlogik **21**
3.1 Boolsche Funktionen . . . 21
3.2 Aussagenlogik . . . 25
3.3 Das Erfüllbarkeitsproblem . . . 28
3.4 Horn-SAT . . . 30

4 Die natürlichen Zahlen **33**
4.1 Die Wohlordnung der natürlichen Zahlen . . . 33
4.2 Vollständige Induktion . . . 34
4.3 Addition, Multiplikation, Exponentiation . . . 35
4.4 Teilbarkeit und Primzahlen . . . 36
4.5 Der euklidische Algorithmus . . . 38

5 Modulare Arithmetik **42**
5.1 Die Homomorphieregel . . . 42
5.2 Uhrzeiten . . . 44
5.3 Die letzten Ziffern . . . 44
5.4 Potenzieren modulo n . . . 44
5.5 Der chinesische Restsatz . . . 45
5.6 Zufall in der Informatik . . . 47
5.7 Anwendung: Rechnen mit großen Zahlen . . . 48

6 Gruppen **52**
6.1 Beispiele . . . 53
6.2 Die multiplikative Gruppe von $\mathbb{Z}_n$ . . . 54
6.3 Zyklische Gruppen . . . 57
6.4 Öffentlich ein Geheimnis vereinbaren . . . 59
6.5 Der Satz von Lagrange . . . 60
6.6 Das Lemma von Euler-Fermat . . . 62
6.7 Kryptographie mit öffentlichen Schlüsseln . . . 63

7 Graphen **69**
7.1 Knotenzusammenhang . . . 71
7.2 Färbbarkeit . . . 71
7.3 Bäume . . . 73
7.4 Zweifacher Zusammenhang . . . 75
7.5 Der Satz von Menger . . . 77
7.6 Kantenzusammenhang und Kantengraphen . . . 80
7.7 Eulersche Graphen . . . 81
7.8 Paarungen . . . 82

8 Äquivalenzrelationen **86**
8.1 Äquivalenz und Partition . . . 87
8.2 Partitionen zählen . . . 88
8.3 Der Kern einer Abbildung . . . 90
8.4 Funktionen zählen . . . 90

9 Ordnungsrelationen **92**
9.1 Lineare Erweiterungen . . . 93
9.2 Quasiordnungen . . . 94
9.3 Transitive Hülle . . . 95
9.4 Ketten, Antiketten, Dilworth . . . 95
9.5 Wohlquasiordnungen . . . 97
9.6 Semilineare Ordnungen . . . 101

10 Bäume zählen **105**
10.1 Beschriftete und unbeschriftete Graphen . . . 105
10.2 Die Formel von Cayley . . . 105
10.3 Der Satz von Kirchhoff . . . 108

11 Planare Graphen **115**
11.1 Die eulersche Polyederformel . . . 116
11.2 Triangulationen . . . 117
11.3 Zeichnungen . . . 118

11.4 Polyeder 121
11.5 Der Dualgraph 121
11.6 Minoren und der Satz von Kuratowski-Wagner 122

12 Gerichtete Graphen **125**
12.1 Tiefensuche 126
12.2 Starke Zusammenhangskomponenten 127
12.3 Anwendung: 2-SAT 130
12.4 Breitensuche 133
12.5 Transportnetze 134
12.6 Der Algorithmus von Edmonds-Karp 138
12.7 Nochmal Paarungen 140

Index **142**

1 Die Sprache der modernen Mathematik

Wenn im Text ein Begriff *hervorgehoben* gedruckt ist, dann soll damit deutlich gemacht werden, dass er an dieser Stelle eingeführt wird. Gewöhnlich geschieht das durch eine Definition. Eine Definition fehlt bei den undefinierten Grundbegriffen (z.B. „Menge") und bei Begriffen, die wir als bekannt voraussetzen (z.b. „reelle Zahl").

1.1 Die Symbole der Mengensprache

Wir schreiben $\emptyset$ für die *leere Menge*, die Menge, die kein Element hat.

- $e \in M$ bedeutet: e ist ein Element der Menge M.
- $e \notin M$ bedeutet: e ist nicht Element der Menge M.
- $T \subseteq M$ bedeutet: T ist eine *Teilmenge* von M, d.h., jedes Element der Menge T ist auch Element der Menge M.

Um Mengen anzugeben, verwenden wir Mengenklammern: Beispielsweise steht $\{1, 2, 3\}$ für die Menge mit genau den Elementen 1, 2, und 3. Mengen können als Elemente von anderen Mengen auftauchen: beispielsweise ist $\{1, \{2, 3\}\}$ eine Menge. Mit $|M|$ bezeichnen wir die Anzahl der Elemente von M, auch *Kardinalität*[2] von M genannt; beispielsweise ist $|\{1, 2, 3\}| = 3$ und $|\{1, \{2, 3\}\}| = 2$. Einige wichtige Mengen und deren Bezeichnungen:

$\mathbb{N}$: die Menge der *natürlichen Zahlen* $\{0, 1, 2, \dots\}$.

$\mathbb{Z}$: die Menge der *ganzen Zahlen* $\{0, 1, -1, 2, -2, \dots\}$.

$\mathbb{Q}$: die Menge der *rationalen Zahlen* ('Bruchzahlen').

$\mathbb{R}$: die Menge der *reellen Zahlen* ('Dezimalzahlen').

1.2 Mengenangaben durch Aussondern

Die allgemeine Form von *Mengenangaben durch Aussonderung* ist folgende:

$$\{x \in M \mid \text{Bedingung}(x)\}\,.$$

Damit ist die Menge aller Elemente in M gemeint, die die gegebene Bedingung erfüllen.

[2] Was dies im Falle von unendlichen Mengen M bedeutet, wird in Kapitel 2.2 thematisiert.

1.3 Mengenoperationen

Aus gegebenen Mengen A und B können mit Hilfe der folgenden Operationen neue Mengen konstruiert werden.

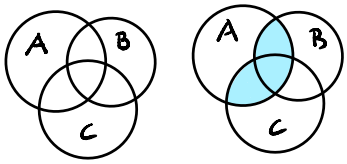
$A \cap B := \{e \mid e \in A \text{ und } e \in B\}$	Der *Schnitt* von A und B
$A \cup B := \{e \mid e \in A \text{ oder } e \in B\}$	Die *Vereinigung* von A und B
$A \setminus B := \{e \mid e \in A \text{ und } e \notin B\}$	Die *Differenz* von A und B

Hierbei bedeutet das Symbol ':=', dass wir den Ausdruck auf der linken Seite (beim Doppelpunkt) durch den Ausdruck auf der rechten Seite (beim Gleichheitszeichen) *definieren*.

Es gibt folgende Rechenregeln für Mengenoperationen:

$A \cap A = A$	Schnitt ist *idempotent*
$A \cup A = A$	Vereinigung ist *idempotent*
$A \cap B = B \cap A$	Schnitt ist *kommutativ*
$A \cup B = B \cup A$	Vereinigung ist *kommutativ*
$A \cap (B \cap C) = (A \cap B) \cap C$	Schnitte sind *assoziativ*
$A \cup (B \cup C) = (A \cup B) \cup C$	Vereinigungen sind *assoziativ*
$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	Schnitt ist *distributiv* über Vereinigung
$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	Vereinigung ist distributiv über Schnitt

Diese Rechenregeln können besonders einfach mit sogenannten Venn-Diagrammen verdeutlicht werden.



Für die Regel $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ beispielsweise sieht man, dass die Ausdrücke auf beiden Seiten die selbe farbige Fläche im Diagram beschreiben.

Auch unendlich viele Mengen können vereinigt oder geschnitten werden; dafür benötigen wir eine entsprechende Schreibweise. Sei M eine Menge, und für jedes $i \in M$ sei S_i eine Menge. Dann schreiben wir $\bigcap_{i \in M} S_i$ für die Menge $\{x \mid x \in S_i \text{ für alle } i \in M\}$, und $\bigcup_{i \in M} S_i$ für $\{x \mid x \in S_i \text{ für ein } i \in M\}$.

Beispiel 1. *Für jedes $i \in \mathbb{N}$ sei $S_i := \{0, 1, \ldots, i-1\}$. Es ist also $S_0 = \emptyset$, $S_1 = \{0\}$, $S_2 = \{0, 1\}$, $S_3 = \{0, 1, 2\}$, und so weiter. Dann gilt*

- $\bigcap_{i\in\mathbb{N}}(S_i) = \emptyset$,
- $\bigcup_{i\in\mathbb{N}}(S_i) = \mathbb{N}$.

Zwei Mengen A und B heissen *disjunkt* falls $A \cap B = \emptyset$.

Übungen.

1. Untersuchen Sie, ob die Mengendifferenz $A \setminus B$ eine assoziative Mengenoperation ist, d.h. ob
$$A \setminus (B \setminus C) = (A \setminus B) \setminus C$$
für alle Mengen A, B, C gilt.

2. Zeigen Sie, dass die *symmetrische Differenz*
$$A\Delta B := (A \setminus B) \cup (B \setminus A)$$
eine assoziative Mengenoperation ist, d.h., dass für alle Mengen A, B, C gilt, dass
$$A\Delta(B\Delta C) = (A\Delta B)\Delta C.$$

1.4 Mengenkomplement

Sei U eine feste Menge (U steht hier für Universum). Dann definieren wir für beliebige Teilmengen M von U das *Komplement von M (bezüglich U)* als die Menge

$$\overline{M} := U \setminus M = \{x \in U \mid x \notin M\}.$$

Wir bemerken, dass die Definition von $\overline{M}$ auch von U abhängt; allerdings sind in vielen Situationen nur Teilmengen einer gewissen Menge U interessant, z.B. nur Teilmengen der natürlichen Zahlen (oder nur Teilmengen der Knoten des Internet, etc) so dass es mühselig wäre, jedesmal 'bezüglich U' dazuzuschreiben. Der Vorteil dieser Notation ist, dass das U nicht immer mitgeschrieben werden muss, was zum Beispiel die folgenden Rechenregeln viel übersichtlicher macht.

$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$
$$\overline{A \cap B} = \overline{A} \cup \overline{B}$$
$$A \cap \overline{B} = A \setminus B$$

Auch hier ist es mit Hilfe von Venn-Diagrammen leicht, sich von der Korrektheit dieser Regeln zu überzeugen.

1.5 Kodieren mit Mengen

Mit Mengen können alle gewöhnlichen mathematischen Objekte 'kodiert' werden. Beispielsweise werden *geordnete Paare* (a, b) als Mengen der Gestalt $\{\{a\}, \{a, b\}\}$ definiert. Insbesondere sind geordnete Paare der Gestalt (a, a) erlaubt.

Proposition 1. *Falls $(a, b) = (c, d)$, dann gilt $a = c$ und $b = d$.*

Beweis. Falls $a = b$, dann ist $(a, b) = \{\{a\}, \{a, a\}\} = \{\{a\}\}$, und damit hat auch $(c, d) = \{\{c\}, \{c, d\}\}$ nur ein Element. Also $\{c\} = \{c, d\}$, und damit ist $c = d$ und $(c, d) = \{\{c\}\}$. Es folgt, dass $\{a\} = \{c\}$ und dass $c = a$, also $b = a = c = d$.

Falls $a \neq b$, dann hat $\{a, b\}$ zwei Elemente, und ist damit nicht gleich zu $\{a\}$. Also hat (a, b) zwei Elemente, und per Annahmen hat (c, d) zwei Elemente. Damit gilt $c \neq d$. Da $\{a\}$ das einzige Element von (a, b) ist mit nur einem Element, und $\{c\}$ das einzige Element von (c, d) mit nur einem Element, gilt $\{a\} = \{c\}$ und damit $a = c$. Ausserdem muss gelten dass $\{a, b\} = \{c, d\}$, da $\{a, b\}$ das einzige Element von (a, b) mit zwei Elementen ist, und $\{c, d\}$ das einzige Element von (c, d) ist mit zwei Elementen. Also $b = d$. $\square$

Die *Produktmenge* $A \times B$ zweier Mengen A, B wird definiert durch

$$A \times B := \{(a, b) \mid a \in A \text{ und } b \in B\}.$$

Es gilt $|A \times B| = |A| \times |B|$. Eine Teilmenge R von $A \times B$ wird auch (*binäre*, oder *zweistellige*) *Relation* genannt. Falls $A = B$, so wird R auch als *zweistellige Relation auf der Grundmenge* A bezeichnet.

Die *Potenzmenge* von A, geschrieben $\mathcal{P}(A)$, ist die Menge aller Teilmengen von A. Es gilt $|\mathcal{P}(A)| = 2^{|A|}$. Beispielsweise ist $|\mathcal{P}(\{1, 2, 3, 4\})| = 2^4 = 16$, denn um eine Teilmenge T von $\{1, 2, 3, 4\}$ zu bilden, muss für jedes $x \in \{1, 2, 3, 4\}$ die Entscheidung getroffen werden, ob x in die Menge T aufgenommen wird oder nicht. Es gibt also $2 \cdot 2 \cdot 2 \cdot 2 = 2^4$ Möglichkeiten, eine solche Menge zu bilden.

Übungen.

3. Geben Sie ein Beispiel für Mengen a, b, c, d an, so dass $\{a, \{b\}\} = \{c, \{d\}\}$, aber nicht $a = c$ und $b = d$ gilt.

4. Erklären Sie, an welcher Stelle der Beweis von Proposition 1 schief geht, wenn man als Definition von (a, b) die Menge $\{\{a, b\}, a\}$ verwendet, anstatt $\{\{a, b\}, \{a\}\}$.

5. Beweisen Sie, dass folgendes für beliebige Mengen A und B gilt.

   - $\mathcal{P}(A) \cap \mathcal{P}(B) = \mathcal{P}(A \cap B)$
   - $\mathcal{P}(A) \cup \mathcal{P}(B) \subseteq \mathcal{P}(A \cup B)$

1.6 Doppeltes Abzählen

„La mathématique est l'art de donner le même nom à des choses différentes"
Henri Poincaré

Das doppelte Abzählen ist ein Beweisprinzip, bei welchem eine gewisse Menge von Objekten auf zwei verschiedene Arten und Weisen abgezählt wird, um auf eine Gleichheit oder andere Aussage zu schließen. Ein gutes Beispiel dafür ist das sogenannte *Handschlaglemma*.

Proposition 2 (Handschlaglemma)**.** *Auf jeder Konferenz ist die Anzahl aller Teilnehmer:innen, die einer ungeraden Anzahl von Teilnehmer:innen die Hand gibt, immer gerade.*

Beweis. Seien $t_1, \ldots, t_n$ die Teilnehmer:innen der Konferenz. Wir zählen die Menge $\mathcal{P}$ von geordneten Paaren (t_i, t_j) von Teilnehmer:innen, die sich die Hand geben, auf zweierlei Art und Weise. Sei x_i die Anzahl von Personen, denen t_i die Hand reicht, und sei y die Anzahl aller Handschläge. Auf der einen Seite ist $|\mathcal{P}| = \sum_{i=1}^n x_i := x_1 + x_2 + \cdots + x_n$, denn es gibt x_i geordnete Paare in $\mathcal{P}$ mit t_i an erster Stelle. Auf der anderen Seite gibt es für jeden Handschlag zwei geordnete Paare (t_i, t_j) und (t_j, t_i) in $\mathcal{P}$. Also erhalten wir insgesamt

$$\sum_{i=1}^n x_i = 2y,$$

d.h., $\sum_{i=1}^n x_i$ ist eine gerade Zahl. Aber wenn die Summe von n Zahlen gerade ist, dann muss eine gerade Anzahl dieser Zahlen ungerade sein (denn wenn wir eine ungerade Anzahl von ungeraden Zahlen und eine beliebige Anzahl von geraden Zahlen addieren, so erhalten wir ein ungerades Ergebnis). □

1.7 Binomialkoeffizienten

Auf wie viele Arten und Weisen lassen sich n Elemente der Reihe nach aufzählen? Wir haben n Möglichkeiten, um das erste Element auszuwählen, dann $n-1$ Möglichkeiten, das zweite Element auszuwählen, usw. Wir schreiben $n!$ (lies: 'n *Fakultät*') für $n \cdot (n-1) \cdot (\cdots) \cdot 2 \cdot 1$, und definieren $0! := 1$. Die Anzahl der Aufreihungen einer Menge mit n Elementen ist also $n!$.

Für die Anzahl aller k-elementigen Teilmengen einer n-elementigen Menge schreiben wir $\binom{n}{k}$, gesprochen n *über* k. Aus Gründen, die später klar werden, heissen diese Zahlen *Binomialkoeffizienten*. Bemerke, dass $\binom{n}{0} = 1$ (die leere Menge) und $\binom{n}{n} = 1$ (die ganze Menge).

Proposition 3. *Für alle natürlichen Zahlen n, k gilt*

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

Insbesondere gilt $\binom{n}{2} = n(n-1)/2$.

Beweis. Um aus einer Menge mit n Elementen k Elemente auszuwählen, wählen wir ein erstes Element, dann ein zweites, und so weiter, bis zum k-ten Element. Dafür gibt es

$$n \cdot (n-1) \cdot (\cdots) \cdot (n-k+1) = n!/(n-k)!$$

Möglichkeiten. Nur spielt aber die Reihenfolge, in der wir die Elemente unserer Teilmenge ausgewählt haben, keine Rolle, und jeweils $k!$ Möglichkeiten führen zur gleichen Teilmenge. Die Gleichung folgt. □

Wir werden in der Folge wichtige Gleichungen für Binominalkoeffizienten sehen; diese Identitäten haben in der Regel sowohl einen *kombinatorischen* (z.B. durch doppeltes Abzählen!) als auch einen *algebraischen* Beweis (z.B. durch Zuhilfenahme der Formel aus Proposition 3).

Beobachtung 1. $\binom{n}{k} = \binom{n}{n-k}$

Kombinatorischer Beweis. Um aus n Spielern eine Mannschaft mit k Spielern aufzustellen, so ist es das gleiche, ob wir die k Spieler der Mannschaft auswählen, oder ob wir die $n-k$ Spieler auswählen, die nicht zur Mannschaft gehören sollen. □

Algebraischer Beweis. Die Aussage folgt direkt aus Proposition 3. □

Beobachtung 2. $\sum_{k=0}^{n} \binom{n}{k} := \binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{n} = 2^n$

Beweis. $\sum_{k=0}^{n} \binom{n}{k}$ zählt die Anzahl aller Möglichkeiten, aus einer n-elementigen Menge eine Teilmenge einer beliebigen Größe k zu bilden. Wie wir bereits früher in Abschnitt 1.5 gesehen haben, gibt es 2^n Teilmengen einer n-elementigen Menge. □

Proposition 4. *Für alle natürlichen Zahlen n, k gilt*

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

Beweis. Sei M eine $(n+1)$-elementige Menge, und sei $x \in M$ beliebig, fest gewählt. Um k Elemente aus M auszuwählen, können wir entweder x auswählen oder nicht. Wenn wir x auswählen, dann müssen weitere $k-1$ Elemente aus $M \setminus \{x\}$ gewählt werden, und dafür gibt es $\binom{n}{k-1}$ Möglichkeiten. Wenn wir x nicht auswählen, dann müssen weiterhin k Elemente aus $M \setminus \{x\}$ gewählt werden. Dafür gibt es $\binom{n}{k}$ Möglichkeiten. Insgesamt erhalten wir also $\binom{n}{k-1} + \binom{n}{k}$ Möglichkeiten, k Elemente aus M auszuwählen. □

$$
\begin{array}{lccccccccc}
n=0: & & & & & 1=\binom{0}{0} & & & & \\
n=1: & & & & 1=\binom{1}{0} & & 1=\binom{1}{1} & & & \\
n=2: & & & 1=\binom{2}{0} & & 2=\binom{2}{1} & & 1=\binom{2}{2} & & \\
n=3: & & 1 & & 3 & & 3 & & 1 & \\
n=4: & 1 & & 4 & & 6 & & 4 & & 1
\end{array}
$$

Abbildung 1: Pascalsches Dreieck

Die Werte von $\binom{n}{k}$ können vorteilhaft in einer Tabelle in Dreiecksgestalt aufgetragen werden, dem sogenannten Pascalschen Dreieck, benannt nach Blaise Pascal[3]. Abbildung 1 zeigt einen Ausschnitt dieses Dreiecks für $0 \leq k \leq n \leq 4$. Proposition 4 zeigt, dass sich jeder Eintrag in der Tabelle als Summe der Einträge links und rechts in der Zeile darüber ergibt.

1.8 Die Russellsche Antinomie

Bei Mengenangaben durch Aussonderung ist Vorsicht geboten: zum Beispiel führt der Ausdruck

$$R := \{x \mid x \notin x\}$$

zu einem Widerspruch, der *Russellschen Antinomie*[4]:

- Falls R sich selbst enthält, dann erfüllt $x = R$ die Bedingung $x \notin x$ nicht, also ist R in der Menge $R := \{x \mid x \notin x\}$ nicht enthalten, ein Widerspruch.
- Falls R sich nicht selbst enthält, dann erfüllt $x = R$ die Bedingung $x \notin x$, und damit ist R in der Menge $R := \{x \mid x \notin x\}$ enthalten, ebenfalls ein Widerspruch.

Berhard Ganter schreibt dazu im Vorläuferskript: *„Die Möglichkeit von inneren Widersprüchen in der Mathematik war ein Schock! (...) In der ersten Hälfte des 20. Jahrhunderts gab es deshalb eine tiefe Auseinandersetzung mit den Grundlagen der Mathematik. Um der Russellschen Antinomie zu entgehen, durfte man nicht mehr völlig beliebige Mengenkonstruktionen zulassen. (...) Das erforderte genaue Regeln, welche Mengen in der Mathematik erlaubt sind und welche nicht. Dies führte zur heute üblichen* axiomatischen Mengenlehre*”.*

[3] Blaise Pascal, geboren am 19. Juni 1623 in Clermont-Ferrand; gestorben am 19. August 1662 in Paris.

[4] Bertrand Arthur William Russell, 3. Earl Russell, geboren am 18. Mai 1872 bei Trellech, Monmouthshire; gestorben am 2. Februar 1970 in Penrhyndeudraeth, Gwynedd.

1.9 Die Axiome von Zermelo-Fraenkel

Die *Zermelo-Fraenkel-Mengenlehre* ist eine weit verbreitete axiomatische Mengenlehre, und heute Grundlage fast aller Gebiete der Mathematik. Die ursprünglich natürlichsprachlichen Mengenaxiome von Zermelo[5] und Fraenkel[6] wurden unter dem Einfluss von *Hilbert's Programm*[7] später formalisiert. Die Axiome der Zermelo-Fraenkel-Mengenlehre ohne Auswahlaxiom werden durch ZF abgekürzt, mit Auswahlaxiom durch ZFC (wobei das C für *choice*, also Auswahl, steht; siehe Kapitel 2.4).

1. *Leere Menge*: Es gibt eine leere Menge.

2. *Extensionalität*: Wenn zwei Mengen die gleichen Elemente haben, dann sind sie gleich.

3. *Paarmenge*: Für alle Mengen A und B gibt es eine Menge $\{A, B\}$ mit der Eigenschaft, dass $C \in \{A, B\}$ genau dann wenn $C = A$ oder $C = B$.

4. *Vereinigung*: Für alle Mengen M existiert eine Menge, die gleich der Vereinigung aller Mengen in M ist.

5. *Unendliche Mengen*: Es gibt eine Menge M, die die leere Menge und die Menge $\{e\}$ für jedes $e \in M$ enthält.

6. *Potenzmengen*: Für jede Menge M gibt es eine Menge, die genau alle Teilmengen von M enthält.

7. *Ersetzungsschema*: Informell: Bilder von Mengen unter definierbaren Funktionen sind selbst wieder Mengen; eine Formalisierung des Funktionsbegriffs folgt in Kapitel 2.

8. *Fundierung*: Jede Menge $M \neq \emptyset$ enthält ein Element e, so dass $e \cap M = \emptyset$.
   Insbesondere: Mengen enthalten sich nicht selbst, denn falls $a = \{a, b, \dots\}$, dann gilt für jedes Element e der Menge $M := \{a\}$, dass
   $$e \cap M = a \cap \{a\} = \{a, b, \dots\} \cap \{a\} = \{a\} \neq \emptyset.$$

Es wird davon ausgegangen, dass ZF und ZFC widerspruchsfrei sind. Allerdings hat Kurt Gödel[8] gezeigt (in ZFC), dass wenn ZFC widerspruchsfrei ist, die Widerspruchsfreiheit von ZFC nicht in ZFC gezeigt werden kann. Doch das sprengt den Rahmen dieser Vorlesung, und wir verweisen auf Logikvorlesungen.

[5] Ernst Friedrich Ferdinand Zermelo, geboren am 27. Juli 1871 in Berlin; gestorben am 21. Mai 1953 in Freiburg im Breisgau.

[6] Abraham Halevi Fraenkel, geboren am 17. Februar 1891 in München; gestorben am 15. Oktober 1965 in Jerusalem.

[7] David Hilbert, geboren am 23. Januar 1862 in Königsberg; gestorben am 14. Februar 1943 in Göttingen.

[8] Kurt Gödel, geboren am 28. April 1906 in Brünn; gestorben am 14. Januar 1978 in Princeton.

Übungen.

6. Finden Sie einen algebraischen Beweis für Proposition 4.

7. Es seien $A_1, A_2, \ldots$ alle zwei-elementigen Teilmengen der Menge $\{1, 2, \ldots, 10\}$. Wie viele verschiedene Mengen können durch wiederholte Anwendung der Mengenoperatoren $\cap$ und $\cup$ aus $A_1, A_2, \ldots$ gewonnen werden?

8. Entwerfe Mengen A_1, A_2, A_3, A_4, so dass die Anzahl n aller Mengen, die durch wiederholte Anwendung von $\cap$, $\cup$, und $\setminus$ ausgehend von A_1, A_2, A_3, A_4 gebildet werden können, möglichst groß ist. Wie groß kann n werden? **Hinweis.** Beantworten Sie die Frage zuerst für drei Mengen A_1, A_2, A_3.

9. Sei $n \in \mathbb{N}$ eine natürliche Zahl und sei $A = \{1, \ldots, n\}$. Gesucht sind Teilmengen $S_1, \ldots, S_n$ und $T_1, \ldots, T_n$ von $A \times A$ mit folgenden Eigenschaften:

   - $|S_1| = |S_2| = \cdots = |S_n| = |T_1| = |T_2| = \cdots = |T_n| = n$.
   - $S_1 \cup \cdots \cup S_n = T_1 \cup \cdots \cup T_n = A \times A$.
   - $|S_i \cap T_j| = 1$ für alle $i, j \in \{1, \ldots, n\}$.
   - Falls $(a, b), (c, d) \in S_i$ dann gilt $a \neq c$ und $b \neq d$ oder $a = c$ und $b = d$.
   - Falls $(a, b), (c, d) \in T_i$ dann gilt $a \neq c$ und $b \neq d$ oder $a = c$ und $b = d$.

   Zeigen Sie, dass es solche Mengen für $n = 3$ gibt, nicht aber für n=2. Gibt es solche Mengen für $n = 4$? Für $n = 5$?

10. Zeigen Sie mit den Axiomen in ZF, dass es keine Mengen x_1, x_2, x_3 gibt mit $x_1 \in x_2$, $x_2 \in x_3$, und $x_3 \in x_1$.

2 Abbildungen

Seien A und B zwei Mengen. Eine *Abbildung* (oder *Funktion*) *von A nach B* weist jedem Element von A genau ein Element aus B zu. Formal ist eine Funktion f von A nach B ein Paar (G_f, B), wobei $G_f \subseteq A \times B$ eine Relation ist, die folgende Eigenschaft erfüllt: zu jedem $a \in A$ gibt es *genau ein* $b \in B$ so dass $(a, b) \in G_f$. Die Relation G_f wird auch der *Graph von f* genannt.

2.1 Notation

Wenn f eine Funktion von A nach B ist, so wird die Schreibweise $f\colon A \to B$ verwendet; A heißt *Definitionsbereich* und B *Zielbereich* der Funktion. Statt $(a, b) \in G_f$ wird meist $f(a) = b$ geschrieben, und gesagt, dass b das *Bild* von a unter f ist. Eine weitere übliche Schreibweise, um Funktionen anzugeben, sind Ausdücke der Form '$a \mapsto f(a)$'.

Beispiel 2. *Wir schreiben $(x, y) \mapsto x + y$ für die Funktion $f\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ die (x, y) abbildet auf $x + y$. In solchen Fällen wird auch $f(x, y)$ anstatt $f((x, y))$ geschrieben (weil es kürzer, schöner, und in der Regel unmissverständlich ist).*

Funktionen von A^n nach A werden auch *(n-stellige) Operationen auf A* genannt. Die im Beispiel definierte Additionsfunktion ist eine zweistellige Operation auf $\mathbb{N}$.

Das *Bild von A unter f* ist die Menge *aller* Bilder von Elementen von A unter f. Wir schreiben $f[A]$ für das Bild von A unter f; allerdings verwenden viele Autorinnen und Autoren auch die Notation $f(A)$. Wir schreiben B^A für die Menge aller Funktionen von A nach B.

Proposition 5. *Es gilt $|B|^{|A|} = |B^A|$.*

Falls $A = \emptyset$ dann ist $|B^A| = 1$. Ansonsten, wenn $A \neq \emptyset$, dann ist $|B^A| = 0$ genau dann wenn $B = \emptyset$. Eine Funktion f heißt

- *injektiv* falls für alle $a_1, a_2 \in A$ mit $f(a_1) = f(a_2)$ gilt dass $a_1 = a_2$. In anderen Worten, keine zwei verschiedenen Elemente von A haben das gleiche Bild unter f;
- *surjektiv* falls $f[A] = B$ ist. In anderen Worten, für jedes $b \in B$ gibt es ein $a \in A$ mit $f(a) = b$;
- *bijektiv* falls sie sowohl injektiv also auch surjektiv ist.

Proposition 6. *Sei A eine endliche Menge, und $f\colon A \to A$ eine Funktion. Dann sind äquivalent:*

1. *f ist injektiv;*
2. *f ist surjektiv;*

3. f is bijektiv.

Wir vertagen den Beweis auf später in der Vorlesung, wenn wir das Beweisprinzip der vollständigen Induktion kennengelernt haben (Übung 18). Proposition 6 ist für unendliche Mengen falsch. Zum Beispiel ist die Abbildung $f\colon \mathbb{N} \to \mathbb{N}$, die für alle $x \in \mathbb{N}$ gegeben ist durch $f(x) = x+1$, injektiv, aber nicht surjektiv. Und die Abbildung $g\colon \mathbb{N} \to \mathbb{N}$, die gegeben ist durch $f(x) = x/2$ falls $x \in \mathbb{N}$ gerade, und $f(x) = (x-1)/2$ falls $x \in \mathbb{N}$ ungerade, ist surjektiv, aber nicht injektiv.

Sei $f\colon A \to B$ eine Funktion, und $g\colon A' \to B$ so dass $A' \subseteq A$, und $g(a) = f(a)$ für alle $a \in A$. Dann heißt g die *Einschränkung* von f auf A'.

Ist $f\colon A \to B$ injektiv, dann definieren wir die *Umkehrabbildung* $f^{-1}\colon f[A] \to A$ wie folgt: es gelte $f^{-1}(b) = a$ genau dann wenn $f(a) = b$. (Für nicht injektives f wird bisweilen $f^{-1}\colon B \to \mathcal{P}(A)$ durch $f^{-1}(b) = \{a \in A \mid f(a) = b\}$ definiert.)

Sind $f\colon A \to B$ und $g\colon B \to C$ Abbildungen, so ist die *Hintereinanderausführung* (auch *Komposition*) $g \circ f$ von f und g durch $g \circ f\colon A \to C$ mit $(g \circ f)(a) := g(f(a))$ für alle $a \in A$ definiert. Falls $A = B$, so definieren wir $f^1 := f$, $f^2 := f \circ f$, und induktiv $f^{n+1} := f^n \circ f$. Die Funktion $\mathrm{id}_A\colon A \to A$, die definiert ist durch $\mathrm{id}_A(a) := a$ für alle $a \in A$, heißt die *Identitätsfunktion* oder kurz *Identität* auf der Menge A. Für alle $f\colon A \to A$ gilt klarerweise $\mathrm{id}_A \circ f = f = f \circ \mathrm{id}_A$.

2.2 Größenvergleich von Mengen

Wir haben bereits die Schreibweise $|A|$ für die Kardinalität von Mengen A verwendet, allerdings nur für endliche Mengen A. Injektionen und Bijektionen können verwendet werden, um auch unendliche Mengen bezüglich ihrer Elementanzahl zu vergleichen. Denn ist $f\colon A \to B$ eine injektive Funktion, dann hat die Menge B mindestens so viele Elemente wie A. Und ist $f\colon A \to B$ bijektiv, dann hat B genau so viele Elemente wie A.

Definition 7. *Für (endliche wie unendliche) Mengen A, B definieren wir $|A| \leq |B|$ falls es eine Injektion $f\colon A \to B$ gibt, und wir definieren $|A| = |B|$ falls es eine Bijektion $f\colon A \to B$ gibt.*

Beispiel 3. *Es gilt $|\mathbb{Z}| = |\mathbb{N}|$, denn die Funktion $f\colon \mathbb{N} \to \mathbb{Z}$ gegeben durch $f(x) := x/2$ für x gerade, und $f(x) := -(x+1)/2$ sonst, ist bijektiv.*

Beispiel 4. *Es gilt $|\mathbb{N} \times \mathbb{N}| = |\mathbb{N}|$, denn die Funktion $f\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ gegeben durch $f(x, y) := ((x+y)^2 + 3x + y)/2$ ist bijektiv. Insbesondere gilt $|\mathbb{Q}| = |\mathbb{Z}| = |\mathbb{N}|$. Siehe Abbildung 2.*

2.3 Der Satz von Cantor-Schröder-Bernstein

Unsere Notation suggeriert bereits, dass $|A| \leq |B|$ und $|B| \leq |A|$ implizieren, dass $|A| = |B|$ gilt; dies erfordert aber einen Beweis!

	0	1	2	3	4	y
0	0	1	3	6	10	...
1	2	4	7	11	...	...
2	5	8	12	...	...	...
3	9	13	...	...	...	...
4	14	...	...	...	...	...
x	...	...	...	...	...	...

Abbildung 2: Illustration der Funktion $f(x, y) := ((x + y)^2 + 3x + y)/2$.

Satz 8 (Satz von Cantor[9]-Schröder[10]-Bernstein[11])**.** *Seien $f\colon A \to B$ und $g\colon B \to A$ Injektionen. Dann existiert auch eine Bijektion zwischen A und B.*

Beweis. Es genügt, den Fall zu betrachten, dass $A \subseteq B$ und dass f die identische Abbildung ist. Definiere $C := \{g^n(x) \mid n \in \mathbb{N}, x \in B \setminus A\}$. Es gilt $B \setminus C \subseteq A$ da $g^0(B \setminus A) = B \setminus A$. Sei $h\colon B \to A$ gegeben durch $h(x) := g(x) \in C$ falls $x \in C$ und $h(x) := x$ falls $x \in B \setminus C$. Die Abbildung h ist injektiv: falls $h(x) = h(y) \in C$, dann gilt $x = y \in C$ wegen der Injektivität von g, und falls $h(x) = h(y) \in B \setminus C$ dann gilt $x = h(x) = h(y) = y$. Die Abbildung h ist auch surjektiv: für jedes $x \in A \cap C$ gibt es ein $y \in C$ mit $x = g(y)$ und für jedes $x \in A \setminus C$ gilt $x = h(x)$. $\square$

2.4 Das Auswahlaxiom

Im vorigen Kapitel haben wir die Existenz von Injektionen und Bijektionen verwendet, um Mengen bezüglich ihrer Größe zu vergleichen. Was lässt sich in diesem Zusammenhang über die Existenz von *Surjektionen* sagen?

Falls $f\colon A \to B$ eine Injektion ist, dann gibt es sicherlich eine Surjektion von $f[A]$ nach A, nämlich die bereits eingeführte Umkehrabbildung f^{-1} von f. Diese Umkehrabbildung können wir beliebig auf Elemente aus $B \setminus f[A]$ fortsetzen (es sei denn, A ist leer; aber das ist natürlich ein uninteressanter Fall). Eine Fortsetzung erhalten wir dann beispielsweise, indem wir alle Elemente aus $B \setminus f[A]$ auf dasselbe Element aus A abbilden. Falls es also eine Injektion von A nach B gibt, so gibt es sicherlich eine Surjektion von B nach A.

Sei nun $f\colon A \to B$ eine Surjektion. Falls A und B endlich sind, so gibt es auch eine Injektion von B nach A: denn für jedes $b \in B$ gibt es ein $a \in A$ so dass $f(a) = b$, und wir

[9] Georg Cantor, geboren am 3. März 1845 in Sankt Petersburg; gestorben am 6. Januar 1918 in Halle an der Saale.

[10] Ernst Friedrich Wilhelm Karl Schröder, geboren am 25. November 1841 in Mannheim; gestorben am 16. Juni 1902 in Karlsruhe.

[11] Felix Bernstein, geboren am 24. Februar 1878 in Halle an der Saale; gestorben am 3. Dezember 1956 in Zürich.

definieren $g(b) = a$. Wenn A und B unendlich sind, so stellt sich die Frage, ob eine solche Funktion f überhaupt existiert.

Es entspricht aber der mathematischen Praxis, die Existenz solcher Funktionen anzunehmen. Und damit kommen wir zum *Auswahlaxiom*, welches wir in Kapitel 1.9 bereits erwähnt, aber noch nicht eingeführt haben. Das Auswahlaxiom (abgekürzt AC für englisch *Axiom of choice*) hat viele verschiedene aber äquivalente Formulierungen. Eine davon ist die folgende.

(AC) Falls $f\colon A \to B$ eine Surjektion ist, so gibt es auch eine Injektion $g\colon B \to A$ mit $f \circ g = \mathrm{id}_B$.

Und tatsächlich ist bekannt, dass die Existenz solcher Funktionen im allgemeinen in ZF nicht gezeigt werden kann! ZF zusammen mit dem Auswahlaxiom wird mit ZFC bezeichnet.

2.5 Die Kontinuumshypothese

Wir schreiben $|A| < |B|$ falls $|A| \leq |B|$ gilt, aber nicht $|A| = |B|$, und sagen dass B *strikt mächtiger* ist als A.

Satz 9 (Satz von Cantor)**.** *Für alle Mengen A gilt $|A| < |\mathcal{P}(A)|$, das heißt, die Potenzmenge einer beliebigen Menge ist strikt mächtiger als die Menge selbst.*

Beweis. Die Funktion $f\colon A \to \mathcal{P}(A)$, gegeben durch $f(a) := \{a\}$ für alle $a \in A$, ist injektiv, und daher gilt $|A| \leq |\mathcal{P}(A)|$.

Es bleibt also zu zeigen, dass es keine Bijektion zwischen A und $\mathcal{P}(A)$ gibt. Wir führen einen Widerspruchsbeweis, und nehmen an, es gäbe so eine Bijektion $f\colon A \to \mathcal{P}(A)$. Sei

$$M := \{x \in A \mid x \notin f(x)\}\,.$$

Da $M \in \mathcal{P}(A)$ und weil f surjektiv ist, gibt es ein $a \in A$ mit $f(a) = M$. Dann gilt aber dass $a \notin M = f(a)$ nach der Definition von M genau dann, wenn $a \in M$. Ein Widerspruch. $\square$

Es stellt sich nun die Frage, ob es Mengen M gibt mit $|\mathbb{N}| < |M| < |\mathcal{P}(\mathbb{N})|$. An der Stelle sei erwähnt, dass $|\mathcal{P}(\mathbb{N})| = |\mathbb{R}|$, die Kardinalität der reellen Zahlen (dem *Kontinuum*). Die Hypothese, dass es solche Mengen M nicht gibt, wird daher auch *die Kontinuumshypothese* genannt, und wird mit CH abgekürzt. Paul Cohen[12] hat 1963 gezeigt, dass CH in ZFC weder bewiesen noch widerlegt werden kann. Die Kontinuumshypothese ist damit formal *unabhängig* von ZFC.

[12] Paul Joseph Cohen, geboren am 2. April 1934 in Long Branch, New Jersey; gestorben am 23. März in Stanford, Kalifornien.

2.6 Permutationen

Eine *Permutation* einer Menge X ist eine Bijektion $\pi\colon X \to X$. Im Folgenden sei $X = \{1, 2, \ldots, n\}$. Eine Permutation π wird oft in der folgenden Form angegeben

$$\begin{pmatrix} 1 & 2 & 3 & \cdots & n \\ \pi(1) & \pi(2) & \pi(3) & \cdots & \pi(n) \end{pmatrix}$$

Die Zahlen in der ersten Reihe können sogar in beliebiger Reihenfolge sein, solange $\pi(x)$ direkt unter x steht.

Proposition 10. *Es gibt $n!$ Permutationen der Menge $X = \{1, \ldots, n\}$.*

Beweis. Es gibt n Möglichkeiten für $\pi(1)$; für $\pi(2)$ gibt es dann nur noch $n-1$ Möglichkeiten, da wir nur noch aus $\{1, \ldots, n\} \setminus \{\pi(1)\}$ auswählen können. Für $\pi(i)$ gibt es entsprechend $n - i + 1$ Möglichkeiten, und das macht insgesamt $n \cdot (n-1) \cdot (\cdots) \cdot 1 = n!$. □

Komposition von Permutationen

Wenn π_1 und π_2 Permutationen einer Menge X sind, dann können diese als Funktionen komponiert werden (siehe Abschnitt 2.1), und erhält wieder eine Permutation von X. Beispielsweise gelten

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 3 & 4 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 3 & 2 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 1 & 4 \end{pmatrix}$$

und

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 1 & 4 \end{pmatrix}^3 = \mathrm{id}_{\{1,2,3,4\}} \ .$$

Zyklenschreibweise

Es gibt noch eine zweite wichtige, kompaktere Schreibweise für Permutationen, die *Zyklenschreibweise*, die wir nun einführen. Eine *zyklische Permutation* ist eine Permutation mit der Eigenschaft dass die Elemente von X aufgezählt werden können mit $x_1, x_2, \ldots, x_n$ so dass $\pi(x_1) = x_2$, $\pi(x_2) = x_3$, $\ldots$, $\pi(x_{n-1}) = x_n$, $\pi(x_n) = x_1$. Mit anderen Worten: $\pi^1(1), \pi^2(1), \pi^3(1), \ldots, \pi^n(1)$ durchläuft alle Elemente von X. Für solche Permutationen wird die Notation $(x_1, x_2, \ldots, x_n)$ verwendet. Diese Schreibweise ist nicht eindeutig: $(x_2, \ldots, x_n, x_1)$ steht für die gleiche zyklische Permutation.

Beispiel 5. *Beispiele von zyklischen Permutation der Menge $X = \{1, \ldots, 5\}$ sind etwa $\left(\begin{smallmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 1 \end{smallmatrix}\right)$ und $\left(\begin{smallmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 3 & 5 & 2 & 1 \end{smallmatrix}\right)$. In der neuen Schreibweise lesen sich diese Permutationen wie $(2, 3, 4, 5, 1)$ und $(4, 2, 3, 5, 1)$. Das kleinste Beispiel einer nicht-zyklischen Permutation ist die Identitätsfunktion auf einer Menge X mit zwei Elementen.*

Wir erweitern die Zyklenschreibweise auf nicht-zyklische Permutationen π wie folgt. Wir beginnen mit einem beliebigen Element $x_0 \in X$, und berechnen $x_1 := \pi^1(x_0), x_2 := \pi^2(x_0), \ldots$ und so fort, bis wir ein Element zum zweiten Mal erreichen; dies muss notwendigerweise das Element x_0 sein. Wir schreiben $(x_1, x_2, \ldots, x_0)$, und nennen $(x_1, x_2, \ldots, x_0)$

einen *Zyklus* von π. Wenn π nicht zyklisch ist, so gibt es ein Element $y_0 \in X$ das wir auf diese Weise nicht erreicht haben. Wir verfahren mit diesem Element genauso, und berechnen $y_1 := \pi^1(y_0), y_2 := \pi^2(y_0), \ldots$ bis wir wieder y_0 erreichen. Dann ist $(y_1, y_2, \ldots, y_0)$ ein weiterer Zyklus von π, und wir schreiben ihn hinter den ersten Zyklus von π. Die Mengen $\{x_1, x_2, \ldots, x_0\}$ und $\{y_1, y_2, \ldots, y_0\}$ sind disjunkt. So fahren wir fort, bis alle Elemente von M in (genau) einem Zyklus auftauchen. In dieser Schreibweise spielt die Reihenfolge der Zyklen keine Rolle; ausserdem kann ein Zyklus $(x_1, x_2, \ldots, x_n, x_0)$ von π gleichwertig durch $(x_2, \ldots, x_n, x_0, x_1)$ ersetzt werden. In Fällen, wo dies zu keinen Mehrdeutigkeiten führen kann (wie zum Beispiel wenn $X \subseteq \{1, 2, \ldots, 9\}$), werden die trennenden Kommata in den Zyklen auch oft weggelassen.

Beispiel 6. *Die Permutation* $\left(\begin{smallmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 1 & 6 & 3 & 5 & 7 & 4 \end{smallmatrix}\right)$ *hat die Zyklenschreibweise* $(12)(3674)(5)$. *Aber auch die Ausdrücke* $(12)(6743)(5)$ *und* $(5)(3674)(21)$ *stehen für die gleiche Permutation. Insgesamt haben wir* 48 *verschiedene Schreibweisen für diese Permutation: es gibt* $3! = 6$ *verschiedene Reihenfolgen der Zyklen, und* $4 \cdot 2 \cdot 1$ *verschiedene Möglichkeiten, Startpunkte für die Zyklen auszuwählen.*

Ein *Fixpunkt* einer Permutation π ist ein Element $x \in X$ mit $\pi(x) = x$. Fixpunkte bilden in der eben eingeführten Schreibweise Zyklen der Länge eins, und werden oft weggelassen, wenn dadurch keine Verwirrung entsteht.

Transpositionen

Eine *Transposition* ist eine Permutation, die aus einem einzigen Zyklus der Länge 2 und beliebig vielen Fixpunkten besteht.

Proposition 11. *Jede Permutation der Menge* $X = \{1, \ldots, n\}$ *kann als eine Komposition der Transpositionen* $(1, 2)$, $(2, 3)$, $\ldots$, $(n-1, n)$ *geschrieben werden.*

Beweis. Eine beliebige Transposition (x, y) mit $x < y$ erhalten wir durch

$$(x, x+1) \circ (x+1, x+2) \circ \cdots \circ (y-1, y) \circ (y-2, y-1) \circ \cdots \circ (x, x+1) \ .$$

Eine zyklische Permutation $(x_1 x_2 x_3 \ldots x_n)$ erhalten wir durch

$$(x_1 x_2) \circ (x_2 x_3) \circ \cdots \circ (x_{n-1}, x_n) \ .$$

Und schließlich erhalten wir eine beliebige Permutation als Komposition der Ausdrücke für die Zyklen. □

Wie viele Permutationen mit n Elementen gibt es? Wir kennen eine Antwort bereits von Proposition 10, nämlich $n!$, aber wie schnell wächst die Fakultätsfunktion? Dazu gibt es eine erstaunlich gute und extrem nützliche Abschätzung, die sogenannte *Stirlingsche Formel*.

Satz 12 (Stirlingsche[13] Formel)**.** *Für alle $n \in \mathbb{N}$ gilt*

$$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n < n! < \sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\frac{1}{12n}} \ .$$

Um Funktionen bezüglich ihres asymptotischen Wachstums zu vergleichen, gibt es die folgende Notation. Wenn $n \in \mathbb{N}$ und $f, g \colon \mathbb{N} \to \mathbb{R}$, dann schreiben wir $f \sim g$ (und sagen dass f und g *asymptotisch gleich* sind) wenn es für alle $\varepsilon > 0$ ein $n_0 \in \mathbb{N}$ gibt, so dass für alle $n \in \mathbb{N}$ mit $n > n_0$ gilt dass $|f(n)/g(n) - 1| < \varepsilon$. Häufig wird die Stirlingsche Formel in der folgenden, schwächeren asymptotischen Form verwendet:

$$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \tag{1}$$

Übungen.

11. Es seien A und B endliche Mengen. Wie viele Injektionen von A nach B gibt es?

12. Berechnen Sie, wie viele Stellen 100! besitzt.

13. Leiten Sie (1) aus Satz 12 her.

[13] James Stirling; geboren im Mai 1692 in Garden bei Stirling; gestorben am 5. Dezember 1770 in Edinburgh.

3 Boolsche Funktionen und Aussagenlogik

3.1 Boolsche Funktionen

Sei n eine natürliche Zahl größer 0, und sei A eine Menge. Wir definieren A^n durch

$$A^n := \underbrace{A \times (A \times \cdots (A \times A) \cdots)}_{n \text{ Auftreten von } A}$$

Formal handelt es sich hier um eine *induktive Definition* (mehr zu Induktion im Kapitel 4 zu den natürlichen Zahlen): wir definieren

- $A^1 := A$, und
- $A^{n+1} := A \times A^n$ wobei wir annehmen, dass wir A^n bereits definiert haben.

Die Elemente von A^n heißen *(n-) Tupel (über A)* und wir schreiben $(a_1, a_2, \ldots, a_{n-1}, a_n)$ anstatt des n-Tupels $(a_1, (a_2, \ldots (a_{n-1}, a_n) \ldots))$. Eine *n-stellige Operation* auf einer *Trägermenge* A ist eine Abbildung $f\colon A^n \to A$ die jedem n-Tupel $(a_1, \ldots, a_n)$ von Elementen aus A ein Element aus A zuordnet (die Elemente $a_1, \ldots, a_n$ heißen dann auch die *Argumente* von f). Die Anzahl der n-stelligen Operationen auf A ist $|A|^{|A|^n}$. Im Fall, dass A zweielementig ist, gibt es also 2^{2^n} n-stellige Operationen auf A. Operationen auf einer zweielementigen Menge werden auch *boolschen Funktionen*[14] genannt. Es ist unwichtig, welche zweielementige Trägermenge betrachtet wird. Gängige Bezeichnungen sind $A = \{W, F\}$ (für *wahr* und *falsch*), $A = \{T, F\}$ (für englisch *true* und *false*), $A = \{\bot, \top\}$ und $A = \{0, 1\}$. Die typischen Entsprechungen sind: $W, T, \top, 1$ zum einen, und $F, F, \bot, 0$ zum anderen.

Besondere Operationen

Konstante Operationen sind solche, für die es ein Element $c \in A$ gibt mit $f(a_1, \ldots, a_n) = c$ für alle $a_1, \ldots, a_n \in A$. Konstante Operationen werden oft einfach mit dem Namen der Konstanten c bezeichnet, und die Angabe der Stelligkeit meist weggelassen. Eine *Projektion* p_i^n, die jedem n-Tupel seine i-te Komponente zuordnen, also $p_i^n(a_1, \ldots, a_n) := a_i$ für alle $a_1, \ldots, a_n \in A$. Wir wissen bereits, dass es auf der Menge $\{0, 1\}$ genau 2 nullstellige, 4 einstellige, und 16 zweistellige Operationen gibt. Die beiden nullstelligen, und zwei der einstelligen sind konstant. Es gibt eine einstellige Projektion, das ist die Identitätsfunktion. Es bleibt also nur noch eine weitere einstellige Operation, und das ist die Operation $x \mapsto 1 - x$. Diese Operation wird oft mit $\neg$ bezeichnet, und *Negation* oder *Nicht-Funktion* genannt.

[14] George Boole; geboren am 2. November 1815 in Lincoln, England; gestorben am 8. Dezember 1864 in Ballintemple, Irland.

$\wedge$	0	1
0	0	0
1	0	1

$\vee$	0	1
0	0	1
1	1	1

Abbildung 3: Die Verknüpfungstafeln der Und-Funktion $\wedge$ und der Oder-Funktion $\vee$.

Und und Oder

Unter den zweistelligen boolschen Operationen werden wir zwei ganz besonders betrachten, und das ist die *Und-Funktion* $\wedge$ (auch *Konjunktion*) und die *Oder-Funktion* $\vee$ (auch *Disjunktion*). Die Verküpfungstafeln dieser beiden Funktionen finden sich in Abbildung 3.

Es ist üblich, den Ausdruck $x \wedge y$ anstatt der Funktionsschreibweise $\wedge(x, y)$ zu verwenden. Diese Notation wird *Infix-Notation* genannt. Analog schreiben wir $x \vee y$ anstatt $\vee(x, y)$, und $\neg x$ anstatt $\neg(x)$. Wir stellen fest: Konjunktion und Disjunktion sind

- *idempotent*, denn $x \wedge x = x \vee x = x$ für alle $x \in \{0, 1\}$;
- *kommutativ*, denn $x \wedge y = y \wedge x$ und $x \vee y = y \vee x$ für alle $x, y \in \{0, 1\}$;
- *assoziativ*, denn es gilt $x \wedge (y \wedge z) = (x \wedge y) \wedge z$ und $x \vee (y \vee z) = (x \vee y) \vee z$ und für alle $x, y, z \in \{0, 1\}$;
- Konjunktion ist *distributiv* über Disjunktion, denn es gilt $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ für alle $x, y, z \in \{0, 1\}$.
- Disjunktion ist *distributiv* über Konjunktion, denn $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ für alle $x, y, z \in \{0, 1\}$.

Der Leser sei an der Stelle aufgefordert, zum Abschnitt 1.3 zurückzublättern. Wegen der Assoziativität der Konjunktion können wir bei Ausdrücken der Gestalt

$$x_1 \wedge (x_1 \wedge (\cdots \wedge (x_{n-1} \wedge x_n) \ldots))$$

auf die Klammerung verzichten (weil die Klammerung auf das Ergebnis keinen Einfluss hat), und schreiben dann auch oft abkürzend

$$\bigwedge_{i \in \{1,\ldots,n\}} x_i \ .$$

Falls $n = 0$, also für die 'leere Konjunktion', soll dieser Ausdruck auch definiert sein, und zwar als 1. Analog definieren wir

$$\bigvee_{i \in \{1,\ldots,n\}} x_i \ .$$

Falls $n = 0$, also für die 'leere Disjunktion', soll dieser Ausdruck auch definiert sein, und zwar als 0.

Bei Ausdrücken mit sowohl $\wedge$ als auch $\vee$ spielt die Klammerung dagegen sehr wohl eine Rolle. Allerdings werden auch hier oft die Klammern weggelassen, da folgende Konvention gilt: $\wedge$ *bindet stärker als* $\vee$. Das bedeutet, dass der Ausdruck $x \wedge y \vee z$ zu lesen ist als $(x \wedge y) \vee z$, und nicht als $x \wedge (y \vee z)$.

Weiterhin erfüllen $\wedge$, $\vee$ und $\neg$ die *De Morganschen Regeln*[15], es gilt nämlich

$$\begin{aligned} & \neg(x \wedge y) = \neg x \vee \neg y \\ \text{und} \qquad & \neg(x \vee y) = \neg x \wedge \neg y\,. \end{aligned}$$

Daraus folgt, dass sich die Und-Funktion durch Komposition mit Hilfe der Oder-Funktion und der Nicht-Funktion darstellen lässt, denn für alle $x, y \in \{0,1\}$ gilt dass

$$x \wedge y = \neg(\neg x \vee \neg y)\ .$$

Umgekehrt lässt sich aus der Und-Funktion und der Nicht-Funktion auch die Oder-Funktion gewinnen, denn es gilt für alle $x, y \in \{0,1\}$ dass $x \vee y = \neg(\neg x \wedge \neg y)$. Wir sagen daher auch, dass sich Und und Oder *dual* zueinander verhalten.

Proposition 13. *Alle boolschen Operationen lassen sich durch Komposition mit Hilfe der Nicht-Funktion und der Und-Funktion darstellen.*

Beweis. Es sei $f\colon \{0,1\}^n \to \{0,1\}$ eine n-stellige boolsche Funktion. Dann gilt für alle $a_1, \ldots, a_n \in \{0,1\}$ dass

$$f(a_1, \ldots, a_n) = \bigvee_{\substack{b_1, \ldots, b_n \in \{0,1\}, \\ f(b_1, \ldots, b_n) = 1}} \left(\bigwedge_{i \in \{1,\ldots,n\} \text{ mit } b_i = 1} a_i \wedge \bigwedge_{i \in \{1,\ldots,n\} \text{ mit } b_i = 0} \neg a_i \right)\ .$$

Zuletzt können wir aus diesem Ausdruck mit der Regel $x \vee y = \neg(\neg x \wedge \neg y)$ auch noch jedes Auftreten der Oder-Operation eliminieren. $\square$

Beispiel 7. *Sei die boolsche Operation* $f\colon \{0,1\}^3 \to \{0,1\}$ *gegeben durch*

[15] Augustus De Morgan; geboren am 27. Juni 1806 in Madurai, Indien; gestorben am 18. März 1871 in London.

x, y, z	$f(x, y, z)$
0,0,0	*0*
0,0,1	*1*
0,1,0	*0*
0,1,1	*1*
1,0,0	*0*
1,0,1	*0*
1,1,0	*0*
1,1,1	*0*

Dann gilt für alle $x, y, z \in \{0, 1\}$ *dass*

$$f(x, y, z) = (\neg x \wedge \neg y \wedge z) \vee (\neg x \wedge y \wedge z)$$

Andere zweistellige Boolsche Operationen

Konjunktion, Disjunktion und Negation reichen also fürs Rechnen mit 0 und 1 aus. Im täglichen Leben ist es aber praktisch, für bestimmte zweistellige boolsche Operationen spezielle Bezeichnungen zu haben. In allen Fällen verwenden wir Infix-Notation. Die wichtigsten sind:

$\Rightarrow$ Wird *Implikation* genannt, und wird definiert wie folgt: für $x, y \in \{0, 1\}$ ist

$$(x \Rightarrow y) := (\neg x \vee y) \ .$$

$\Leftrightarrow$ *Äquivalenz*. Die Definition ist für $x, y \in \{0, 1\}$

$$(x \Leftrightarrow y) := \big((x \Rightarrow y) \wedge (y \Rightarrow x)\big) \ .$$

xor Das *exklusive Oder*, wird auch mit $\oplus$ bezeichnet. Für $x, y \in \{0, 1\}$ ist

$$x \oplus y := (x \vee y) \wedge \neg(x \wedge y) \ .$$

Diese boolsche Operation wird eine wichtige Rolle im Abschnitt 5 spielen: es handelt sich bei $\oplus$ um die Addition modulo 2.

nand Der *Sheffer-Strich*, oder *nand*, ist die Funktion $(x, y) \mapsto \neg(x \wedge y)$. Das bemerkenswerte an dieser Funktion ist, dass sich mit ihrer Hilfe bereits alle anderen Funktionen durch Komposition herstellen lassen: denn $\neg x = (x \text{ nand } x)$, und $x \wedge y = \neg(x \text{ nand } y)$. Damit folgt diese Behauptung aus Proposition 13.

nor Die *Peirce-Funktion* ist dual zum Sheffer-Strich, und hat ebenfalls die Eigenschaft, dass alle anderen Funktionen damit gebaut werden können.

3.2 Aussagenlogik

Wir haben bisher im Skript häufig den Ausdruck 'Ausdruck' verwendet. Aber was ist das, ein Ausdruck? Was ist eine mathematische *Aussage*? Und was ist eigentlich ein *Beweis*? Häufig werden diese Begriffe in der Mathematik metasprachlich verwendet. Aber im Rahmen der Logik werden diese Begriffe auch formal definiert und verwendet. Wir betrachten hier allerdings nur den speziellen Fall der *Aussagenlogik*. Den allgemeineren und extrem wichtigen Fall der *Prädikatenlogik erster Stufe* werden Sie später noch kennenlernen.

Ein grundlegendes Prinzip in der Logik ist die Trennung zwischen Syntax und Semantik. Die *Syntax* behandelt Symbole und Regeln, wie man die Symbole zu Ausdrücken zusammenfügt. Die *Semantik* behandelt, wie man den Ausdrücken Bedeutung zuweist.

Syntax

Die Symbole der Aussagenlogik sind die Symbole $\wedge, \vee, \neg, \bot, \top$ und Variablensymbole. Häufig verwendete Variablensymbole in der Aussagenlogik sind X, Y, Z oder $X_1, X_2, \ldots$ aber die Wahl der Symbole für die Variablen spielt hier keine Rolle. Die Syntax der Aussagenlogik ist sehr einfach, und rekursiv definiert.

- Variablensymbole sind Ausdrücke.
- $\top$ und $\bot$ sind Ausdrücke.
- Wenn A ein Ausdruck ist, dann ist auch $\neg A$ ein Ausdruck.
- Wenn A_1 und A_2 Ausdrücke sind, dann sind auch $A_1 \wedge A_2$ und $A_1 \vee A_2$ Ausdrücke.

Solche (aussagenlogischen) Ausdrücke werden oft auch *aussagenlogische Formeln* genannt.

Semantik

Es sei A ein Ausdruck mit den Variablen $X_1, \ldots, X_n$. Dann ist eine *Belegung* von $X_1, \ldots, X_n$ eine Funktion $f\colon \{X_1, \ldots, X_n\} \to \{0, 1\}$. Wir definieren nun rekursiv, wann f den Ausdruck A *erfüllt*:

- $\top$ wird von f erfüllt (unabhängig von f).
- Falls A von der Gestalt X_i ist, dann wird A von f genau dann erfüllt, wenn $f(X_i) = 1$.
- Falls A von der Gestalt $\neg A$ ist, dann wird A von f genau dann erfüllt, wenn A von f *nicht* erfüllt wird.
- Falls A von der Gestalt $A_1 \wedge A_2$ ist, dann wird A von f genau dann erfüllt, wenn sowohl A_1 also auch A_2 von f erfüllt werden.

- Falls A von der Gestalt $A_1 \vee A_2$ ist, dann wird A von f genau dann erfüllt, wenn A_1 oder A_2 von f erfüllt werden.

Definition 14 (Erfüllbarkeit und Äquivalenz)**.** *Ein Ausdruck A heißt* erfüllbar *falls er eine erfüllende Belegung besitzt, und* unerfüllbar *(oder* widersprüchlich*) sonst. Zwei Ausdrücke mit den gleichen Variablen heißen* äquivalent *wenn sie die gleichen erfüllenden Belegungen besitzen.*

Beispielsweise sind die beiden Ausdrücke $X \wedge Y$ und $\neg(\neg X \vee \neg Y)$ (syntaktisch) nicht gleich, aber äquivalent im obigen Sinn.

Auch die Symbole $\Rightarrow$ und $\Leftrightarrow$ werden häufig in Ausdrücken verwendet; $A_1 \Rightarrow A_2$ ist als abkürzende Schreibweise für $\neg A_1 \vee A_2$ zu verstehen, und $A_1 \Leftrightarrow A_2$ als abkürzende Schreibweise für

$$(A_1 \Rightarrow A_2) \wedge (A_2 \Rightarrow A_1) \ ,$$

was äquivalent ist zu

$$(A_1 \wedge A_2) \vee (\neg A_1 \wedge \neg A_2) \ .$$

Ein Ausdruck ist eine *Tautologie*, wenn jede Belegung der Variablen eine erfüllende Belegung ist. Beispielsweise ist $X \vee \neg X$ eine Tautologie. Wir bemerken:

- A_1 und A_2 sind genau dann äquivalent, wenn $A_1 \Leftrightarrow A_2$ eine Tautologie ist.
- A ist genau dann unerfüllbar, wenn $\neg A$ eine Tautologie ist.

Definition 15. *Es seien A_1 und A_2 aussagenlogische Ausdrücke mit den Variablen $X_1, \ldots, X_n$. Dann sagen wir, dass A_1 den Ausdruck A_2* impliziert*, falls jede erfüllende Belegung von A_1 auch eine erfüllende Belegung von A_2 ist. Dafür schreiben wir auch $A_1 \models A_2$.*

Wir bemerken, dass $A_1 \models A_2$ genau dann gilt, wenn $A_1 \Rightarrow A_2$ eine Tautologie ist.

Übungen.

14. Zeigen Sie, dass für alle aussagenlogischen Ausdrücke A und B die Ausdrücke $A \Rightarrow B$ und $\neg B \Rightarrow \neg A$ äquivalent sind.[16]

Aussagenlogik und Boolsche Operationen

Jeder aussagenlogische Ausdruck A mit den Variablen $x_1, \ldots, x_n$ entspricht auf natürliche Weise einer boolschen Operation f_A der Stelligkeit n, die $(a_1, \ldots, a_n) \in \{0,1\}^n$ genau dann auf 1 abbildet, wenn die Abbildung $x_i \mapsto a_i$ eine erfüllende Belegung von A ist. Wir sagen auch, dass A die Operation f_A *definiert*.

[16] Der Ausdruck $\neg B \Rightarrow \neg A$ heißt *Kontraposition* des Audrucks $A \Rightarrow B$. Beim Aufschrieb von Beweisen kann bisweilen praktischer sein, die Kontraposition zu beweisen.

Beispiel 8. *Der Ausdruck $x_1 \wedge x_2$ definiert die boolsche Operation der Konjunktion, wie wir sie in Abschnitt 3.1 mit Hilfe einer Tabelle eingeführt haben. Der Ausdruck $\neg x_1 \wedge x_2$ definiert die boolsche Operation $\Rightarrow$, wie wir sie ebenfalls bereits in Abschnitt 3.1 eingeführt haben. Und so weiter.*

Im Beweis von Proposition 13 sehen wir, dass sich alle boolschen Funktionen als Disjunktion von Konjunktionen von Variablen oder negierten Variablen darstellen lassen. Diese spezielle Gestalt von Ausdrücken wird *disjunktive Normalform (DNF)* genannt. Proposition 13 übersetzt sich nun direkt in die Sprache der Aussagenlogik.

Proposition 16. *Jeder aussagenlogische Ausdruck ist äquivalent zu einem in disjunktiver Normalform.*

Ein aussagenlogischer Ausdruck ist in *Konjunktive Normalform (KNF)* falls er eine Konjunktion von Disjunktion von Variablen und negierten Variablen ist (die Definition ist *dual* zu der der Disjunktiven Normalform).

Korollar 17. *Jeder aussagenlogische Ausdruck ist äquivalent zu einem in konjunktiver Normalform.*

Beweisidee. Sei A ein aussagenlogischer Ausdruck. Nach Proposition 16 ist insbesondere $\neg A$ äquivalent zu einem aussagenlogischen Ausdruck B in DNF. Mit Hilfe der De Morganschen Regeln und der Äquivalenz von $\neg(\neg X))$ und X lässt sich $\neg B$ schrittweise in einen äquivalenten Ausdruck in KNF überführen. □

Das Arbeiten mit konjunktiver Normalform hat viele Vorzüge. Es gibt sogar spezielle Terminologie dafür:

- Die Teilausdrücke (von einem Ausdruck A) der Gestalt X oder $\neg X$ (wobei X ein Variablensymbol ist) werden *Literale* (von A) genannt.
- Die *Konjunkte* eines Ausdruck in KNF werden *Klauseln* genannt; man fasst Klauseln als Mengen von Literalen auf (sie sind implizit verodert).

Da man jeden Ausdruck in einen äquivalenten in konjunktiver Normalform (KNF) schreiben kann, beginnen viele Beweise mit der Annahme, dass die Ausdrücke bereits in konjunktiver Normalform vorliegen. Allerdings kann sich in manchen (tatsächlich in *vielen*) Fällen die Länge der Formel beim Übergang von DNF zu KNF exponentiell aufblähen. Dual dazu ist der Übergang von KNF zu DNF im allgemeinen auch teuer.

Wir betrachten zum Beispiel die folgende Formel mit $2n$ Variablen und n Disjunkten, die in DNF vorliegt.

$$(X_1 \wedge Y_1) \vee (X_2 \wedge Y_2) \vee \cdots \vee (X_n \wedge Y_n)$$

Eine Übersetzung nach KNF führt auf eine Formel mit 2^n Konjunkten.

$$(X_1 \vee \cdots \vee X_{n-1} \vee X_n) \wedge (X_1 \vee \cdots \vee X_{n-1} \vee Y_n) \wedge \cdots \wedge (Y_1 \vee \cdots \vee Y_{n-1} \vee Y_n) \quad (2)$$

Diese beiden Ausdrücke sind äquivalent: falls eine Belegung das Disjunkt $X_i \wedge Y_i$ erfüllt, dann erfüllt sie auch jede Klausel von (2), da das jeweils i-te Literal jeder Klausel wahr gemacht wird. Falls es dagegen für jedes $i \in \{1, \ldots, n\}$ ein $Z_i \in \{X_i, Y_i\}$ gibt, so dass Z_i nicht erfüllt wird, dann auch nicht die Klausel $\{Z_1, \ldots, Z_n\}$ von (2). Die Äquivalenz der beiden Ausdrücke folgt auch durch wiederholtes Anwenden der Distributivgesetze; auf diese Weise erhählt man auch einen alternativen Beweis von Korollar 17.

Boolsche Relationen

Eine *boolsche Relation* der Stelligkeit $n \in \mathbb{N}$ ist eine Teilmenge von $\{0,1\}^n$. Jeder boolschen Operation f der Stelligkeit n können wir eine boolsche Relation R_f der Stelligkeit n zuordnen, nämlich die Relation

$$R_f := \{(a_1, \ldots, a_n) \mid f(a_1, \ldots, a_n) = 1\} .$$

Umgekehrt können wir jeder n-stelligen Boolschen Relation R eine n-stellige boolsche Operation zuweisen, nämlich die Operation f_R, für die $f(a_1, \ldots, a_n) = 1$ genau dann gilt, wenn $(a_1, \ldots, a_n) \in R$. Klarerweise gilt

$$f_{R_f} = f \text{ und } R_{f_R} = R.$$

Wenn A ein aussagenlogischer Ausdruck ist, der die Operation f definiert, dann sagen wir auch, dass A die Relation $R := R_f$ definiert. In diesem Fall schreiben wir für R auch R_A.

Beispiel 9. *Der Ausdruck* $\neg X_1 \vee X_2$ *definiert die boolsche Relation* $\{(0,0), (0,1), (1,1)\}$.

Wir bemerken, dass $A_1 \models A_2$ genau dann gilt, wenn $R_{A_1} \subseteq R_{A_2}$.

3.3 Das Erfüllbarkeitsproblem

Ein zentrales Berechnungsproblem ist das folgende:

Gegeben: Ein aussagenlogischer Ausdruck A.
Gefragt: Ist A erfüllbar?

Dieses Berechnungsproblem wird auch das *Erfüllbarkeitsproblem* (der Aussagenlogik) genannt. Ein naiver Ansatz zum Lösen des Erfüllbarkeitsproblems besteht darin, alle Belegungen der Reihe nach zu erzeugen, und für jede Belegung nachzurechnen, ob sie erfüllend ist. Die Rechnung, ob eine Belegung erfüllend ist, lässt sich effizient implementieren. Aber das Durchprobieren aller Belegungen ist schon bei ganz kleinen Anzahlen von Variablen nicht mehr praktikabel: es gibt zu viele davon.

Komplexität

Man geht davon aus, dass es kein Verfahren gibt, welches das Erfüllbarkeitsproblem mit polynomieller Rechenzeit löst. Aber leider ist dafür kein Beweis bekannt.

Das Erfüllbarkeitsproblem gehört zu einem der schwierigsten Probleme seiner Klasse (nämlich der Klasse NP). Genaueres dazu werden Sie in anderen Vorlesungen lernen. Grob gesagt soll das heißen, dass wenn es einen polynomiellen Algorithmus gäbe für das Erfüllbarkeitsproblem, es auch einen Algorithmus gäbe für alle anderen Probleme, bei denen sich eine gegebene Lösung in Polynomialzeit nachrechnen lässt. Von solchen Problemen werden wir in dieser Vorlesung noch einige kennenlernen:

- Das Faktorisieren von Zahlen (Abschnitt 4.4).
- Die Berechnung des diskreten Logarithmus (Abschnitt 6.4).
- Die Berechnung der geheimen Schlüssels beim Diffie-Helman-Merkle Protokoll (Abschnitt 6.4).
- Das Knacken von RSA Verschlüsselung (Abschnitt 6.7).
- Das Berechnungsproblem zu entscheiden, ob ein gegebener Graph mit k Farben gefärbt werden kann (Abschnitt 7.2).
- Das Problem, Hamiltonkreise zu finden (Abschnitt 7.7).
- Das Berechnungsproblem zu entscheiden, ob ein gegebener Graph G für ein ebenfalls gegebenes k eine Knotenüberdeckung der Größe höchstens k besitzt. (Hier ist G nicht notwendigerweise bipartit! Siehe Abschnitt 7.8.)

Für all diese Probleme kennt man keine polynomiellen Algorithmen. Ein polynomieller Algorithmus fürs Erfüllbarkeitsproblem der Aussagenlogik würde für jedes dieser Probleme einen polynomiellen Algorithmus liefern!

Bereits die Einschränkung des Erfüllbarkeitsproblems auf Ausdrücke in KNF ist schwer. Tatsächlich würde ein polynomieller Algorithmus für Aussagen in KFN bereits einen polynomiellen Algorithmus für beliebige Aussagen liefern. Dazu betrachten wir folgendes Beispiel.

Beispiel 10. *Es seien A_1, A_2, A_3, A_4 beliebige aussagenlogische Ausdrücke, und sei*

$$A := (A_1 \wedge A_2) \vee (A_3 \wedge A_4).$$

Sei X nun eine neue aussagenlogische Variable und definiere

$$A' := (A_1 \vee X) \wedge (A_2 \vee X) \wedge (A_3 \vee \neg X) \wedge (A_4 \vee \neg X).$$

Die Ausdrücke A und A' sind nicht *äquivalent, da sie nicht die gleichen Variablen besitzen. Allerdings sind sie* erfüllbarkeitsäquivalent*, da A genau dann erfüllbar ist, wenn A' erfüllbar ist.*

Mit Hilfe der Idee aus dem vorigen Beispiel lässt sich zeigen, dass sich jeder Ausdruck schrittweise in einen erüllbarkeitsäquivalenten, unwesentlich größeren Ausdruck umformen lässt.

3.4 Horn-SAT

Es gibt eine wichtige Einschränkung des Erfüllbarkeitsproblems, für die man effiziente Algorithmen kennt. Wieder liegt die Eingabe in konjunktiver Normalform vor, diesmal aber fordern wir, dass in jeder Klausel höchstens ein positives Literal auftritt. Solche aussagenlogischen Ausdrücke heißen auch *Horn-Formeln*. Das Berechnungsproblem, von einem solchen aussagenlogischen Ausdruck festzustellen, ob er erfüllbar ist, heißt *Horn-SAT*[17].

Eine Methode, um Horn-SAT in polynomieller Zeit zu lösen, ist *positive 1-Resolution*. In diesem Verfahren wird zunächst nach Klauseln der Gestalt $\{X\}$ in der Eingabe gesucht, also nach Klauseln, die nur ein einziges Literal enthalten, und zwar ein positives. Wir werden solche Klauseln im folgenden mit *positiven 1-Klauseln* bezeichnen. Wir bemerken, dass jede erfüllende Belegung der Variablen X einer positiven 1-Klausel den Wert 1 zuweisen muss. Der Algorithmus streicht also jedes Auftreten von $\neg X$ in anderen Klauseln (denn solche Literale sind in erfüllenden Belegungen immer falsch), und markiert die Variable X. Wir wiederholen dieses Verfahren so oft, bis wir keine unmarkierten Variablen in positiven 1-Klauseln mehr finden. Aber Achtung: durch das Wegstreichen von Literalen kann es passieren, dass sich größere Klauseln in positive 1-Klauseln verwandeln, und diese müssen natürlich mit betrachtet werden.

Proposition 18. *Eine Horn-Formel ist genau dann unerfüllbar, wenn mit 1-Resolution die leere Klausel hergeleitet wird.*

Beweis. Wenn ein aussagenlogischer Ausdruck eine erfüllende Belegung besitzt, und eine Klausel durch positive 1-Resolution aus dem Ausdruck hergeleitet wird, dann erfüllt die Belegung auch die neue Klausel. Die leere Klausel ist stets unerfüllbar. Wenn also die leere Klausel hergeleitet wird, dann folgt dass bereits der ursprüngliche Ausdruck unerfüllbar war.

Wir behaupten, dass andernfalls die Eingabe erfüllbar ist. Eine erfüllende Belegung kann man gewinnen, indem man alle markierten Variablen auf 1 setzt, und alle unmarkierten auf 0. Dadurch werden alle positiven 1-Klauseln erfüllt, denn die entsprechenden Variablen wurden allesamt auf 1 gesetzt. Alle anderen Klauseln enthalten mindestens ein negatives Literal, werden von dieser Belegung also ebenfalls erfüllt. □

Welche aussagenlogischen Ausdrücke sind äquivalent zu Horn-Formeln? Für kleine Ausdrücke kann man das noch mit Durchprobieren wie in folgendem Beispiel feststellen.

[17] Alfred Horn; geboren am 17. Februar 1918 in Manhattan; gestorben am 16. April 2001 in Pacific Palisades, Kalifornien.

Beispiel 11. *Der aussagenlogische Ausdruck* $(A \vee B) \wedge \neg B$ *ist nicht Horn, aber äquivalent zum Horn Ausdruck* $A \wedge \neg B$*. Auf der anderen Seite gibt es keine Horn-Formel, die zum Ausdruck* $A \vee B$ *äquivalent wäre: denn alle Horn Klauseln mit den Variablen* A *und* B *(davon gibt es nur endlich viele, und man kann sie der Reihe nach durchprobieren) sind tautologisch (wie etwa* $A \vee \neg A$*) oder werden nicht von* $A \vee B$ *impliziert.*

Wir wollen nun eine Methode angeben, wie man boolsche Relationen effizient darauf testen kann, ob sie mit einer Horn-Formel definiert werden können.

Definition 19. *Eine boolsche Relation* $R \subseteq \{0,1\}^n$ *wird von einer boolschen Funktion* f *der Stelligkeit* k erhalten*, falls für alle* $(a_1^1, \ldots, a_n^1), \ldots, (a_1^k, \ldots, a_n^k) \in R$ *gilt, dass* $(f(a_1^1, \ldots, a_1^k), \ldots, f(a_n^1, \ldots, a_n^k)) \in R$*. Ein aussagenlogischer Audruck* A *wird von* f *erhalten, falls* R_A *von* f *erhalten wird.*

Ein aussagenlogischer Ausdruck A in konjunktiver Normalform heißt *reduziert* falls jeder Ausdruck, den man aus A durch Wegstreichen eines Literals gewinnen kann, nicht äquivalent ist zu A.

Lemma 20. *Jeder ausagenlogische Ausdruck* A *ist äquivalent zu einem reduzierten.*

Beweis. Wie wir schon wissen, ist A äquivalent zu einem Ausdruck B in konjunktiver Normalform. Wenn B bereits reduziert ist, bleibt nichts mehr zu zeigen. Ansonsten gibt es ein Literal in B, das man wegstreichen kann, so dass der resultierende Ausdruck B' äquivalent ist zu B. Wir fahren dann mit B' anstatt B fort, und erreichen so nach endlich vielen Schritten einen reduzierten Ausdruck. □

Proposition 21. *Eine boolsche Relation* R *hat genau dann eine Definition durch eine Horn-Formel, wenn* R *erhalten wird von der Operation* $\wedge$*.*

Beweis. Wir zeigen zuerst, dass R von $\wedge$ erhalten wird, falls R durch eine Horn-Formel A definiert wird. Man sieht leicht, dass wenn eine boolsche Operation f die aussagenlogischen Ausdrücke B_1 und B_2 erhält, auch der Ausdruck $B_1 \wedge B_2$ von f erhalten wird. Es genügt also zu zeigen, dass die Operation $\wedge$ Klauseln $x_0 \vee \neg X_1 \vee \cdots \vee \neg X_n$ mit höchstens einem positiven Literal erhält. Sei $X_0 \vee \neg X_1 \vee \cdots \vee \neg X_n$ eine solche Klausel, und seien f_1, f_2 zwei erfüllende Belegungen von dieser Klausel. Wenn die Belegung $f\colon X_i \mapsto f_1(X_i) \wedge f_2(X_i)$ diese Klausel *nicht* erfüllen würde, so müsste gelten $f(X_0) = 0, f(X_1) = 1, \ldots, f(X_n) = 1$. Also müsste gelten $f_1(X_i) = 1$ und $f_2(X_i) = 1$ für alle $i \in \{1, \ldots, n\}$, denn der Funktionswert von $\wedge$ ist nur dann 1, wenn beide Argumente 1 sind. Da sowohl f_1 als auch f_2 erfüllende Belegungen sind, muss gelten $f_1(X_0) = 1$ und $f_2(X_0)$, und damit auch $f(X_0) = f_1(X_0) \wedge f_2(X_0) = 1$. Das steht im Widerspruch zur Annahme, dass $f(X_0) = 0$.

Nun zur umgekehrten Richtung der Aussage. Sei R eine boolsche Relation, die von $\wedge$ erhalten wird. Sei ϕ ein reduzierter aussagenlogischer Ausdruck in CNF, der R definiert. Wir führen einen Widerspruchsbeweis und nehmen an, dass ϕ eine Klausel C enthält mit zwei positive Literalen x und y. Da ϕ reduziert ist, finden wir eine Belegung s_1, die ϕ erfüllt,

so dass gilt $s_1(x) = 1$, und so dass alle anderen Literale von C zu 0 auswerten. Analog gibt es eine erfüllende Belegung s_2 für ϕ so dass $s_2(y) = 1$ und alle anderen Literale von C zu 0 auswerten. Dann erfüllt die Belegung $s_0 \colon x \mapsto (s_1(x) \wedge s_2(y))$ nicht die Klausel C, und daher auch nicht ϕ, im Widerspruch zur Annahme, dass R von $\wedge$ erhalten wird. $\square$

Beispiel 12. *Wir betrachten wieder den aussagenlogische Ausdruck $A \vee B$. Die Relation $R_{A \vee B} = \{(0,1),(1,0),(1,1)\}$ wird nicht erhalten von $\wedge$, da $(f(0,1), f(1,0)) = (0,0) \notin R_A$. Also impliziert Lemma 21 dass $A \vee B$ nicht äquivalent zu einem Horn-Ausdruck ist.*

Übungen.

15. Finden Sie einen Algorithmus mit garantiert polynomieller Laufzeit, der das Erfüllbarkeitsproblem löst im Spezialfall, dass der gegebene Ausdruck in Disjunktiver Normalform vorliegt.

16. Jeder aussagenlogische Ausdruck ist zu einem in DNF äquivalent, und das Erfüllbarkeitsproblem für DNF lässt sich nach vorstehender Übung in polynomieller Zeit lösen. Warum erhalten wir damit noch keinen polynomiellen Algorithmus für das Erfüllbarkeitsproblem?

17. Hat die boolsche Relation $R = \{(0,0,1),(0,1,0),(1,0,0)\}$ eine Definition durch eine Horn-Formel? Falls nein, geben Sie die kleinste boolsche Relation an, die R enthält und durch eine Horn Formel definiert wird. Geben Sie eine solche Horn Formel an.

4 Die natürlichen Zahlen

Es gibt verschiedene Weisen, die natürlichen Zahlen als Mengen aufzufassen. Die Standardkodierung jedoch ist die folgende. Für eine beliebige Menge M definieren wir

$$M^+ := M \cup \{M\} \ .$$

Die Menge M^+ heißt der *Nachfolger* von M, und M heißt *Vorgänger* von M^+. Man betrachtet die Folge

$$\begin{aligned} \emptyset \\ \emptyset^+ &= \{\emptyset\} \\ \emptyset^{++} &= \{\emptyset, \{\emptyset\}\} \\ \emptyset^{+++} &= \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\} \\ \dots \end{aligned}$$

und gibt diesen Mengen abkürzend die vertrauten Namen

$$0 := \emptyset, \quad 1 := \emptyset^+, \quad 2 := \emptyset^{++}, \quad \dots$$

Die Menge der natürlichen Zahlen $\mathbb{N}$ ist dann $\{0, 1, 2, \dots\}$.

Sei A eine Menge und $k \in \mathbb{N}$. In Abschnitt 3.1 hatten wir bereits A^k, die Menge aller k-Tupel über A definiert. Im wesentlichen die gleiche Definition ist, ein k-Tupel über A aufzufassen als eine Funktion von $\{1, 2, \dots, k\}$ nach A. Das Element $(a_1, (a_2, \dots, (a_{k-1}, a_k) \dots))$ wird dann aufgefasst als die Funktion, die i abbildet auf a_i.

4.1 Die Wohlordnung der natürlichen Zahlen

Für $n, m \in \mathbb{N}$ gilt $n < m$ (im bekannten Sinn) genau dann, wenn $n \in m$. Wir schreiben $n \leq m$ falls $n < m$ oder $n = m$ gilt. Die Relation $\leq$ ist eine *lineare Ordnung* (synonym, eine *totale Ordnung*):

- Für alle $a \in \mathbb{N}$ gilt $a \leq a$ (*Reflexivität*);
- Für alle $a, b \in \mathbb{N}$ mit $a \leq b$ und $b \leq a$ gilt $a = b$ (*Antisymmetrie*);
- Falls $a, b, c \in \mathbb{N}$ so dass $a \leq b$ und $b \leq c$, dann gilt $a \leq c$ (*Transitivität*);
- Für alle $a, b \in \mathbb{N}$ gilt $a \leq b$ oder $b \leq a$ (*Totalität*).

Ausserdem ist $\leq$ eine *Wohlordnung*: für jede Teilmenge T von $\mathbb{N}$ existiert ein *kleinstes Element*, das heißt, für jedes $T \subseteq \mathbb{N}$ gibt es ein Element $x \in T$, so dass es kein $y \in T$ gibt mit $y < x$. Wir bemerken, dass $<$ und $\leq$ binäre Relationen auf $\mathbb{N}$ sind: wir begreifen

$\leq$ als die Teilmenge von $\mathbb{N} \times \mathbb{N}$, die alle geordneten Paare (n, m) mit $n \leq m$ enthält. Ordnungsrelationen bilden ein eigenes Kapitel der Vorlesung (Abschnitt 9).

Im Gegensatz dazu ist beispielsweise die übliche und aus der Schule bekannte Ordnung der ganzen Zahlen $\mathbb{Z}$ *keine* Wohlordnung, denn bereits $\mathbb{Z}$ selbst besitzt kein kleinstes Element. Ebensowenig wohlgeordnet ist die übliche und aus der Schule bekannte Ordnung der nicht-negativen rationalen Zahlen $\mathbb{Q}$; diese besitzt zwar ein kleinstes Element, die 0, aber bereits die Teilmenge der positiven rationalen Zahlen hat kein kleinstes Element. Das *Wohlordnungsprinzip* besagt, dass jede Menge wohlgeordnet werden kann, d.h., für jede Menge A gibt es eine Wohlordnung auf A. Dieses Prinzip ist äquivalent zum Auswahlaxiom (siehe Abschnitt 2.4).

4.2 Vollständige Induktion

Betrachte eine Aussage A_n, die für natürliche Zahlen n gelten kann, aber von n abhängt. Die vollständige Induktion ist eine Methode, um zu zeigen, dass A_n für alle $n \in \mathbb{N}$ gilt. Dazu genügt es nämlich zweierlei zu zeigen:

1. *Induktionsanfang* (*Induktionsbasis*): A_0 gilt, und
2. *Induktionsschritt*: für alle n gilt: wenn A_n gilt (*Induktionsvoraussetzung*), dann auch A_{n+1} (*Induktionsbehauptung*).

Der *Induktionsschluss* ist dann, dass A_n für alle $n \in \mathbb{N}$ gilt.

Beispiel. Die Aussage A_n ist

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

Beweis. <u>Induktionsanfang</u> $n = 1$:

$$\sum_{i=1}^{1} i = 1 = \frac{1(1+1)}{2}$$

<u>Induktionsschritt</u>: Es gelte A_n, zu zeigen ist A_{n+1}.

$$\begin{aligned}
\sum_{i=1}^{n+1} i &= \sum_{i=1}^{n} i + (n+1) \\
&= \frac{n(n+1)}{2} + \frac{2}{2}(n+1) && \text{(Induktionsvoraussetzung)} \\
&= \frac{n^2 + n + 2n + 2}{2} \\
&= \frac{(n+1)(n+2)}{2}
\end{aligned}$$

□

Eine häufige Variante ist die, in der man im Induktionsschritt die Induktionsvoraussetzung nicht nur für n, sondern auch für $n' \in \mathbb{N}$ mit $n' < n$ verwendet.

Übungen.

18. Zeigen Sie Proposition 6 mit Hilfe von vollständiger Induktion.

4.3 Addition, Multiplikation, Exponentiation

Die *Addition* ist eine zweistellige Operation auf den natürlichen Zahlen, das heißt, eine Funktion $+ : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, induktiv definiert wie folgt:

$$\begin{aligned} n + 0 &:= n \\ n + m^+ &:= (n + m)^+ \end{aligned}$$

Die *Subtraktion* von zwei Zahlen $n, m \in \mathbb{N}$ ist für $m \leq n$ definiert als diejenige Zahl $d \in \mathbb{N}$, die $d + m = n$ erfüllt. Damit ist die Subtraktion blos eine *partielle* Operation auf $\mathbb{N}$, da sie für $n, m \in \mathbb{N}$ mit $n < m$ nicht definiert ist.

Allerdings ist die Subtraktion eine (totale, d.h., überall definierte) zweistellige Operation auf den *ganzen Zahlen* $\mathbb{Z}$. Die ganzen Zahlen enthalten die natürlichen Zahlen, und enthalten zudem für jedes $n \in \mathbb{N} \setminus \{0\}$ ein weiteres Element, welches mit $-n$ bezeichnet wird. Die Addition wird wie folgt von $\mathbb{N}$ auf $\mathbb{Z}$ erweitert: für $n, m \in \mathbb{N}$ definieren wir

$$\begin{aligned} (-n) + (-m) &:= -(n + m) \\ (-n) + m &:= \begin{cases} n - m & \text{falls } m \leq n \\ -(n - m) & \text{sonst.} \end{cases} \end{aligned}$$

Für $n \in \mathbb{N}$ definieren wir $-(-n) := n$. Dann gilt für alle $z \in \mathbb{Z}$ dass $-z = 0 - z$.

Auch die *Multiplikation* $\cdot : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ definiert man leicht induktiv mit Hilfe der Addition:

$$\begin{aligned} n \cdot 0 &:= 0 \\ n \cdot m^+ &:= n \cdot m + n \end{aligned}$$

Und schließlich die *Exponentiation* $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$ mit Hilfe der Multiplikation:

$$\begin{aligned} n^0 &:= 1 \\ n^{m^+} &:= n^m \cdot n \end{aligned}$$

Addition und Multiplikation sind kommutativ, assoziativ, und Multiplikation ist distributiv über Addition (siehe Abschnitt 1.3). Auch Multiplikation und Exponentiation lassen sich in der gewohnten Weise auf ganz $\mathbb{Z}$ definieren. In Summen können wir wegen der Assoziativität auf das Setzen von Klammern verzichten, ebenso in Produkten. Wir verwenden die Schreibweise

- $\sum_{i=1}^{n} a_i$ für $a_1 + a_2 + \cdots + a_n$. Per Definition gilt $\sum_{i=1}^{0} a_i := 0$.
- $\prod_{i=1}^{n} a_i$ für $a_1 \cdot (\cdots) \cdot a_n$. Per Definition gilt $\prod_{i=1}^{0} a_i := 1$.

In Ausdrücken mit $+$ und $\cdot$, wie zum Beispiel $(x + 1) \cdot (3 + 2 \cdot y + x \cdot z) + 3$, gilt die Konvention, dass Multiplikation stärker bindet als Addition. Das Multiplikationszeichen wird oft weggelassen.

Solche Ausdrücke können durch *Ausmultiplizieren* und *Zusammenfassen* in eine Normalform gebracht werden, in der auf Klammern ganz verzichtet werden kann: in unserem Beispiel wäre das $3x + 2xy + x^2y + 6 + 2y + xz$. Dieses Ergebnis ist eindeutig bis auf Vertauschung von Summanden und Vertauschung von Faktoren.

Übungen.

19. Beweisen Sie die Kommutativität der in diesem Abschnitt eingeführten Addition.

20. Geben Sie eine formale Definition der Multiplikation auf ganz $\mathbb{Z}$ (die sich natürlich so verhalten soll, wie wir es aus der Schule gewohnt sind).

4.4 Teilbarkeit und Primzahlen

Wir definieren auf $\mathbb{N}$ die *Teilbarkeitsrelation*: für $a, b \in \mathbb{N}$ gelte $a|b$ (sprich: *a teilt b*) genau dann, wenn es ein $k \in \mathbb{N}$ gibt mit $a \cdot k = b$. In diesem Fall heißt a *Teiler* von b. Sicherlich ist $0 \in \mathbb{N}$ durch jede Zahl $n \in \mathbb{N}$ teilbar. Für alle Zahlen $n \in \mathbb{N}$ gilt $1|n$ und $n|n$.

Definition 22. *Eine Zahl $p \in \mathbb{N}$ heißt* Primzahl *(oder* prim*), wenn sie größer als* 1 *ist und nur durch* 1 *und sich selbst teilbar ist.*

Ein *Primteiler* von n ist ein Teiler von n, der prim ist.

Lemma 23. *Jede Zahl $n \in \mathbb{N}$ mit $n > 1$ hat einen Primteiler.*

Beweis. Falls die Aussage nicht gilt, dann gibt es eine kleinste Zahl $n \in \mathbb{N}$ mit $n > 1$ ohne Primteiler (da $\mathbb{N}$ eine Wohlordnung ist!). Insbesondere n selbst ist nicht prim, hat also einen Teiler t, der von 1 und n verschieden ist. Da $t < n$ muss wegen der Minimalität von n die Zahl t einen Primteiler p besitzen. Dann ist p aber auch ein Primteiler von n, ein Widerspruch. $\square$

Seien $a, b_1, b_2 \in \mathbb{N}$ so dass $a|b_1$, $a|b_2$, und $b_1 < b_2$. Dann gilt $a|(b_2 - b_1)$: denn wenn $b_1 = k_1 a$ und $b_2 = k_2 a$, dann ist $b_2 - b_1 = (k_2 - k_1)a$, also durch a teilbar. Diese Beobachtung wird verwendet in folgendem Beweis, der sich bereits im Werk *Elemente* von Euklid[18] findet. Vielleicht der älteste aller Beweise!

Satz 24. *Es gibt unendlich viele Primzahlen.*

[18] Euklid von Alexandria, lebte vermutlich im 3. Jahrhundert v. Chr.

Beweis von Euklid. Seien $p_1, \ldots, p_\ell$ Primzahlen. Sei $n := p_1 \cdot p_2 \cdot (\cdots) \cdot p_\ell + 1$, und t ein Primteiler von n (Lemma 23). Dann ist t von allen p_i verschieden, da ansonsten t sowohl n als auch $p_1 \cdot p_2 \cdot (\cdots) \cdot p_\ell$ teilen würde, und somit auch die 1, was nicht sein kann. Da dieses Argument für beliebiges ℓ funktioniert, muss es unendlich viele Primzahlen geben. $\square$

Beispiel 13. *Teilbarkeitsregeln:*

- *$n \in \mathbb{N}$ ist genau dann durch 2 teilbar, wenn die letzte Ziffer von n im Dezimalsystem* 0, 2, 4, 6, *oder* 8 *lautet.*
- *$n \in \mathbb{N}$ ist genau dann durch 4 teilbar, wenn die letzten beiden Ziffern von n im Dezimalsystem, als Zahl $b \in \{0, \ldots, 99\}$ aufgefasst, durch* 4 *teilbar ist. Das liegt daran, dass $4|100$ und entsprechend $n = 100 \cdot k + b$ genau dann durch* 4 *teilbar ist, wenn b durch* 4 *teilbar ist.*

Übungen.

21. Ist das Produkt der ersten ℓ Primzahlen plus eins eine Primzahl?
22. Zeigen Sie: Eine Zahl $n \in \mathbb{N}$ ist genau dann durch 8 teilbar, wenn Ihre letzten 3 Ziffern, als Zahl $b \in \{0, \ldots, 999\}$ aufgefasst, durch 8 teilbar ist.

Der folgende Satz erlaubt eine Abschätzung der Dichte der Primzahlen mit Hilfe der Logarithmusfunktion, von Gauß[19] und Legendre[20] vermutet und bewiesen von Hadamard[21] und unabhängig von Poussin[22]. Für $x \in \mathbb{R}$, $x > 0$, schreiben wir $\pi(x)$ für die Anzahl der Primzahlen, die kleiner gleich x sind.

Satz 25 (Der Primzahlsatz)**.** $\pi(x) \sim \frac{x}{\ln(x)}$.

(Die Definition des Symbols $\sim$ findet sich gegen Ende von Abschnitt 2.6.)

Satz 26 (Fundamantalsatz der Arithmetik)**.** *Jede natürliche Zahl $n > 0$ kann auf genau eine Weise als Produkt*

$$n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot (\cdots) \cdot p_k^{\alpha_k}$$

geschrieben werden, wobei $k \in \mathbb{N}$, $p_1, p_2, \ldots, p_k$ Primzahlen mit $p_1 < p_2 < \cdots < p_k$ und $\alpha_1, \alpha_2, \ldots, \alpha_k \in \mathbb{N} \setminus \{0\}$.

[19] Johann Carl Friedrich Gauß; geboren am 30. April 1777 in Braunschweig; gestorben am 23. Februar 1855 in Göttingen.

[20] Adrien-Marie Legendre; geboren am 18. September 1752 in Paris; gestorben am 10. Januar 1833 in Paris.

[21] Jacques Salomon Hadamard; geboren am 8. Dezember 1865 in Versailles; gestorben am 17. Oktober 1963 in Paris.

[22] Charles-Jean Gustave Nicolas Baron de La Vallée Poussin; geboren am 14. August 1866 in Löwen; gestorben am 2. März 1962 in Brüssel.

Beweis. **Existenz.** Für $n = 1$, wähle das leere Produkt mit $k = 0$ (siehe die Definition von $\prod_{i=1}^{0} a_i$ in Abschnitt 4.3). Wir führen einen Widerspruchsbeweis, und nehmen an, es gibt Zahlen, die sich nicht als Produkt von Primzahlen darstellen lassen. Da $\mathbb{N}$ wohlgeordnet ist, gibt es eine kleinste solche Zahl $n > 1$. Da n keine Primzahl sein kann, besitzt n einen Teiler, und $n = ab$ für $a, b \in \mathbb{N}$ größer 1. Nach der Wahl von n lassen sich sowohl a als auch b als Produkt von Primzahlen schreiben. Dann ist aber das Produkt dieser beiden Produkte eine Primfaktorzerlegung von n, Widerspruch.

Eindeutigkeit. Wird im nächsten Abschnitt gezeigt. □

Es ist kein effizientes Verfahren bekannt, um für eine gegebene natürliche Zahl die Primfaktorzerlegung zu berechnen. Ob es einen Algorithmus gibt, der in *polynomieller Zeit* die Antwort liefert, ist eines der wichtigsten offenen Probleme der theoretischen Informatik. Auf der anderen Seite haben Agrawal[23], Kayal[24] und Saxena[25] im Jahre 2002 einen Algorithmus mit polynomieller Laufzeit entdeckt, um für eine gegebene Zahl zu entscheiden, ob sie eine Primzahl ist.

4.5 Der euklidische Algorithmus

Der *größte gemeinsame Teiler* von $a, b \in \mathbb{N}$ ist die größte natürliche Zahl d, die a und b teilt; im Falle von $a = b = 0$ definieren[26] wir $\mathrm{ggT}(0, 0) := 0$. Wir schreiben $\mathrm{ggT}(a, b)$ für diese Zahl d. Wir werden in diesem Abschnitt ein sehr effizientes Verfahren kennenlernen, um den größten gemeinsamen Teiler von zwei gegebenen Zahlen auszurechnen (ohne einen einzigen Teiler von a und b zu bestimmen!).

Für $z \in \mathbb{Z}$ definieren wir $|z| := z$ falls $z \in \mathbb{N}$, und $|z| := -z$ falls $z \in \mathbb{Z} \setminus \mathbb{N}$.

Lemma 27 (Division mit Rest)**.** *Seien $a, b \in \mathbb{Z}$ und $b \neq 0$. Dann gibt es eindeutige $q, r \in \mathbb{Z}$ mit $a = qb + r$ und $0 \leq r < |b|$.*

Beweis. **Existenz.** Wir betrachten zunächst den Fall $b > 0$. Es sei $q \in \mathbb{Z}$ größtmöglich so dass $bq \leq a$. Setze $r := a - bq \in \mathbb{N}$. Nach der Wahl von q gilt $b(q+1) > a$, also $r = a - bq < b$.

Falls $b < 0$ so sei $q \in \mathbb{Z}$ größtmöglich so dass $-bq \leq a$. Setze nun $r := a + bq \in \mathbb{N}$. Nach der Wahl von q gilt $-b(q + 1) = -bq - b > a$, also $r = a + bq < -b$.

Eindeutigkeit. Sei q', r' ein weiteres Paar mit $a = bq' + r'$ mit $0 \leq r' \leq |b| - 1$. Subtraktion liefert $(q - q')b = r - r'$. Gilt $q = q'$, dann folgt $r = r'$. Gilt $q \neq q'$, so nehmen wir ohne Beschränkung der Allgemeinheit an, dass $q > q'$. Dann folgt $b \geq r - r'$, im Widerspruch zu $r < b$. □

[23] Manindra Agrawal, geboren am 20. Mai 1966 in Allahabad.

[24] Neeraj Kayal, geboren in Guwahati.

[25] Nitin Saxena, geboren am 3. Mai 1981 in Allahabad.

[26] Da jede natürliche Zahl ein Teiler der 0 ist, kann man 0 als *größtes* Element von $\mathbb{N}$ bezüglich Teilbarkeit auffassen; und damit ist die 0 auch der *größte* gemeinsame Teiler von 0 und 0.

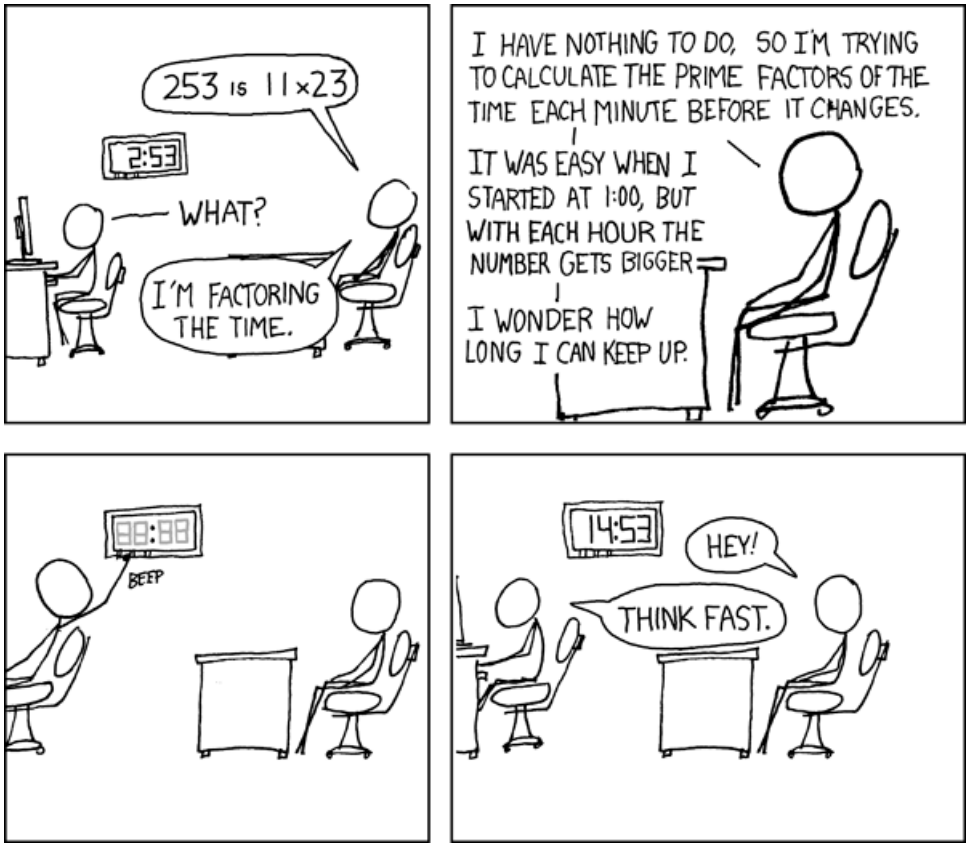


Abbildung 4: Heruntergeladen von http://xkcd.com/247.

Für die Zahl r aus Lemma 27 schreiben wir auch $a \bmod b$; es handelt sich hier um den *Rest* beim bekannten Verfahren der schriftlichen Division.

Für $q \in \mathbb{Q}$ schreiben wir $\lfloor q \rfloor$ für die (eindeutige) größte Zahl $z \in \mathbb{Z}$ die kleiner ist als q. Dann gilt für $a, b \in \mathbb{N}$ und $b \neq 0$, dass $a = \lfloor a/b \rfloor \cdot b + (a \bmod b)$. Das folgende Beispiel zeigt, dass die Annahme, dass a und b nicht negativ sind, für diese Aussage wichtig ist.

Beispiel 14. *Es gilt*

$$13 = 4 \cdot 3 + (13 \bmod 3) = 12 + 1,$$
$$\textit{aber } -13 = 5 \cdot -3 + (-13 \bmod -3) = -15 + 2$$

obwohl $\lfloor -13/-3 \rfloor = \lfloor 13/3 \rfloor = 4$.

Lemma 28. *Es seien* $a, b \in \mathbb{N}$ *mit* $b > 0$. *Dann gilt* $\mathrm{ggT}(a, b) = \mathrm{ggT}(b, a \bmod b)$.

```
// Der euklidische Algorithmus EUKLID(m, n)
// Eingabe: m, n ∈ ℕ mit m ≤ n.
// Ausgabe: ggT(m, n).
Falls m = 0 gebe n aus.
Falls m|n gebe m aus.
Gebe EUKLID(n mod m, m) aus.
```

Abbildung 5: Der euklidische Algorithmus.

Beweis. Seien $q, r \in \mathbb{N}$ so dass $a = bq + r$ mit $(a \bmod b) = r < b$. Sei $d := \mathrm{ggT}(a, b)$ und $d' := \mathrm{ggT}(b, r)$. Da d sowohl a als auch b teilt, so teilt d auch $r = a - bq$. Also $d \leq d' = \mathrm{ggT}(r, b)$. Umgekehrt ist d' ein Teiler von b und von r, also auch von $a = bq + r$. Also $d' \leq d = \mathrm{ggT}(a, b)$, und daher $d = d'$. □

Lemma 28 ist die zentrale Beobachtung im Korrektheitsbeweis für den Algorithmus aus Abbildung 5 zur Berechnung des größten gemeinsamen Teilers zweier gegebener Zahlen $m, n \in \mathbb{N}$, dem *euklidischen Algorithmus*.

Korollar 29 (Lemma von Bézout[27])**.** *Es seien $m, n \in \mathbb{N}$. Dann gibt es ganze Zahlen $a, b \in \mathbb{Z}$ mit $\mathrm{ggT}(m, n) = am + bn$.*

Beweis. Wir führen einen Beweis mit Hilfe von vollständiger Induktion (siehe Abschnitt 4.2). Für $k \in \mathbb{N}$ besagt die Induktionsaussage A_k, dass für alle $m, n \in \{0, \ldots, k\}$ Zahlen $a, b \in \mathbb{Z}$ existieren mit $\mathrm{ggT}(m, n) = am + bn$.

Induktionsanfang: Für $k = 0$ ist $m = n = 0$ und es gilt $\mathrm{ggT}(m, n) = 0 = 0m + 0n$; also gibt es $a, b \in \mathbb{Z}$ wie gefordert.

Induktionsschritt: Angenommen, A_k gilt. Wir wollen A_{k+1} zeigen. Seien $m, n \in \{0, \ldots, k+1\}$. Falls $m = n$, dann ist $\mathrm{ggT}(m, n) = m$ und wir wählen $a = 1$ und $b = 0$. Falls $m \neq n$ können wir ohne Beschränkung der Allgemeinheit annehmen, dass $m < n$; ansonsten vertauschen wir im folgenden die Rollen von m und n und von a und b. Wähle $q, r \in \mathbb{N}$ so dass $n = mq + r$ für $r < m \leq k + 1$ (Division mit Rest; Lemma 27). Da $r < m$ und $m < n$ und $n \leq k + 1$ ist, so gibt es nach Induktionsannahme $a', b' \in \mathbb{Z}$ mit $\mathrm{ggT}(r, m) = a'r + b'm$. Dann impliziert Lemma 28, dass

$$\mathrm{ggT}(m, n) = \mathrm{ggT}(r, m) = a'r + b'm = a'(n - mq) + b'm = (b' - a'q)m + a'n\,,$$

und die Aussage A_{k+1} folgt mit $a := b' - a'q$ und $b := a'$.

Induktionsschluss: Die Aussage A_k gilt für alle $k \in \mathbb{N}$, und das ist gerade die zu zeigende Aussage. □

[27] Etienne Bézout, geboren am 31. März 1730 in Nemours, Seine-et-Marne; gestorben am 27. September 1783 in Basses-Loges nahe Fontainebleau.

// Der erweiterte euklidische Algorithmus E-EUKLID(m, n)
// Eingabe: $m, n \in \mathbb{N}$ mit $m \leq n$.
// Ausgabe: $a, b \in \mathbb{Z}$ so dass $\mathrm{ggT}(m, n) = am + bn$.
Falls $m|n$ gebe $(1, 0)$ aus.
Sei (a', b') die Ausgabe von E-EUKLID$(n \bmod m, m)$.
Gebe $(b' - a'\lfloor n/m \rfloor, a')$ aus.

Abbildung 6: Der erweiterte euklidische Algorithmus.

Durch eine kleine Erweiterung kann der euklidische Algorithmus auch dazu verwendet werden, um für gegebene $m, n \in \mathbb{N}$ die Zahlen $a, b \in \mathbb{Z}$ aus dem Lemma von Bézout zu berechnen; siehe Abbildung 6.

Beispiel 15. *Betrachten* $m = 7$ *und* $n = 16$. *Es gilt* $\mathrm{ggT}(7, 16) = 1$, *wir suchen also* $a, b \in \mathbb{Z}$ *so dass* $1 = a \cdot 7 + b \cdot 16$. *Für E-Euklid*$(7, 16)$ *wird zunächst E-Euklid*$(2, 7)$ *berechnet, und dafür wiederum E-Euklid*$(1, 2)$, *was* $(1, 0)$ *zurückliefert. Also gilt*

$$\begin{aligned} \textit{E-Euklid}(2,7) &= (0 - 1 \cdot \lfloor 7/2 \rfloor, 1) = (-3, 1) \\ \textit{und } \textit{E-Euklid}(7,16) &= (1 + 3\lfloor 16/7 \rfloor, -3) = (7, -3). \end{aligned}$$

Und tatsächlich gilt $7 \cdot 7 - 3 \cdot 16 = 49 - 48 = 1$.

Korollar 30 (Lemma von Euklid)**.** *Teilt eine Primzahl das Produkt zweier natürlicher Zahlen, so auch mindestens einen der Faktoren.*

Beweis. Seien $n, m \in \mathbb{N}$ beliebig. Angenommen, eine Primzahl p teilt nm, aber nicht den Faktor n. Dann ist zu zeigen, dass p ein Teiler von m ist. Da $\mathrm{ggT}(p, n) = 1$, existieren nach dem Lemma von Bézout zwei ganze Zahlen a und b so dass $ap + bn = 1$ gilt. Durch Multiplizieren mit m erhalten wir $p(am) + (nm)b = m$. Dann teilt p beide Summanden, und daher $p|m$. □

Eindeutigkeit der Primfaktorzerlegung. Angenommen, es gibt positive natürliche Zahlen mit verschiedenen Zerlegungen, dann gibt es eine kleinste solche Zahl $n > 0$. Es folgt aus dem Lemma von Euklid, dass jeder Primfaktor p der einen Zerlegung einen Primfaktor q der anderen Zerlegung teilt. Da q prim ist, gilt $p = q$. Dann hat aber bereits n/p zwei verschiedene Zerlegungen, ein Widerspruch zur Wahl von n. □

5 Modulare Arithmetik

Auf der Menge $\mathbb{Z}_n := \{0, 1, \ldots, n-1\}$ der natürlichen Zahlen kleiner als n definieren wir die folgenden zweistelligen Operationen: für $a, b \in \mathbb{Z}_n$, setze

- $a +_{\text{mod } n} b := (a + b) \mod n$ (*Addition modulo* n).
- $a -_{\text{mod } n} b := (a - b) \mod n$ (*Subtraktion modulo* n).
- $a \cdot_{\text{mod } n} b := (a \cdot b) \mod n$ (*Multiplikation modulo* n).

Allerdings ist es zu umständlich, die Operationszeichen mit dem Subskript „mod n" zu benutzen. Deshalb lässt man diese Information im Subskript gerne weg, benutzt einfach die Zeichen $+$, $-$, und $\cdot$, und sagt oder schreibt dazu, dass man modulo n rechnet. Die Elemente von $\mathbb{Z}_n$ nennt man auch die *Restklassen* modulo n; woher dieser Name kommt, wird später erklärt werden (im Abschnitt 5.1).

Rechnen modulo 2

Beim Rechnen modulo n = 2 stimmen Addition und Subtraktion überein. Addition und Multiplikation werden durch die folgenden Verknüpfungstafeln beschrieben.

$+$	0	1
0	0	1
1	1	0

$\cdot$	0	1
0	0	0
1	0	1

Abbildung 7: Die Verknüpfungstafeln für $+$ und $\cdot$ modulo 2.

Diese Tabelle ist vertrauter, wenn man das Symbol 0 als „gerade" und 1 als „ungerade" liest. Man hat dann „gerade plus gerade gleich gerade", „gerade mal ungerade gleich gerade", usw.

Rechnen modulo 5

Die Verknüpfungstafeln für $+$, $-$, und $\cdot$ finden sich in Abbildung 8. Die Subtraktion kann man einfacher mit Hilfe der einstelligen bijektiven Operation $x \mapsto -x$ mit $-x := 0 - x$ definieren (siehe die Tabelle in Abbildung 9), denn $x - y = x + (-y)$.

5.1 Die Homomorphieregel

Wenn man umfangreiche Rechnungen modulo n auszuführen hat, dann ist die *Homomorphieregel* außerordentlich hilfreich. Sie besagt, dass man auch Zwischenergebnisse modulo

$+$	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

$-$	0	1	2	3	4
0	0	4	3	2	1
1	1	0	4	3	2
2	2	1	0	4	3
3	3	2	1	0	4
4	4	3	2	1	0

$\cdot$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Abbildung 8: Die Verknüpfungstafeln für $+$ und $\cdot$ modulo 5.

x	0	1	2	3	4
$-x$	0	4	3	2	1

Abbildung 9: Die Abbildungstafel für $x \mapsto -x$.

n rechnen darf, ohne dass sich das Endergebnis ändert. Formal besagt sie, dass für ganze Zahlen a, b stets gilt:

$$\begin{aligned}(a+b) \bmod n &= (a \bmod n + b \bmod n)\\ (a-b) \bmod n &= (a \bmod n - b \bmod n)\\ (a \cdot b) \bmod n &= (a \bmod n \cdot b \bmod n)\end{aligned}$$

Hierbei sind $+$, $-$, und $\cdot$ auf der linken Seite die bekannten und im letzten Kapitel eingeführten Operationen der Addition, Subtraktion und Multiplikation auf den ganzen Zahlen, während die gleichen Symbole auf der rechten Seite für die in diesem Kapitel eingeführten Operationen auf $\mathbb{Z}_n$ stehen.

Der ständige Zusatz $\bmod n$ wird rasch lästig und gern weggelassen. Um Missverständnisse zu vermeiden, kann man ihn am Ende der Rechnung in Klammern angeben und die Gleichheitszeichen durch $\equiv$ ersetzen, wie im folgenden Beispiel.

$$108 \cdot 33 - 22 \equiv 3 \cdot 3 - 2 \equiv 9 + 3 \equiv 2 \pmod 5$$

Formal definiert man $a \equiv b \pmod n$ als Abkürzung von $a \bmod n = b \bmod n$, und liest dies einprägsam als *a ist kongruent zu b modulo n.*

Aufgrund der Homomorphieregel kann man sich die Restklasse $i \in \mathbb{Z}_n$ auch vorstellen als die Menge aller Zahlen aus $\mathbb{Z}$, die bei Division durch n den Rest i lassen; daher der Name *Restklasse*. Auch gebräuchlich ist der Name *Kongruenzklasse modulo n*, denn in dieser Sichtweise ist jedes Element von $\mathbb{Z}_n$ eine Menge von Zahlen aus $\mathbb{Z}$, die paarweise kongruent sind modulo n. Dieses Thema wird uns in abstrakter Form in Abschnitt 8 wiederbegegnen.

5.2 Uhrzeiten

Eine größere Menge Bauschotter wird mit einer Eisenbahn von A nach B transportiert, dafür sind 50 Fahrten erforderlich. Das Beladen des Zuges dauert vier Stunden, jede Fahrt zwei Stunden pro Richtung und das Abladen drei Stunden. Pausen werden nicht gemacht. Mit dem Beladen für die erste Fahrt wurde mittags um 12 Uhr begonnen. Zu welcher Uhrzeit wird der letzte Zug zurück erwartet?

Antwort: Für jede Fahrt wird vom Beginn des Beladens bis zur Rückkehr ein Zeitraum von 11 Stunden benötigt, insgesamt also $50 \cdot 11$ Stunden. Da nur nach der Uhrzeit der Rückkehr gefragt ist, kann modulo 24 gerechnet werden.

$$12 + 50 \cdot 11 \equiv 12 + 2 \cdot 11 \equiv 34 \equiv 10 \pmod{24}$$

Man erhält, dass der Zug um zehn Uhr morgens ankommt.

5.3 Die letzten Ziffern

Aufgabe: was sind die letzten beiden Ziffern von $333333 \cdot 444444 \cdot 56789$?
Kunstgriff: Die letzten beiden Ziffern einer natürlichen Zahl n sind offenbar die Ziffern von n modulo 100. Also

$$\begin{aligned}
333333 \cdot 444444 \cdot 56789 &\equiv 33 \cdot 44 \cdot 89 \\
&\equiv 33 \cdot 11 \cdot 4 \cdot 89 \\
&\equiv (330 + 33) \cdot (320 + 36) \\
&\equiv 63 \cdot 56 \\
&\equiv 3528 \\
&\equiv 28 \pmod{100}
\end{aligned}$$

5.4 Potenzieren modulo n

Bei Anwendungen der modularen Arithmetik z.B. in der Kryptographie hat man oft Ausdrücke auszuwerten wie zum Beispiel

$$13948451109438^{240598340963405982} \mod 23450983\,.$$

Gewöhnlicherweise sind die Zahlen noch viel größer als in diesem Beispiel, z.B. 1000-stellig. Es ist unmöglich, zuerst explizit die Potenz auszurechnen, und danach zu dividieren, um den gewünschten Rest zu berechnen. Wie kann man diesen Ausdruck also effizient ausrechnen? Eine Lösung dazu hat Al-Kachi[28] beschrieben, und ist die Kombination zweier Überlegungen.

[28] Ghiyath ad-Din Jamshid Mas'ud al-Kashi; geboren um 1380 in Kaschan, Iran; gestorben am 22. Juni 1429 in Samarkand, Timuridenreich, heute in Usbekistan.

- Man kann mittels der sogenannten *Methode des Quadrierens und Multiplizierens* (Englisch: *square and multiply*) die Berechnung der Potenz in kleine, handhabbare Schritte zerlegen; diese Methode wird oft *binäre Exponentiation* genannt, und war schon 200 v. Chr. in Indien bekannt.

- Bei jedem Rechenschritt kann man modular vereinfachen (Homomorphieregel); damit vermeidet man eine Explosion der Zwischenergebnisse.

Wir erklären anhand eines kleinen Beispiels: wir rechnen $2^{100000} \mod 100001$. Zunächst schreiben wir 100000 als Summe von Zweierpotenzen.

$$100000 = 2^{16} + 2^{15} + 2^{10} + 2^{9} + 2^{7} + 2^{5}$$

In anderen Worten: die Zahl 100000 in Binärschreibweise lautet

$$11000011010100000.$$

Also gilt

$$\begin{aligned} 2^{100000} &= 2^{2^{16}} \cdot 2^{2^{15}} \cdot 2^{2^{10}} \cdot 2^{2^{9}} \cdot 2^{2^{7}} \cdot 2^{2^{5}} \\ &= (((((\mathbf{2}^2 \cdot 2)^{2\cdot 2\cdot 2\cdot 2\cdot 2} \cdot 2)^2 \cdot 2)^{2\cdot 2} \cdot 2)^{2\cdot 2} \cdot 2)^{2\cdot 2\cdot 2\cdot 2\cdot 2}\,. \end{aligned} \tag{3}$$

Zum Ausdruck in (3) gelangt man wie folgt. Falls der Exponent größer als 1 ist, so startet die Binärschreibweise links mit einer 1. In unserem Ausdruck starten wir dann mit der Basis, in unserem Fall der 2 (in (3) ist diese fett markiert). Falls nun die nächste Ziffer in der Binärdarstellung von links gelesen eine 0 ist, so wird der Ausdruck quadriert. Falls die nächste Ziffer dagegen eine 1 ist, so wird der Ausdruck quadriert *und* mit 2 multipliziert.

Den Ausdruck (3) werten wir nun aus, wobei wir die Zwischenergebnisse modulo 100001 vereinfachen. Das kann man nun einen Computer machen lassen. Wir erhalten

$$2^{100000} \equiv 1024 \mod 100001.$$

5.5 Der chinesische Restsatz

Betrachte $m \cdot n$ Felder, die in einem Rechteck der Höhe m und der Breite n angeordnet sind. Nummeriere die Felder in der folgenden Art und Weise (siehe auch Abbildung 10): beginne mit dem Feld in der nullten Zeile und nullten Spalte. Nehmen wir nun an, in Schritt x befinden wir uns in der k-ten Zeile und ℓ-ten Spalte. Wir betrachten folgende Fälle.

- $k < m - 1$ und $\ell < n - 1$. Dann fahren wir mit dem Feld in der $(k+1)$-ten Zeile und $(\ell+1)$-ten Spalte fort.

- $k = m - 1$ und $\ell < n - 1$. Fahre mit dem Feld in der nullten Zeile und $(\ell+1)$-ten Spalte fort.

- $k < m - 1$ und $\ell = n - 1$. Fahre mit dem Feld in der $(k+1)$-ten Zeile und nullten Spalte fort.
- $k = m - 1$ und $\ell = n - 1$. Stoppe.

Unsere Frage lautet: für welche Werte von m und n werden mit diesem Verfahren *alle* Felder des Rechtecks erreicht?

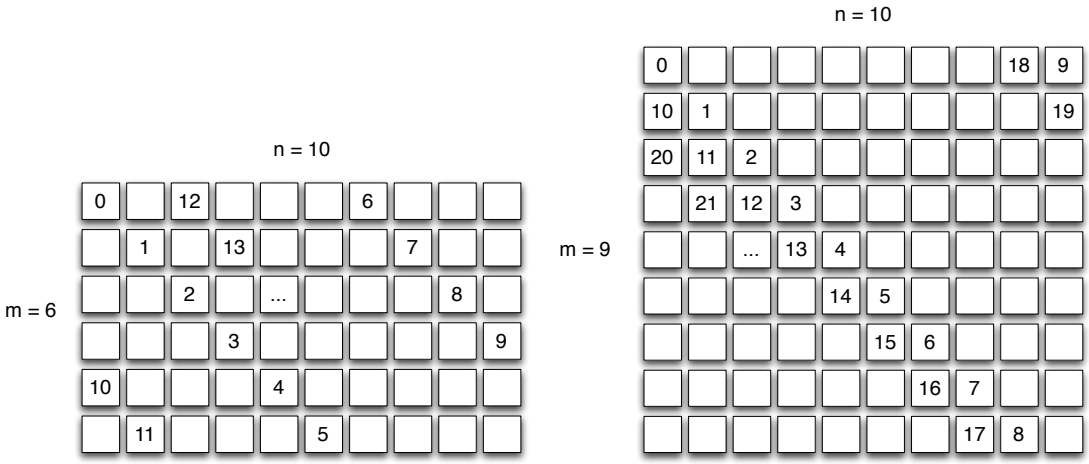


Abbildung 10: Ein Schnappschuss beim Durchlaufen von $\{0, \ldots, 5\} \times \{0, \ldots, 9\}$ und von $\{0, \ldots, 8\} \times \{0, \ldots, 9\}$.

Zur Lösung dieser Frage ist die erste wichtige Beobachtung, dass wir uns beim x-ten Schritt gerade in Zeile $x \bmod m$ und in Spalte $x \bmod n$ befinden. Wir behaupten, dass genau dann alle Felder durchlaufen werden, wenn m und n *teilerfremd* sind, d.h., wenn $\mathrm{ggT}(m, n) = 1$.

Betrachte zunächst den Fall $\mathrm{ggT}(m, n) > 1$, in anderen Worten, m und n haben einen gemeinsamen Teiler $d > 1$. Wenn wir das Feld (k, l) in Schritt x erreichen, dann muss gelten dass $x \equiv k \pmod{d}$ und $x \equiv l \pmod{d}$. Das geht aber nur dann, wenn $k \equiv l \pmod{d}$ und ist offensichtlich nicht für alle Paare (k, l) der Fall (siehe Abbildung 10, linke Seite).

Betrachten wir umgekehrt den Fall $\mathrm{ggT}(m, n) = 1$. In diesem Fall behaupten wir also, dass es für jedes Feld (k, l) mit $k \leq m$, $l \leq n$, ein i gibt so dass

$$\begin{aligned} x &\equiv k \pmod{m} \\ x &\equiv l \pmod{n} \,. \end{aligned}$$

Siehe Abbildung 10, rechte Seite. Da $\mathrm{ggT}(m, n) = 1$, gibt es nach dem Lemma von Bézout $a, b \in \mathbb{Z}$ so dass $a \cdot m + b \cdot n = 1$.

Behauptung: Die natürliche Zahl $x := l \cdot a \cdot m + k \cdot b \cdot n$ leistet das Gewünschte. Denn

$$lam + kbn \equiv kbn \equiv (1 - am)k \equiv k \mod m$$

und

$$lam + kbn \equiv lam \equiv (1 - bn)l \equiv l \mod n\ .$$

Diese Idee lässt sich unschwer auf endlich viele Kongruenzen mit teilerfremden Moduli $n_1, \ldots, n_r$ verallgemeinern. Diese Verallgemeinerung ist auch unter dem Namen *chinesischer Restsatz* bekannt, und findet sich bereits in *Sun Zi's Handbuch der Arithmetik*[29]; dieser Satz wird im 3. Semester nochmals ausführlicher behandelt werden.

Satz 31. *Es seien $n_1, \ldots, n_r \in \mathbb{N}$ teilerfremd, und $a_1, \ldots, a_r \in \mathbb{Z}$. Dann gibt es genau eine natürliche Zahl $x \in \{0, \ldots, n_1 \cdot (\cdots) \cdot n_r - 1\}$ mit $x \equiv a_i \pmod{n_i}$ für alle $i \in \{1, \ldots, r\}$.*

Im Beweis zeigt man, dass die Abbildung

$$\mathbb{Z}_{n_1 \cdots n_r} \to \mathbb{Z}_{n_1} \times \cdots \times \mathbb{Z}_{n_r}, \quad x \mapsto (x \bmod n_1, \ldots, x \bmod n_r)$$

surjektiv (und damit bijektiv! Proposition 6) ist, wenn $\mathrm{ggT}(n_i, n_j) = 1$. Umgekehrt ist diese Abbildung nicht injektiv (also insbesondere nicht bijektiv), wenn $\mathrm{ggT}(n_i, n_j) \neq 1$.

5.6 Zufall in der Informatik

Eine wichtige Klasse von Berechnungsproblemen in der theoretischen Informatik ist die Klasse aller Probleme, die ein Computer in polynomieller Zeit lösen kann. Polynomiell bedeutet hier: polynomiell in der Größe der Eingabe. Wenn n die Eingabegröße bezeichnet, dann ist also ein Algorithmus, der stets mit $n^5 + 1000$ Rechenschritten auskommt, polynomiell, aber ein Algorithmus, der manchmal 2^n Rechenschritte benötigt, nicht. Die Klasse von Problemen mit einem polynomiellen Algorithmus wird mit P bezeichnet. Eine formale Definition dieser Klasse werden Sie in den weiterführenden Informatikvorlesungen kennenlernen.

In der Praxis ist man aber auch oft mit einem Algorithmus zufrieden, der *Zufallsbits* verwenden darf, und dessen Laufzeit im *Erwartungsfall* polynomiell ist. Solche Algorithmen werden auch *Las Vegas Algorithmen* genannt. Die Klasse aller Probleme, die von einem solchen Algorithmus gelöst werden können, nennt man ZPP (*Zero-Error Probabilistic Polynomial Time*). Interessanterweise kennt man kein Problem in ZPP, von dem man nicht auch wüsste, dass es in P liegt. Lange Zeit hatte das bereits in Abschnitt 4.4 erwähnte Primalitätsproblem diesen Status: man kennt einen *randomisierten* Algorithmus mit erwartet polynomieller Laufzeit, aber man wusste nicht, ob das Problem in P ist. Seit 2002 weiss man aber, dass es auch einen polynomiellen deterministischen (d.h., nicht randomisierten) Algorithmus für den Test auf Primalität gibt [1].

[29] Sun Zi, lebte irgendwann zwischen dem 3ten und 5ten Jahrhundert in China.

Es gibt noch eine weitere interessante Art von randomisierten Algorithmen. Anstatt zu fordern, dass die Laufzeit des Algorithmus im Erwartungsfall polynomial ist, fordert man, dass der Algorithmus immer polynomial ist, aber nur mit großer Wahrscheinlichkeit das richtige Ergebnis liefern muss. Randomisierte Algorithmen mit garantierter Laufzeit, aber nur wahrscheinlich korrekter Ausgabe werden auch *Monte Carlo Algorithmen* genannt. Konkret interessiert man sich zum Beispiel für die Klasse der Entscheidungsprobleme, für die es einen randomisierten Algorithmus mit garantiert polynomieller Laufzeit gibt, so dass

- wenn die korrekte Antwort 'Nein' lautet, das Programm garantiert 'Nein' sagt;
- wenn die korrekte Antwort 'Ja' lautet, das Program 'Ja' sagt mit Wahrscheinlichkeit von mindestens 2/3.

Die Klasse aller Probleme, die von einem solchen Algorithmus gelöst werden können, nennt man RP (*Randomized Polynomial Time*). Der genau Wert 2/3 ist in dieser Definition von keiner besonderen Bedeutung: wir hätten jede andere Zahl p, die strikt zwischen 0 und 1 liegt, verwenden können. Das liegt daran, dass durch wiederholtes Anwenden des Algorithmus die Fehlerwahrscheinlichkeit sehr schnell verbessert werden kann. Wenn man anstatt von p die Erfolgswahrscheinlichkeit $0 < q < 1$ erreichen will, führt man den Algorithmus n mal aus (das kann man sehr gut *parallel* tun, das heißt, gleichzeitig auf verschiedenen Rechnern). Der neue Algorithmus gibt nur dann 'Ja' aus, wenn *eines* der n Ergebnisse 'Ja' lautetet, und sonst 'Nein'. Falls die korrekte Antwort 'Nein' lautet, dann lauten alle Ergebnisse 'Nein', und unser neuer Algorithmus gibt 'Nein' aus. Falls die korrekte Antwort 'Ja' lautet, dann gibt der neue Algorithmus mit Wahrscheinlichkeit $1 - (1 - p)^n$ die Antwort 'Ja'. Für großes n geht der Ausdruck $1 - (1 - p)^n$ schnell gegen 1; inbesondere können wir ein $n \in \mathbb{N}$ wählen, so dass die Wahrscheinlichkeit größer wird als q. Bereits für $n = 100$ wird die Fehlerwahrscheinlichkeit geringer als die Wahrscheinlichkeit, dass kosmische Strahlung in unserem Computer zu einem falschen Ergebnis führt. Es gilt

$$\mathrm{P} \subseteq \mathrm{ZPP} \subseteq \mathrm{RP}$$

aber man weiß nicht, ob diese Inklusionen strikt sind. Ein Problem aus RP, von dem man aber nicht weiss, ob es auch in P (oder ZPP) liegt, wird im nächsten Abschnitt eingeführt.

5.7 Anwendung: Rechnen mit großen Zahlen

Betrachte ein Computerprogramm der folgenden eingeschränkten Form: es sind lediglich Befehle der folgenden vier Formen zugelassen.

1. $x := 1$,
2. $x := y + z$,
3. $x := y - z$, und

4. $x := y \cdot z$.

Beispiel 16. *Ein Beispiel eines solchen Programmes ist*

$$\begin{aligned} x_1 &:= 1 \\ x_2 &:= x_1 + x_1 \\ x_3 &:= x_2 + x_1 \\ x_4 &:= x_3 \cdot x_3 \\ x_0 &:= x_4 - x_1 \ . \end{aligned}$$

Wir nehmen an, dass jede Variable genau einmal auf der linken Seite einer solchen Zuweisung auftaucht, und dass sich jedes weitere Auftreten der Variablen später im Programm befindet. Daher sind zu jedem Zeitpunkt der Auswertung des Programmes die Variablen auf der rechten Seite einer Zuweisung bereits ausgewertet. Sei x_0 die Variable, die auf der linken Seite der letzten Zuweisung auftaucht. Wenn wir das Programm ausführen, wird dieser Variablen eine eindeutige ganze Zahl zugewiesen. Wir wollen gerne wissen, ob diese Zahl die Null ist.

In Beispiel (16) wertet x_1 zu 1 aus, x_2 zu 2, x_3 zu 3, x_4 zu 9, und x_0 zu 8.

Das Problem, zu gegebenem Programm festzustellen, ob die letzte Zuweisung im Programm einen Wert liefert, der von der Null verschieden ist, nennen wir *Test-auf-nicht-Null*. Das Problem mit diesem Problem ist, dass die Werte der Variablen sehr groß werden können, so dass wir exponentiell viel Zeit benötigen, um diese Werte explizit zu berechnen. Hierzu ein Beispiel.

$$\begin{aligned} x_1 &:= 1 \\ x_2 &:= x_1 + x_1 \\ x_3 &:= x_2 \cdot x_2 \\ x_4 &:= x_3 \cdot x_3 \\ x_5 &:= x_4 \cdot x_4 \\ \cdots &:= \cdots \\ x_n &:= x_{n-1} \cdot x_{n-1} \end{aligned}$$

Bei der Auswertung dieses Programms wird

- x_2 der Wert 2,
- x_3 der Wert 2^2,
- x_4 der Wert $2^2 \cdot 2^2 = 2^4$,
- x_5 der Wert $2^4 \cdot 2^4 = 2^8$,

und allgemein x_n für $n \geq 2$ der Wert $2^{2^{n-2}}$ zugewiesen, wie man leicht mit vollständiger Induktion zeigt. Die Binärdarstellung dieser Zahl hat die Länge 2^{n-2}: zu groß, um von einem polynomiellen Algorithmus ausgerechnet zu werden. Durch Anwendung von Subtraktion im Programm ist es jedoch nicht ausgeschlossen, dass die letzte Variable Null ist, auch wenn die Zwischenergebnisse sehr groß werden.

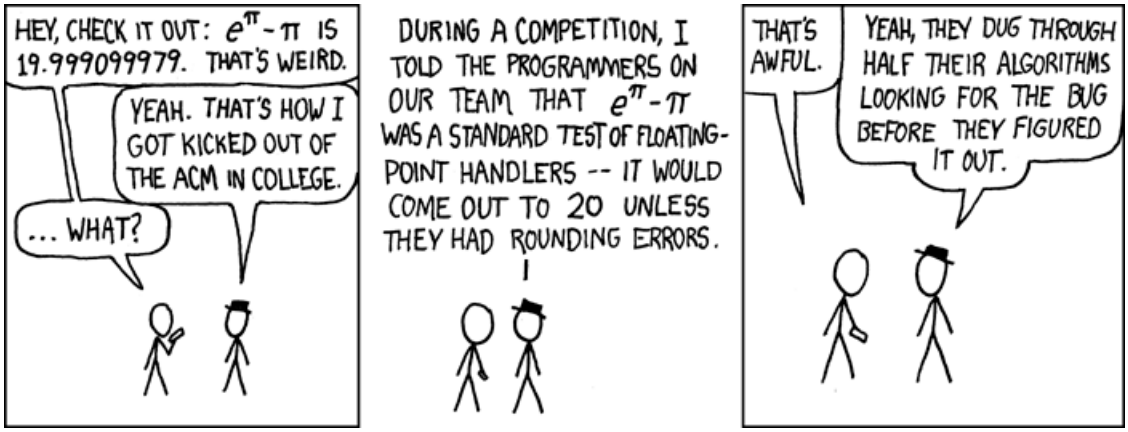


Abbildung 11: Heruntergeladen von http://xkcd.com/217.

Gibt es dennoch einen Algorithmus polynomieller Laufzeit, der das Problem Test-auf-nicht-Null löst? Dies ist ein wichtiges offenes Problem der theoretischen Informatik.

Proposition 32. *Das Problem 'Test-auf-nicht-Null' ist in RP. Das heißt, es gibt einen randomisierten Algorithmus mit garantiert polynomieller Laufzeit für das Problem Test-auf-nicht-Null:*

- *Falls die letzte Variable im Programm zu Null auswertet, so sagt das Programm garantiert 'Nein'.*
- *Ansonsten sagt das Programm mit Wahrscheinlichkeit von mindestens* $2/3$ *'Ja'.*

Der Beweis von Proposition 32 basiert auf zwei Ideen.

1. Um die Größenexplosion der Werte in den Variablen zu vermeiden, werden wir bei allen Rechenschritten modulo einer zufälligen Zahl m aus einem geeignet großen Bereich rechnen.
2. Um die Wahrscheinlichkeit für eine korrekte Antwort zu erhöhen, wiederholen wir die Rechnung hinreichend oft.

Wir wiederholen cn^2 mal:
    Wähle $m \in \{2, \ldots, 2^{n^2}\}$ gleichverteilt zufällig.
    Werte das Programm in jedem Schritt modulo m aus.
Falls alle Auswertungen Null ergeben
    gebe 'Nein' aus
ansonsten
    gebe 'Ja' aus.

Abbildung 12: Ein Monte-Carlo Algorithmus für das Problem 'Test-auf-nicht-Null'.

Bitte mehr Details. Wir verwenden den in Abbildung 12 angegebenen Algorithmus, für eine hinreichend große Konstante c. Falls die letzte Variable zu Null auswertet, dann sicher auch modulo m. In diesem Fall sagt der Algorithmus also garantiert 'Nein'. Falls die letzte Variable nicht zu Null auswertet, so kann man zeigen, dass mit Wahrscheinlichkeit von mindestens $2/3$ ein m existiert so dass das Programm modulo m *nicht* zu Null auswertet. In diesem Fall gibt der Algorithmus mit Wahrscheinlichkeit von mindestens $2/3$ 'Ja' aus. Die genaue Rechnung findet sich in [7] (Lemma 6-U) und verwendet den Primzahlsatz (Satz 25). $\square$

6 Gruppen

Eine *Gruppe* ist ein Tupel $(G; \circ, ^{-1}, e)$ bestehend aus

- einer Menge G, den *Gruppenelementen*;
- einer zweistelligen Operation $\circ\colon G^2 \to G$, der *Gruppenkomposition*;
- einer einstelligen Operation $^{-1}\colon G \to G$, der *Inversenbildung*; und
- dem *neutralen Element* $e \in G$;

mit den folgenden Eigenschaften:

- $a \circ (b \circ c) = (a \circ b) \circ c$ für alle $a, b, c \in G$ (die Komposition ist assoziativ);
- $g \circ e = g = e \circ g$ für alle $g \in G$; und
- $g \circ g^{-1} = g^{-1} \circ g = e$ für alle $g \in G$.

Für $g \in G$ nennt man g^{-1} das zu g *inverse Element*. Die Gestalt der Operationssymbole ist nebensächlich, oft werden zum Beispiel auch die Symbole $+, -, 0$ benutzt. Auch auf die Namen der Gruppenelemente kommt es nicht immer an.

Für ein Gruppenelement g und eine positive natürliche Zahl m bezeichnet g^m den Ausdruck $g \circ g \circ (\cdots) \circ g$, wobei im Ausdruck das Symbol g genau m Mal auftritt. Wir vereinbaren außerdem $g^0 := e$. Schließlich erlauben wir auch negative Exponenten, und definieren g^{-m} über die Inversenbildung als $g^{-m} := (g^m)^{-1}$.

Alternative Gruppendefinition. Durch die zweistellige Kompositionsoperation sind die anderen beiden Operationen einer Gruppe eindeutig bestimmt. Manche Autoren definieren deshalb eine Gruppe als ein Paar $(G; \circ)$ mit folgenden Eigenschaften:

- Komposition ist assoziativ;
- es gibt ein Element $e \in G$, so dass für alle $g \in G$ die Gleichung $e \circ g = g \circ e = g$ gilt;
- für alle $g \in G$ gibt es ein Element $g^{-1} \in G$ mit $g \circ g^{-1} = g^{-1} \circ g = e$.

Wir bevorzugen die erste Definition, weil sie logisch etwas einfacher ist, denn alle Bedingungen sind reine Gleichungen, die für alle Gruppenelemente gelten, und sich nur mit Hilfe der Gruppenoperationen hinschreiben lassen.

6.1 Beispiele

Wir haben bereits viele Beispiele von Gruppen kennengelernt:

- $(\mathbb{Z}; +, -, 0)$, wobei $-$ die Funktion $x \mapsto -x$ bezeichnen soll, ist eine Gruppe; ebenso wie $(\mathbb{Q}; +, -, 0)$, $(\mathbb{R}; +, -, 0)$, und $(\mathbb{C}; +, -, 0)$.
- $(\mathbb{Z}_n; +, -, 0)$, wobei $+$ und $-$ die im letzten Kapitel eingeführten Operationen auf $\mathbb{Z}_n$ bezeichnen.
- Die von Null verschiedenen rationalen Zahlen bilden bezüglich der Multiplikation ebenfalls eine Gruppe, ebenso wie die von Null verschiedenen reellen und komplexen Zahlen. Das neutrale Element ist $e := 1$, und das Inverse zu einer Zahl z aus diesen Mengen ist $1/z$.

Alle diese Gruppen sind *abelsch*, was das gleiche bedeutet wie kommutativ: sie erfüllen die Gleichung $a \circ b = b \circ a$ für alle Gruppenelemente a, b. Nichtabelsche Gruppen spielen ebenfalls eine große Rolle. Die Menge aller Permutationen einer Menge D auf sich selbst, mit der Hintereinanderausführung von Permutationen als Gruppenoperation, ist ebenfalls eine Gruppe. Sie wird die *symmetrische Gruppe* auf D genannt, und mit $\mathrm{Sym}(D)$ bezeichnet. Hat D mehr als zwei Elemente, dann ist die symmetrische Gruppe nicht abelsch.

Beispiel 17. *Es sei* $D := \{1, 2, 3\}$. *Betrachte die Permutationen* $a := \left(\begin{smallmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{smallmatrix}\right)$ *und* $b := \left(\begin{smallmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{smallmatrix}\right)$ *aus* $\mathrm{Sym}(D)$. *Dann ist* $a \circ b = \left(\begin{smallmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{smallmatrix}\right)$, *aber* $b \circ a = \left(\begin{smallmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{smallmatrix}\right)$. *Wir sehen also, dass* $\mathrm{Sym}(\{1, 2, 3\})$ *nicht abelsch ist.*

Beispiel 18. *Die Symmetrien eines Quadrats bilden eine Gruppe. Ein Quadrat besitzt die folgenden Symmetrien: Horizontale Spiegelung* $\leftrightarrow$, *vertikale Spiegelung* $\updownarrow$, *zwei diagonale Spieglungen* $\nwarrow\!\!\searrow$ *und* $\swarrow\!\!\nearrow$, *die Drehung* $\circlearrowleft$ *gegen den Uhrzeigersinn um 90 Grad, die Drehung* $\curvearrowleft$ *um 180 Grad, die Drehung* $\circlearrowright$ *gegen den Uhrzeigersinn um 270 Grad, und die identische Abbildung. Um die Symmetrien zu beschreiben, benennen wir die Eckpunkte des Quadrats* a, b, c, d; *hierbei ist* a *oben links, und die übrigen Elemente folgen gegen den Uhrzeigersinn.*

	Bedeutung	*Eckenpermutation*
id	*identische Abbildung*	$\left(\begin{smallmatrix} a & b & c & d \\ a & b & c & d \end{smallmatrix}\right)$
$\circlearrowleft$	*Drehung um 90 Grad*	$\left(\begin{smallmatrix} a & b & c & d \\ b & c & d & a \end{smallmatrix}\right)$
$\curvearrowleft$	*Drehung um 180 Grad*	$\left(\begin{smallmatrix} a & b & c & d \\ c & d & a & b \end{smallmatrix}\right)$
$\circlearrowright$	*Drehung um 270 Grad*	$\left(\begin{smallmatrix} a & b & c & d \\ d & a & b & c \end{smallmatrix}\right)$
$\leftrightarrow$	*Vertikale Spiegelung*	$\left(\begin{smallmatrix} a & b & c & d \\ d & c & b & a \end{smallmatrix}\right)$
$\updownarrow$	*Horizontale Spiegelung*	$\left(\begin{smallmatrix} a & b & c & d \\ b & a & d & c \end{smallmatrix}\right)$
$\nwarrow\!\!\searrow$	*Spiegelung mit Achse bd*	$\left(\begin{smallmatrix} a & b & c & d \\ c & b & a & d \end{smallmatrix}\right)$
$\swarrow\!\!\nearrow$	*Spiegelung mit Achse ac*	$\left(\begin{smallmatrix} a & b & c & d \\ a & d & c & b \end{smallmatrix}\right)$

Die folgende Tabelle zeigt für jedes Element sein Inverses an.

x	id	$\circlearrowleft$	$\curvearrowleft$	$\circlearrowright$	$\leftrightarrow$	$\updownarrow$	$\nwarrow\!\!\searrow$	$\swarrow\!\!\nearrow$
x^{-1}	id	$\circlearrowright$	$\curvearrowleft$	$\circlearrowleft$	$\leftrightarrow$	$\updownarrow$	$\nwarrow\!\!\searrow$	$\swarrow\!\!\nearrow$

Und schließlich die Kompositionstabelle.

$\circ$	id	$\circlearrowleft$	$\curvearrowleft$	$\circlearrowright$	$\leftrightarrow$	$\updownarrow$	$\nwarrow\!\!\searrow$	$\swarrow\!\!\nearrow$
id	id	$\circlearrowleft$	$\curvearrowleft$	$\circlearrowright$	$\leftrightarrow$	$\updownarrow$	$\nwarrow\!\!\searrow$	$\swarrow\!\!\nearrow$
$\circlearrowleft$	$\circlearrowleft$	$\curvearrowleft$	$\circlearrowright$	id	$\swarrow\!\!\nearrow$	$\nwarrow\!\!\searrow$	$\updownarrow$	$\leftrightarrow$
$\curvearrowleft$	$\curvearrowleft$	$\circlearrowright$	id	$\circlearrowleft$	$\updownarrow$	$\leftrightarrow$	$\swarrow\!\!\nearrow$	$\nwarrow\!\!\searrow$
$\circlearrowright$	$\circlearrowright$	id	$\circlearrowleft$	$\curvearrowleft$	$\nwarrow\!\!\searrow$	$\swarrow\!\!\nearrow$	$\leftrightarrow$	$\updownarrow$
$\leftrightarrow$	$\leftrightarrow$	$\swarrow\!\!\nearrow$	$\updownarrow$	$\nwarrow\!\!\searrow$	id	$\curvearrowleft$	$\circlearrowright$	$\circlearrowleft$
$\updownarrow$	$\updownarrow$	$\nwarrow\!\!\searrow$	$\leftrightarrow$	$\swarrow\!\!\nearrow$	$\curvearrowleft$	id	$\circlearrowleft$	$\circlearrowright$
$\nwarrow\!\!\searrow$	$\nwarrow\!\!\searrow$	$\leftrightarrow$	$\swarrow\!\!\nearrow$	$\updownarrow$	$\circlearrowleft$	$\circlearrowright$	id	$\curvearrowleft$
$\swarrow\!\!\nearrow$	$\swarrow\!\!\nearrow$	$\updownarrow$	$\nwarrow\!\!\searrow$	$\leftrightarrow$	$\circlearrowright$	$\circlearrowleft$	$\curvearrowleft$	id

Diese Gruppe hat den Namen D_4, eine sogenannte Diedergruppe. *Auch diese Gruppe ist nicht abelsch (warum?).*

6.2 Die multiplikative Gruppe von $\mathbb{Z}_n$

Wie steht es mit der Multiplikation in $\mathbb{Z}_n$? Klarerweise ist die Multiplikation assoziativ, und wir haben mit der 1 ein Element, das für jedes $x \in \mathbb{Z}_n$ die Gleichung $1 \cdot x = x \cdot 1 = x$ erfüllt. In Gruppen muß es allerdings auch zu jedem Element ein Inverses geben.

Beispiel 19. *Wir betrachten die Multiplikation in $\mathbb{Z}_6$. Wenn $2 \in \mathbb{Z}_6$ ein Inverses i besitzen würde, dann müßte gelten $2 \cdot i = 1$. Allerdings erhalten wir dann die widersprüchliche Gleichung*

$$3 \equiv 3 \cdot 1 \equiv 3 \cdot 2 \cdot i \equiv 6 \cdot i = 0 \pmod 6 \,.$$

Definition 33 (Nullteiler)**.** *Man nennt $a \in \mathbb{Z}_n \setminus \{0\}$ einen* Nullteiler*, wenn es ein $b \in \mathbb{Z}_n \setminus \{0\}$ gibt mit $a \cdot b = 0$.*

Wie wir in Beispiel 19 gesehen haben, ist 2 ein Nullteiler in $\mathbb{Z}_6$. Und 3 ist ebenfalls ein Nullteiler in $\mathbb{Z}_6$, da die Multiplikation in $\mathbb{Z}_6$ kommutativ ist.

Definition 34 (Einheit)**.** *Man nennt $a \in \mathbb{Z}_n$ eine* Einheit*, wenn es ein $b \in \mathbb{Z}_n \setminus \{0\}$ gibt mit $a \cdot b = 1$.*

Selbstverständlich ist damit 0 keine Einheit in $\mathbb{Z}_n$. Durch Einheiten kann man 'dividieren', denn b verhält sich wie ein Kehrwert zu a. Man sagt, b sei *multiplikativ invers* zu a. Man dividiert durch a, indem man mit b multipliziert. Welche Zahlen sind Einheiten modulo n?

Lemma 35. *Sei $m \in \mathbb{Z}_n \setminus \{0\}$. Dann sind äquivalent:*

1. *m ist Einheit in $\mathbb{Z}_n$;*

2. *m ist kein Nullteiler in $\mathbb{Z}_n$;*

3. *m und n sind teilerfremd.*

Beweis. $1 \Rightarrow 2$. Wir zeigen die Kontraposition: Angenommen m ist Nullteiler in $\mathbb{Z}_n$. Das heißt, es gibt $b \in \mathbb{Z}_n \setminus \{0\}$ mit $m \cdot b = 0$. Dann wäre $0 = m^{-1} \cdot 0 = m^{-1} \cdot m \cdot b = b$, also wäre m keine Einheit in $\mathbb{Z}_n$.

$2 \Rightarrow 3$. Wieder zeigen wir die Kontraposition. Angenommen, 3. gilt nicht, das heißt, $\mathrm{ggT}(m, n) > 1$. Dann ist

$$m' := n / \mathrm{ggT}(m, n)$$

eine Zahl aus $\mathbb{Z}_n \setminus \{0\}$ so dass $m \cdot m'$ ein Vielfaches ist von n. Also $m \cdot m' \equiv 0 \pmod{n}$, und m ist ein Nullteiler.

$3 \Rightarrow 1$. Wenn $\mathrm{ggT}(m, n) = 1$, dann existieren nach dem Lemma von Bézout $a, b \in \mathbb{Z}$ mit $1 = a \cdot m + b \cdot n$. Also ist a multiplikativ invers zu m, denn

$$m \cdot a \equiv 1 - b \cdot n \equiv 1 \pmod{n}\,. \qquad \square$$

Es folgt aus dem Beweis von Lemma 35 ($3. \Rightarrow 1.$) dass sich das multiplikative Inverse einer Einheit in $\mathbb{Z}_n$ mit dem Erweiterten Euklidischen Algorithmus berechnen lässt (siehe Abschnitt 4.5).

Die Menge der Einheiten modulo n bildet mit der Multiplikation modulo n eine Gruppe:

- Multiplikation ist auf ganz $\mathbb{Z}_n$ assoziativ, also insbesondere auf den Einheiten;
- $e := 1$ ist eine Einheit, und das neutrale Element der Gruppe;
- jedes Element a besitzt ein Inverses a^{-1} (nach Definition der Einheiten), und dieses Inverse ist ebenfalls wieder eine Einheit, weil a dazu invers ist.
- wenn a und b Einheiten in $\mathbb{Z}_n$ sind, und a^{-1} und b^{-1} die entsprechenden Inversen, dann ist $b^{-1} \cdot a^{-1}$ multiplikativ invers zu $a \cdot b$, denn

$$(a \cdot b) \cdot (b^{-1} \cdot a^{-1}) = a \cdot (b \cdot b^{-1}) \cdot a^{-1} = 1\,.$$

Diese Gruppe heißt *Einheitengruppe* von $\mathbb{Z}_n$, und wird mit $\mathbb{Z}_n^*$ (soll heißen $(\mathbb{Z}_n^*, \cdot, ^{-1}, 1)$) bezeichnet.

Beispiel 20. *Es gilt $\mathbb{Z}_6^* = \{1, 5\}$. Die Multiplikationstabelle ist*

$\cdot$	*1*	*5*
1	*1*	*5*
5	*5*	*1*

und die 5 ist in dieser Gruppe zu sich selbst invers.

Wie viele Elemente hat $\mathbb{Z}_n^*$? Dazu müssen wir die Anzahl aller zu n teilerfremden Zahlen in $\mathbb{Z}_n$ bestimmen (siehe Lemma 35). Diese Anzahl wird auch mit $\phi(n)$ bezeichnet; und die Funktion $n \mapsto \phi(n)$ nennt man die *eulersche ϕ-Funktion*[30]. Die ersten Werte dieser Funktion sind:

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\phi(n)$	1	1	2	2	4	2	6	4	6	4	10	4	12	6	8	8

Der Wert $\phi(1) = 1$ erklärt sich dadurch, dass die Zahl 1 (als der Wert des leeren Produktes) auch zu sich selbst teilerfremd ist.

Beobachtungen: Falls p prim ist, gilt

- $\phi(p) = p - 1$;
- $\phi(p^k) = (p-1)p^{k-1} = p^k - p^{k-1}$: denn $\mathrm{ggT}(p, a) > 1$ gilt genau dann, wenn

$$a \in \{p, 2p, 3p, \dots, p^{k-2}p\},$$

  und $p^{k-1}p = p^k \equiv 0 \mod p^k$ fehlt ebenfalls in $\mathbb{Z}_{p^k}^*$.

Im allgemeinen haben wir die folgende Aussage.

Proposition 36. *Hat $n \in \mathbb{N}$ die Primfaktorzerlegung $n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot (\dots) \cdot p_k^{\alpha_k}$, dann gilt*

$$\begin{aligned} \phi(n) &= \phi(p_1^{\alpha_1}) \cdot \dots \cdot \phi(p_k^{\alpha_k}) && (4)\\ &= (p_1 - 1){p_1}^{\alpha_1 - 1} \cdot (\cdots) \cdot (p_k - 1){p_k}^{\alpha_k - 1} && (5)\\ &= n \cdot (1 - 1/p_1) \cdot (\cdots) \cdot (1 - 1/p_k)\,. && (6) \end{aligned}$$

Beispiel: $\phi(240) = \phi(2^4 \cdot 3 \cdot 5) = 2^3 \cdot 2 \cdot 4 = 64$.

Beweis. Sei $n = n_1 n_2$ mit $\mathrm{ggT}(n_1, n_2) = 1$. Wir werden zeigen, dass $\phi(n) = \phi(n_1) \cdot \phi(n_2)$; per vollständiger Induktion folgt damit die Gleichung (4). Definiere $\alpha\colon \mathbb{Z}_n^* \to \mathbb{Z}_{n_1}^* \times \mathbb{Z}_{n_2}^*$ wie folgt:

$$\alpha(m) = (m \bmod n_1, m \bmod n_2).$$

Hier muss man sich davon überzeugen, dass $m \bmod n_1$ tatsächlich in $\mathbb{Z}_{n_1}^*$ liegt: das ist der Fall, weil $m \bmod n_1$ teilerfremd ist zu n_1; analog sieht man, dass $m \bmod n_2$ in $\mathbb{Z}_{n_2}^*$ liegt. Sei nun $(m_1, m_2) \in \mathbb{Z}_{n_1}^* \times \mathbb{Z}_{n_2}^*$; dann gibt es nach dem chinesischen Restsatz (Satz 31) genau ein $m \in \{0, \dots, n-1\}$ mit $\alpha(m) = (m_1, m_2)$. Dann liegt m in $\mathbb{Z}_n^*$, da

[30] Leonhard Euler; geboren am 15. April 1707 in Basel; gestorben am 7. September 1783 in Sankt Petersburg.

$\mathrm{ggT}(m,n) = \mathrm{ggT}(m, n_1 n_2) = 1$. Also hat α eine Umkehrfunktion, d.h., α is bijektiv, und damit gilt

$$\begin{aligned}\phi(n) &= |\mathbb{Z}_n^*| \\ &= |\mathbb{Z}_{n_1}^* \times \mathbb{Z}_{n_1}^*| && \text{(da } \alpha \text{ bijektiv)} \\ &= \phi(n_1) \cdot \phi(n_2).\end{aligned}$$

Gleichung (5) folgt aus den Beobachtungen von vor der Proposition, und Gleichung (6) erhalten wir durch reine Umformung. □

Bemerkung 37. *Einheiten und Nullteiler können nicht nur für $\mathbb{Z}_n$, sondern für beliebige* Ringe *definiert werden. Der Begriff eines Rings verallgemeinert den Begriff des Körpers, wie Sie ihn in der linearen Algebra kennenlernen, und wird Ihnen später im Studium wieder begegnen.*

6.3 Zyklische Gruppen

Zyklische Gruppen sind die einfachsten Gruppen.

Definition 38. *Eine Gruppe G heißt* zyklisch *falls sie von einem Element* erzeugt *wird, das heißt, es gibt ein Gruppenelement $g \in G$ (den* Erzeuger*) so dass sich G schreiben läßt als*

$$G = \{e, g, g^{-1}, g \circ g, (g \circ g)^{-1}, \dots\} = \{g^m \mid m \in \mathbb{Z}\}.$$

Zyklische Gruppen können vollständig klassifiziert werden. Um die Klassifikation angeben zu können, benötigen wir den Begriff des *Gruppenisomorphismus*. Informell sind zwei Gruppen isomorph wenn sie sich durch Umbenennen der Elemente ineinander überführen lassen; die formale Definition lautet wie folgt.

Definition 39. *$(G; \circ, ^{-1}, e)$ und $(H; \circ, ^{-1}, e)$ heißen* isomorph *falls es eine Bijektion $f\colon G \to H$ gibt so dass für alle $g_1, g_2 \in G$ gilt dass $f(g_1 \circ g_2) = f(g_1) \circ f(g_2)$.*

Man sieht zum Beispiel leicht, dass jede zyklische Gruppe entweder isomorph ist zu $(\mathbb{Z}; +, -, 0)$ oder zu $(\mathbb{Z}_n; +, -, 0)$ für ein $n \in \mathbb{N}$.

Proposition 40. *Die Anzahl der Erzeuger von $(\mathbb{Z}_n, +, -, 0)$ ist $\phi(n)$.*

Beweis. Sei $m \in \mathbb{Z}_n$. Falls $\mathrm{ggT}(m,n) > 1$, so ist m kein Erzeuger, denn $\mathrm{ggT}(m,n)$ ist ein Teiler von

$$km := \underbrace{m + \cdots + m}_{n-\text{mal}}$$

für alle $k \in \mathbb{Z}$. Also werden Elemente, die nicht von $\mathrm{ggT}(m,n)$ geteilt werden, nicht erzeugt.

Falls $\mathrm{ggT}(m,n)=1$, dann gibt es nach Bézout $a,b\in\mathbb{Z}$ mit $am+bn=1$. Ein Element $c\in\mathbb{Z}_n$ kann also mit m erzeugt werden durch

$$\begin{aligned} cam &\equiv c - cbn \\ &\equiv c \qquad\qquad\qquad\qquad (\bmod n). \end{aligned}$$

Also gibt es $\phi(n)$ viele Erzeuger von $(\mathbb{Z}_n,+,-,0)$. □

Der folgende Satz wurde bereits von Lambert und Euler vermutet, aber vollständig erstmals von Gauß bewiesen.

Satz 41 (Gauß)**.** *Sei p eine Primzahl. Dann ist $G=(\mathbb{Z}_p^*;\cdot,^{-1},1)$ zyklisch.*

Beweis für Interessierte; setzt den Begriff des Polynoms voraus. Wir zeigen zunächst, dass ein Polynom $p(x)$ vom Grad d höchstens d Nullstellen in $(\mathbb{Z}_p^*;+,-,0,\cdot,^{-1},1)$ besitzt. Wir führen diesen Beweis per Induktion über $d\geq 1$. Ein Polynom vom Grad 1 ist von der Gestalt $ax+b$ für $a,b\in\mathbb{Z}_p^*$ mit $a\neq 0$. Dieses Polynom hat genau eine Nullstelle, nämlich $-b\cdot a^{-1}$. Wir nehmen nun an, dass die Aussage gilt für Polynome vom Grad d, und zeigen die Aussage für Polynome vom Grad $d+1$. Ein solches Polynom $p(x)$ kann geschrieben werden als

$$c_{d+1}x^{d+1}+c_dx^d+\cdots+c_1x+c_0$$

wobei $c_1,\ldots,c_{d+1}\in\mathbb{Z}_p^*$ und $c_{d+1}\neq 0$.

Fall 1. $p(x)$ has keine Nullstellen in $\mathbb{Z}_p^*$. Da $0\leq d+1$ bleibt hier nichts zu zeigen.

Fall 2. $p(x)$ has eine Nullstelle r in $\mathbb{Z}_p^*$. Dann lässt sich $p(x)$ schreiben als $(x-r)g(x)$ wobei $g(x)$ ein Polynom vom Grad d ist[31] Nach Induktionsvoraussetzung hat $g(x)$ höchstens d Nullstellen. Wir verwenden nun das Lemma von Euklid (Lemma 30), welches mit dem gleichen Beweis wie in Abschnitt 4.5 auch für $\mathbb{Z}_n$ anstatt natürlicher Zahlen gilt: wenn $a\in\mathbb{Z}_n$ so dass $(a-r)\cdot g(a)=0$, dann muss bereits $a-r=0$ oder $g(a)=0$ gelten. Also hat $p(x)$ höchstens $d+1$ Nullstellen.

Die *Ordnung* eines Elementes $a\in\mathbb{Z}_p^*$ ist die kleinste Zahl $n\in\mathbb{N}$, so dass $a^n=1$ (dieser Begriff wird uns wieder in Abschnitt 6.5 begegnen). Sei m die maximale Ordnung der Elemente von $\mathbb{Z}_p^*$. Es genügt zu zeigen, dass $m=p-1$, denn ein Element der Ordnung $p-1$ muss ein Erzeuger von $\mathbb{Z}_p^*$ sein. Die Ordnung n jedes Elementes $a\in\mathbb{Z}_p^*$ muss m teilen: denn wenn $g\in\mathbb{Z}_p^*$ die Ordnung m hat, und m von n nicht geteilt wird, dann gibt es eine Primzahl q so dass der Exponent e von q in der Primfaktorzerlegung von m kleiner ist als der Exponent f von q in der Primfaktorzerlegung von n. Dann hat das Element $g^{p^e}a^{n/p^f}$ die Ordnung $\frac{m}{p^e}p^f=mp^{f-e}>m$, im Widerspruch zur Maximalität von m.

Also gilt $a^m=1$, und das Polynom $x^m=1$ hat $p-1$ Nullstellen. Nach der eingangs gezeigten Aussage hat dieses Polynom aber höchstens m Nullstellen. Also gilt $p-1\leq m$. Offensichtlich ist $m\leq p-1$, also haben wir $m=p-1$ und die Aussage ist gezeigt. □

[31] Schreibe $p(x)=c_n(x^n-r^n)+c_{n-1}(x^{n-1}-r^{n-1})+\cdots+c_1(x-r)$ und klammere $(x-r)$ aus.

Was ist ein Erzeuger von $\mathbb{Z}_7^*$? Die 2 ist es nicht, denn wir erreichen die 3, 5, und 6 nicht:

m	0	1	2	3	4	5
2^m	1	2	4	1	2	4

Aber die 3 tut es:

m	0	1	2	3	4	5
3^m	1	3	2	6	4	5

Die Erzeuger von $\mathbb{Z}_n^*$ heissen auch *Primitivwurzeln von* $\mathbb{Z}_n^*$. Die folgende Tabelle zeigt die Primitivwurzeln von $\mathbb{Z}_n^*$ für $n \in \{2, 3, 5, 7, 11, 13, 17, 19\}$.

n	$\phi(\phi(n))$	Primitivwurzeln
2	1	1
3	1	2
5	2	2, 3
7	2	3, 5
11	4	2, 6, 7, 8
13	4	2, 6, 7, 11
17	8	3, 5, 6, 7, 10, 11, 12, 14
19	6	2, 3, 10, 13, 14, 15

Proposition 42. *Sei p eine Primzahl. Die Anzahl der Erzeuger von $\mathbb{Z}_p^*$ ist $\phi(\phi(p))$.*

Beweis. Wegen Satz 41 ist $\mathbb{Z}_p^*$ (die multiplikative Gruppe) isomorph zu $(\mathbb{Z}_{\phi(p)}, +, -, 0)$; nach Proposition 40 hat diese Gruppe $\phi(\phi(p))$ Erzeuger. Also gibt es auch $\phi(\phi(p))$ viele Erzeuger von $\mathbb{Z}_p^*$. □

Für viele einfach zu stellenden Fragen über Primitivwurzeln kennt man die Antwort nicht. Zum Beispiel ist folgende Vermutung offen.

Vermutung 1 (Ein Spezialfall der Artin'schen Vermutung[32])**.** *Es gibt unendlich viele Primzahlen p so dass die 2 Primitivwurzel ist in $\mathbb{Z}_p^*$.*

6.4 Öffentlich ein Geheimnis vereinbaren

Zwei Teilnehmer:innen möchten abhörsicher miteinander kommunizieren und ein Verschlüsselungsverfahren benutzen. Dazu wollen sie einen gemeinsamen geheimen Schlüssel verwenden. Wie können sie sich über eine nicht abhörsichere Verbindung auf ein gemeinsames Geheimnis einigen? Wir stellen hier das Verfahren von Diffie-Hellman-Merkle vor.

[32] Emil Artin; geboren am 3. März 1898 in Wien; gestorben am 20. Dezember 1962 in Hamburg.

1. A und B einigen sich zunächst öffentlich auf eine große Primzahl p und eine Primitivwurzel g von $\mathbb{Z}_p^*$.

2. A erzeugt eine zufällige geheime große Zahl a und berechnet $a' := g^a \bmod p$.

3. B erzeugt eine zufällige geheime große Zahl b und berechnet $b' := g^b \bmod p$.

4. A und B teilen sich die Zahlen a' und b' mit.

5. A berechnet $c := (b')^a \mod p$.
   Bemerke: $c = (b')^a \equiv (g^b)^a = g^{a \cdot b} \pmod p$

6. B berechnet $(a')^b \mod p$.
   Bemerke: $(a')^b \equiv (g^a)^b = g^{a \cdot b} \equiv c \pmod p$.

Die Zahl c ist das gewünschte Geheimnis, denn es ist kein Verfahren bekannt, aus p, g, a' und b' in vernünftiger Zeit das Geheimnis c zu berechnen.

Interessanterweise brauchen im Verfahren a und b nicht weiter gespeichert zu werden, sondern können gelöscht werden.

Zum Knacken des Verfahrens von Diffie-Hellman-Merkle würde es natürlich genügen, aus p, g, und a' die Zahl a berechnen zu können. Auch dafür ist kein effizientes Verfahren bekannt. Das Problem, aus der Angabe von $g^a \in \mathbb{Z}_p^*$ den Exponenten a zu bestimmen, wird auch das Problem vom *diskrete Logarithmus* genannt. Formal ist der diskrete Logarithmus zur Basis g von einem Element $x \in \mathbb{Z}_p^*$ die kleinste Zahl $m \in \{0, \ldots, p-2\}$ mit der Eigenschaft dass $x = g^m$. Da sich umgekehrt aus g und m die Zahl $g^m \bmod p$ schnell berechnen läßt (siehe Abschnitt 5.4), ist die diskrete Exponentialfunktion eine *Einbahnfunktion*: die Funktion lässt sich effizient berechnen, aber die Umkehrfunktion nach dem derzeitigen Stand der Kunst nicht.

Bemerkung 43. *Wenn sich A und B öffentlich auf ein Geheimnis c einigen können, dann können Sie sich auch geheime Botschaften schicken. Es sei s_i das i-te Bit der geheimen Botschaft s; es ist also $s_i \in \mathbb{Z}_2$. Wir nehmen an, dass c in Binärdarstellung mehr Bits besitzt als die geheime Nachricht und schreiben c_i für das i-te Bit der Binärdarstellung von c. Dann verschickt A für jedes i die Zahl $t_i := s_i + c_i \mod 2$ an B. Daraufhin kann B das geheime i-te Bit s_i mit Hilfe von $t_i + c_i = (s_i + c_i) + c_i \equiv s_i \pmod 2$ berechnen. Selbstverständlich muss für jede zu verschlüsselnde Nachricht ein neues gemeinsames Geheimnis c berechnet werden, damit das Verfahren sicher bleibt.*

6.5 Der Satz von Lagrange

Sei $(G; \circ, ^{-1}, e)$ eine Gruppe. Eine *Untergruppe* von G ist eine Teilmenge U von G, die das neutrale Element e enthält, und die unter $^{-1}$ und $\circ$ *abgeschlossen* ist. Das soll heissen, dass mit jedem Element $g \in U$ auch $g^{-1} \in U$, und dass für alle $g_1, g_2 \in U$ auch $g_1 \circ g_2 \in U$. Jede Untergruppe U von G ist, ausgestattet mit den auf U eingeschränkten Operationen $\circ$ und

$^{-1}$, und demselben neutralen Element e, selbst wieder eine Gruppe. Um anzuzeigen, dass U eine Untergruppe von G ist, schreibt man $U \leq G$.

Beispiel 21. *Wir betrachten die Gruppe der Symmetrien des Quadrats aus Beispiel 18, mit den Elementen* $\{\mathrm{id}, \circlearrowleft, \curvearrowleft, \circlearrowright, \leftrightarrow, \updownarrow, \nwarrow\!\!\searrow, \swarrow\!\!\nearrow\}$. *Wie man leicht anhand der Tabelle für die Inversenbildung und der Kompositionstabelle feststellt, ist* $\{\mathrm{id}, \circlearrowleft, \curvearrowleft, \circlearrowright\}$ *eine Untergruppe.*

Zu jeder Teilmenge T von G gibt es eine kleinste Untergruppe, die T enthält. Man nennt sie die *von T erzeugte Untergruppe* $\langle T \rangle$.

Beispiel 22. *In Beispiel 18 erzeugt* $T := \{\circlearrowleft\}$ *die Untergruppe* $\langle T \rangle = \{\mathrm{id}, \circlearrowleft, \curvearrowleft, \circlearrowright\}$, *die Untergruppe der Drehungen. Für* $T := \{\circlearrowleft, \leftrightarrow\}$ *ist* $\langle T \rangle$ *bereits die volle Gruppe. Das heißt, dass sich alle acht Gruppenelemente aus* $\circlearrowleft$ *und* $\leftrightarrow$ *durch Komposition und Inversenbildung gewinnen lassen.*

Die Anzahl $|G|$ der Gruppenelemente nennt man auch die *Ordnung* von G. Als die *Ordnung eines Elementes* a einer Gruppe G bezeichnet man die Ordnung der von $\{a\}$ erzeugten Untergruppe von G. Die Ordnung eines Elementes a ist also entweder unendlich, oder eine natürliche Zahl m; in letzterem Fall ist dies auch die kleinste natürliche Zahl $m > 0$ mit $a^m = 1$.

Beispiel 23. *In Beispiel 18 hat das Element* $\circlearrowleft$ *die Ordnung vier. Die Spiegelungen* $\leftrightarrow$, $\updownarrow$, $\nwarrow\!\!\searrow$, $\swarrow\!\!\nearrow$ *haben allesamt Ordnung zwei.*

Definition 44 (Nebenklassen)**.** *Ist U eine Untergruppe der Gruppe G, und g ein Element von G, dann nennt man* $g \circ U := \{g \circ u \mid u \in U\}$ *eine* (Links-) Nebenklasse *von U in G.*

Beispiel 24. *In Beispiel 18 gibt es zur Untergruppe U der Drehungen*

$$U = \{\mathrm{id}, \circlearrowleft, \curvearrowleft, \circlearrowright\}$$

genau zwei Nebenklassen, nämlich U selbst, und die Menge der Spiegelungen $\{\leftrightarrow, \updownarrow, \nwarrow\!\!\searrow, \swarrow\!\!\nearrow\}$. *Die Nebenklasse der Spiegelungen ist keine Untergruppe (sie enthält nicht das neutrale Element* id*).*

Wir wollen zeigen, dass zwei Linksnebenklassen von U entweder disjunkt sind oder gleich. Dazu benötigen wir das folgende Lemma.

Lemma 45. *Es sei U eine Untergruppe von G und $g_1, g_2 \in G$. Falls $g_1 \in g_2 \circ U$, dann gilt $g_1 \circ U = g_2 \circ U$.*

Beweis. Wenn $g_1 \in g_2 \circ U$, so gibt es ein $u \in U$ mit $g_1 = g_2 \circ u$.

'$\subseteq$': Sei $h_1 \in g_1 \circ U$ beliebig. Dann gibt es ein $u' \in U$ mit $h_1 = g_1 \circ u'$. Da U eine Untergruppe ist, so ist $u \circ u' \in U$, und daher gilt $h_1 = g_1 \circ u' = g_2 \circ u \circ u' \in g_2 \circ U$. Es folgt $g_1 \circ U \subseteq g_2 \circ U$.

'$\supseteq$': Sei nun $h_2 \in g_2 \circ U$ beliebig. Dann gibt es ein $u'' \in U$ mit $h_2 = g_2 \circ u''$. Da U eine Untergruppe ist, so ist $u^{-1} \circ u'' \in U$, und daher gilt $h_2 = g_2 \circ u'' = g_1 \circ u^{-1} \circ u'' \in g_1 \circ U$. Es folgt $g_2 \circ U \subseteq g_1 \circ U$, und damit $g_1 \circ U = g_2 \circ U$. $\square$

Lemma 46. *Je zwei Nebenklassen $a \circ U$ und $b \circ U$ sind entweder gleich oder disjunkt.*

Beweis. Wenn $a \circ U$ und $b \circ U$ nicht disjunkt sind, gibt es ein $x \in a \circ U \cap b \circ U$. Also gilt $a \circ U = x \circ U = b \circ U$ nach Lemma 45. □

Definition 47 (Index)**.** *Es sei G eine Gruppe und U eine Untergruppe von G. Der* Index *von U in G ist die Anzahl der Nebenklassen von U in G, und wird $[G : U]$ geschrieben.*

Man kann den Index leicht bestimmen, wenn man beachtet, dass jede Nebenklasse von U die gleiche Mächtigkeit hat wie U. Das ist deshalb richtig, weil die Abbildung $U \to g \circ U$ mit $u \mapsto g \circ u$ stets bijektiv ist. Die Nebenklassen von U zerlegen also die Menge G in gleichgroße disjunkte Teilmengen.

Satz 48 (Satz von Lagrange[33])**.** *Ist U eine Untergruppe einer endlichen Gruppe G, dann gilt $[G : U] = |G|/|U|$.*

Weil $[G : U]$ ganzzahlig ist, teilt also die Ordnung einer Untergruppe einer endlichen Gruppe G stets die Ordnung der Gruppe. Es folgt, dass die Ordnung eines Elementes $a \in G$ ein Teiler ist von $|G|$. Ausserdem gilt $a^{|G|} = e$: denn wenn $U = \langle\{a\}\rangle$ die von a erzeugte (zyklische!) Untergruppe ist, dann ist

$$a^{|G|} = a^{|U| \cdot [G:U]} = e^{[G:U]} = e\,.$$

6.6 Das Lemma von Euler-Fermat

Eine weitere Konsequenz des Satzes von Lagrange ist eine zahlentheoretische Aussage, die in der Kryptographie eine Rolle spielt, nämlich das Lemma von Euler-Fermat. Wir stellen hier zuerst einen Spezialfall vor, den sogenannten *kleinen Fermat*.

Lemma 49 (Lemma von Fermat[34])**.** *Ist p eine Primzahl, dann gilt für jede ganze Zahl a, die nicht durch p teilbar ist,*

$$a^{p-1} \equiv 1 \pmod{p}\,.$$

Beweis. Die Gruppe $\mathbb{Z}_p^*$ hat $p-1$ Elemente, nämlich $\{1, \ldots, p-1\}$. Da a nicht durch p teilbar ist, ist $b := a \mod p$ ein Element dieser Gruppe. Wie wir im letzten Kapitel angemerkt haben, gilt $b^{|G|} = 1$, und damit $a^{p-1} \equiv 1 \pmod{p}$. □

Wir können das Lemma von Fermat dazu verwenden, um Potenzen modulo p ganz leicht auszurechnen. Ein Beispiel hierzu.

[33] Joseph-Louis de Lagrange; geboren am 25. Januar 1736 in Turin als Giuseppe Lodovico Lagrangia; gestorben am 10. April 1813 in Paris.

[34] Pierre de Fermat; geboren im Jahre 1607 in Beaumont-de-Lomagne, Tarn-et-Garonne; gestorben am 12. Januar 1665 in Castres.

Beispiel 25. *Die Zahl* 997 *ist eine Primzahl. Deshalb gilt*

$$2^{1000} \equiv 2^{996} \cdot 2^4 \equiv 16 \pmod{997}\,.$$

Man kann mit Hilfe des Lemmas von Fermat auch beweisen, dass manche Zahlen keine Primzahlen sind, ohne einen Teiler anzugeben. Ein Beispiel dazu.

Beispiel 26. *Ist* 100001 *eine Primzahl? Wenn ja, dann muss für jede Zahl* $0 < a < 100001$ *folgendes gelten:*

$$a^{100000} \equiv 1 \mod 100001\,.$$

Für $a := 2$ *hatten wir diese Rechnung in Abschnitt 5.4 schon mit der Methode der binären Exponentiation durchgeführt und ausgerechnet, dass*

$$a^{100000} \equiv 1024 \not\equiv 1 \mod 100001\,.$$

100001 *ist also keine Primzahl.*

Das Lemma von Fermat gilt nur für Primzahlen p. Im Beweis wurde benutzt, dass die Zahlen $\{1, 2, \ldots, p-1\}$ bezüglich der Multiplikation modulo p eine Gruppe bilden. Für diese Gruppe wurde der Satz von Lagrange angewendet.

Wenn n keine Primzahl ist, dann bilden die Zahlen $\{1, 2, \ldots, n-1\}$ bezüglich der Multiplikation modulo n keine Gruppe, denn dann sind nicht alle dieser Zahlen Einheiten modulo n. Aber die Einheiten bilden eine Gruppe! Das Lemma von Fermat lässt sich auf beliebige Zahlen n verallgemeinern, wenn man es für Einheiten formuliert. Der Beweis geht dann wie im Beweis vom kleinen Fermat.

Lemma 50 (Lemma von Euler-Fermat)**.** *Ist* a *zu* n *teilerfremd, dann gilt*

$$a^{\phi(n)} \equiv 1 \pmod{n}\,.$$

6.7 Kryptographie mit öffentlichen Schlüsseln

Man kann das Lemma von Euler-Fermat benutzen, um Verschlüsselungsverfahren mit öffentlichem Schlüssel zu entwerfen. Das bekannteste ist das von Rivest[35], Shamir[36] und Adleman[37] (RSA). Dabei teilt derjenige, der eine verschlüsselte Nachricht empfangen möchte (im Folgenden der *Empfänger* genannt), der Absenderin vorab öffentlich einen "Schlüssel" mit. Mit diesem Schlüssel kann man Nachrichten verschlüsseln, nicht aber entschlüsseln. Zum Entschlüsseln wird ein anderer Schlüssel benötigt, den der Emfänger geheim für sich behält.

Und hier sind die Details:

[35] Ronald Linn Rivest; geboren am 6. Mai 1947 in Schenectady, New York.
[36] Adi Shamir; geboren am 6. Juli 1952 in Tel-Aviv.
[37] Leonard Adleman; geboren am 31. Dezember 1945 in San Francisco.

1. (Schlüssel anlegen) Der Empfänger
   (a) wählt zwei große Primzahlen p, q, am besten mit einigen Tausend Stellen, zufällig, und zwar unabhängig voneinander zufällig, und berechnet $n := p \cdot q$;
   (b) wählt eine zu $\phi(n) = (p-1) \cdot (q-1)$ teilerfremde Zahl $d \in \mathbb{Z}_{\phi(n)}$;
   (c) berechnet das multiplikative Inverse i von d modulo $\phi(n)$. Denn nach dem Satz von Bézout existieren $i, h \in \mathbb{Z}$ mit $i \cdot d + h \cdot \phi(n) = \mathrm{ggT}(\phi(n), d) = 1$. Insbesondere gilt also $i \cdot d \equiv 1 \pmod{\phi(n)}$. Die Zahlen i und h können mit dem erweiterten euklidischen Algorithmus berechnet werden.
   (d) Der Empfänger teilt n und i öffentlich der Absenderin mit.
2. (Verschlüsseln) Die Absenderin möchte dem Empfänger eine Nachricht mitteilen. Wir nehmen im Folgenden an, dass die Nachricht eine Zahl $m \in \mathbb{Z}_n$ ist. Im allgemeinen Fall wird eine Nachricht auf geeignete Weise in eine Folge von solchen Zahlen zerlegt, die einzeln übermittelt werden.

   Die Absenderin berechnet $c := m^i \bmod n$ mit Hilfe der binären Exponentiation, und schickt c an den Empfänger.
3. (Entschlüsseln) Der Empfänger berechnet $c^d \bmod n$ mit Hilfe der Methode der binären Exponentiation.

Wir behaupten, dass das Ergebnis des Empfängers tatsächlich m ist. Zu zeigen ist also: $m \equiv c^d \pmod{n}$. Es gilt

$$c^d = (m^i)^d = m^{i \cdot d} = m^{1 - h \cdot \phi(n)} = m \cdot (m^{\phi(n)})^{-h}\,.$$

Falls m und n teilerfremd sind, dann gilt nach dem Lemma von Euler-Fermat dass $m^{\phi(n)} \equiv 1 \pmod{n}$. Also ist $c^d = m \cdot (m^{\phi(n)})^{-h} \equiv m \pmod{n}$.

Ein Beispiel. Angenommen, die Absenderin möchte dem Empfänger die geheime Nachricht "I LOVE YOU" übermitteln. Zur Illustration arbeiten wir mit zwei kleinen Primzahlen $p = 59$ und $q = 71$; diese sind für einen guten kryptographischen Schutz viel zu klein; stattdessen sollte man besser Primzahlen mit einigen Tausend Stellen wählen.

Setup. Die Person, die Nachrichten empfangen will, berechnet

$$n := p \cdot q = 59 \cdot 71 = 4189$$

und eine zu

$$\phi(n) = \phi(59) \cdot \phi(71) = 58 \cdot 70 = 4060$$

teilerfremde Zahl $d \in \mathbb{Z}_{\phi(n)}$. Tatsächlich sind die meisten ungeraden Zahlen teilerfremd zu $\phi(n)$; wir können also eine solche Zahl $d \in \mathbb{Z}_{\phi(n)}$ zufällig auswählen, und den ggT von d und

$\phi(n)$ berechnen (mit dem euklidischen Algorithmus geht das auch für großes n effizient). Falls $\mathrm{ggT}(d, \phi(n)) = 1$ so haben wir die gewünschte Zahl gefunden; falls nicht, wiederholen wir die Suche mit anderen zufälligen ungeraden Zahlen aus $\mathbb{Z}_{\phi(n)}$. So fahren wir fort, bis wir eine gewünschte Zahl gefunden haben. In unserem Fall zum Beispiel $d = 13$. Von d berechnen wir mit dem erweiterten euklidischen Algorithmus auch gleich das multiplikative Inverse $i \in \mathbb{Z}_{\phi(n)}$:

$$\begin{aligned}
\mathrm{Euklid}(13, 4060) &= \mathrm{Euklid}(4060 \bmod 13, 13) = \mathrm{Euklid}(4, 13) \\
&= \mathrm{Euklid}(13 \bmod 4, 4) = \mathrm{Euklid}(1, 4) = 1 \\
\text{E-Euklid}(1, 4) &= (1, 0) \\
\text{E-Euklid}(4, 13) &= (0 - 1\lfloor 13/4 \rfloor, 1) = (-3, 1) \\
\text{E-Euklid}(13, 4060) &= (1 + 3\lfloor 4060/13 \rfloor, -3) = (1 + 3 \cdot 312, -3) = (937, -3)
\end{aligned}$$

Wir machen die Probe: $937 \cdot 13 - 3 \cdot 4060 = 12181 - 12180 = 1$. Das multiplikative Inverse von 13 in $\mathbb{Z}_{4060}$ ist daher $i = 937$. Die Zahlen n und i werden vom Empfänger öffentlich bekannt gegeben, p, q, und d bleiben geheim.

Verschlüsseln. Verschlüsselt wird nun wie folgt. Zunächst kodieren wir unser Alphabet.

Symbol	Leerzeichen	A	B	C	D	E	F	G	H	I	...
Code	00	01	02	03	04	05	06	07	08	09	...

Die Nachricht verwandelt sich also in

Nachricht	I		L	O	V	E		Y	O	U
Kodierung	09	00	12	15	22	05	00	25	15	21

.

Durch Umgruppierung der Ziffern erhalten wir die Folge von Nachrichtenpaketen

$$090 \quad 012 \quad 152 \quad 205 \quad 002 \quad 515 \quad 210 \ .$$

Diese Zahlen sind allesamt kleiner als $n = 4189$. Um das erste Packet 090 zu verschlüsseln, berechnet die Absenderin

$$c := m^i = 90^{937} \mod 4189.$$

Hierfür verwenden wir den Algorithmus von Al-Kashi. Zuerst wird 937 geschrieben als Summe von Zweierpotenzen:

$$937 = 2^9 + 2^8 + 2^7 + 2^5 + 2^3 + 2^0,$$

wir erhalten also die Binärdarstellung

$$1110101001.$$

Mit der Methode des Quadrierens und Multiplizierens erhalten wir also

$$90^{937} = \underbrace{\underbrace{\underbrace{\underbrace{(\underbrace{90^2 \cdot 90}_{\equiv 114})^2 \cdot 90}_{\equiv 909})^{2\cdot 2} \cdot 90}_{\equiv 3716})^{2\cdot 2} \cdot 90}_{\equiv 3630})^{2\cdot 2\cdot 2} \cdot 90 \equiv 998 \mod 4189.$$

Die Absenderin verschickt also das verschlüsselte erste Packet 998.

Entschlüsseln. Um das erste Packet zu entschlüsseln, muss

$$c^d = 998^{13} \mod 4161$$

berechnet werden; wieder verwenden wir den Algorithmus von Al Kashi. Wir schreiben den Exponenten als $13 = 2^3 + 2^2 + 2^0$, in Binärdarstellung also 1101, und die Potenz als

$$998^{13} = (\underbrace{998^2 \cdot 998}_{\equiv 4182})^{2\cdot 2} \cdot 998 \equiv 90 \mod 4189.$$

Der Empfänger hat damit das korrekte erste Packet 90 entschlüsselt.

Übungen.

23. Verschlüsseln und Entschlüsseln Sie das nächste Packet 012 der Nachricht oben für das gegebene p, q, d, i.

Ein Spezialfall. Was ist mit dem Fall, dass m und n nicht teilerfremd sind? Den Satz von Fermat können wir ohne diese Annahme auf diese Weise nicht verwenden. Zunächst sollte man vorwegschicken, dass das ein wirklicher Extremfall ist: denn da $m < n$, und $n = p \cdot q$, bedeutet das, dass die Absenderin ein Vielfaches von einem der geheimen Primfaktoren von n geschickt hat! Viele Autoren verhindern das, indem das Protokoll so abgeändert wird, dass nur Nachrichten m verschickt werden, die n nicht teilen (und das ist durch Anhängen eines zusätzlichen Bits problemlos zu erreichen). Allerdings gilt die Behauptung oben auch ohne die Annahme, dass m und n teilerfremd sind. Um das zu verstehen, zeigen wir zunächst das folgende Lemma.

Lemma 51. *Es seien q_1 und q_2 teilerfremde Zahlen. Dann gilt $a \equiv b \pmod{q_1}$ und $a \equiv b \pmod{q_2}$ genau dann, wenn $a \equiv b \pmod{q_1 q_2}$.*

Beweis. Wenn $a \equiv b \pmod{q_1 q_2}$, dann sicher auch $a \equiv b \pmod{q_i}$ für $i = 1$ und $i = 2$. Umgekehrt nehmen wir an dass $a \equiv b \pmod{q_i}$ für $i = 1$ und $i = 2$. Also gibt es $t_1, t_2 \in \mathbb{Z}$ mit $a = b + t_i q_i$. Wenn wir die eine dieser Gleichungen von der andere abziehen, erhalten wir $t_1 q_1 = t_2 q_2$. Da q_1 und q_2 teilerfremd sind, muss dieser Wert ein Vielfaches sein von $q_1 q_2$. Also gilt $a \equiv b \pmod{q_1 q_2}$. $\square$

Zurück zu unserer Behauptung. Die Primzahlen p und q sind teilfremd, also genügt es nach dem eben bewiesenen Lemma zu zeigen, dass $m \equiv c^d \pmod{p}$ und $m \equiv c^d \pmod{q}$. Falls p ein Teiler ist von m, dann gilt $c^d = m^{1+h\phi(n)} \equiv 0 \equiv m \pmod{p}$. Ansonsten gilt

$$\begin{aligned}
c^d &= m^{1+h\phi(n)} && \text{(wie oben)}\\
&= m \cdot m^{(p-1)(q-1)h} && \text{(Proposition 36)}\\
&\equiv m \cdot (1)^{(q-1)h} \pmod{p} && \text{(kleiner Fermat; } p \text{ teilt nicht } m\text{)}\\
&\equiv m \pmod{p}
\end{aligned}$$

Analog zeigen wir, dass $c^d \equiv m \pmod{q}$, und mit Lemma 51 erhalten wir die Behauptung.

Sicherheit. Warum ist das Verfahren sicher? Der zentrale Punkt hier ist die Tatsache, dass kein effizientes Verfahren bekannt ist, aus c, n, und i die Zahl m zu berechnen. Wenn man die Primfaktoren von n berechnen könnte, so hätte man auch Zugang zum geheimen Schlüssel d, und damit zur Nachricht m: denn mit p und q kennt man auch $\phi(n)$ nach Proposition 36. Dann aber kann man zum öffentlichen i auch das inverse d berechnen. Also kann man mit einem schnellen Faktorisierungsalgorithmus RSA knacken. (Umgekehrt ist allerdings nicht bekannt, ob man mit einem Algorithmus, der RSA knacken kann, auch einen effizienten Faktorisierungsalgorithmus konstruieren kann.)

Wenn man Sicherheit von Systemen diskutiert, ist es wichtig, das Gesamtsystem zu betrachten. Wir illustrieren die Herausforderung, gute Begriffe von Sicherheit zu definieren, anhand eines Beispiels einer weiteren Angriffsform. Dabei rät der Angreifer die Nachricht der Absenderin (zum Beispiel 'Ja.' oder 'Nein.'), verschlüsselt diese mit dem öffentlichen Schlüssel, und vergleicht die verschlüsselte Nachricht mit der, die die Absenderin verschickt hat. Wenn die verschlüsselten Nachrichten gleich sind, so hat der Angreifer die Gewissheit, richtig geraten zu haben (im Beispiel, wo der Angreifer ein 'Ja' oder 'Nein' erwartet, ist das vielleicht schon alles, was er wissen wollte). In der Praxis kann man dem leicht begegnen, indem die Absenderin einige Zufallsbits an die Nachricht hängt.

Um wirklich sicher zu gehen, dass sich beim Schlüsselaustausch niemand für jemand anderen ausgibt, sollte man hierfür einen vertrauenswerten Dritten verwenden. Dafür gibt es spezielle Server, bei denen öffentliche Schlüssel hinterlegt sind. Wenn man eine sichere Verbindung zu A aufbauen will, so schlägt man den öffentlichen Schlüssel von A also besser über eine sichere Verbindung bei einem solchen Server des Vertrauens nach. Denn wenn man den Schlüssel direkt bei A erfragt, kann man sich eventuell nicht sicher sein, ob sich jemand für A ausgibt, und seinen eigenen öffentlichen Schlüssel ausgibt, um damit spätere verschlüsselte Nachrichten entschlüsseln zu können.

Signieren von Nachrichten. Das Verfahren aus dem letzten Abschnitt eignet sich auch dazu, Nachrichten zu *signieren*, damit der Empfänger Gewissheit über den Autor der Nachricht hat. Es werden die gleichen öffentlichen und privaten Schlüssel verwendet wie oben.

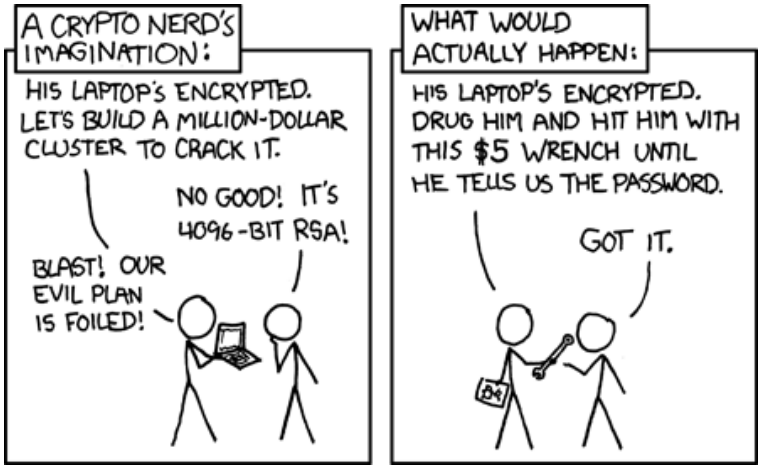


Abbildung 13: Heruntergeladen von http://xkcd.com/538/.

Sei d der private Schlüssel, und (n, i) der öffentliche Schlüssel der Absenderin. Das Vorgehen zum Signieren ist dann wie folgt:

- die Absenderin schickt neben der Nachricht m auch $s := m^d \pmod{n}$.
- Zum Prüfen berechnet der Empfänger $m' := s^i \bmod n$, und vergleicht das Ergebnis mit m. Wenn beide Zahlen übereinstimmen, ist die Signatur gültig.

Da $m' = s^i = m^{di} \equiv m \pmod{n}$ ist die Signatur ungültig, falls die beiden Zahlen m und m' nicht übereinstimmen.

Das Signieren lässt sich auch mit dem Verschlüsseln kombinieren: dazu wird zuerst signiert, und dann werden sowohl m als auch s verschlüsselt und versandt.

7 Graphen

Graphen sind in der Informatik und Mathematik allgegenwärtig. Das WWW mit seinen Webseiten und Verweisen zwischen Webseiten zum Beispiel kann man auf natürliche Weise als einen gerichteten Graphen betrachten. In der Mathematik lässt sich häufig der kombinatorische Kern eines Sachverhalts elegant mit Graphen formulieren. Viele kombinatorische Beweisprinzipien lassen sich sehr gut mit Aussagen für Graphen illustrieren.

Es gibt *gerichtete* und *ungerichtete* Graphen. Wir beginnen hier mit den ungerichteten. Die gerichteten Graphen folgen in Kapitel 12. In manchen Zusammenhängen ergibt es Sinn, sogenannte *Mehrfachkanten* zuzulassen; dann aber werden wir es extra dazu sagen.

Für eine Menge M schreiben wir $\binom{M}{2}$ für die Menge aller zwei-elementigen Teilmengen von M. Es gilt $|\binom{M}{2}| = \binom{|M|}{2}$. Im Übrigen folgen wir meist der Notation im Lehrbuch "Graphentheorie" von Reinhard Diestel [5]. Ein ausgezeichnetes Buch, das allerdings weit über den Stoff dieser Vorlesung hinausgeht.

Definition 52. *Ein (*schlichter[38]*,* ungerichteter*)* Graph G *ist ein Paar* (V, E) *bestehend aus einer* Knotenmenge V *und einer* Kantenmenge $E \subseteq \binom{V}{2}$*. Die Knotenmenge von* G *wird auch mit* $V(G)$*, und die Kantenmenge mit* $E(G)$ *bezeichnet.*

Die Elemente von V heißen Knoten (bisweilen in der deutschen Literatur auch *Ecken*; ein Knoten heißt auf englisch *vertex*) und die Elemente von E die *Kanten* des Graphen (eine Kante heißt auf englisch *edge*). Wir behandeln nun zwei Beispiele von ungerichteten Graphen.

- Knoten: Facebookprofile. Kanten: Zwei Profile sind mit einer Kante verbunden, falls die entsprechenden Profile befreundet sind.
- Knoten: Schauspieler. Kanten: Zwei Schauspieler sind mit einer Kante verbunden, falls sie zusammen in einem Film aufgetreten sind.

Wenn $\{u, v\} \in E(G)$, dann sagt man auch, dass u und v *adjazent* sind, und dass v ein *Nachbar* ist von u. Die Anzahl der Nachbarn von x in G ist der *Grad* von x. Ein Graph heisst *endlich*, wenn er endlich viele Knoten (und damit auch nur endlich viele Kanten) besitzt, und sonst heisst er *unendlich*. Die meisten Graphen, die wir in diesem Kurs betrachten, sind endlich; in vielen Situationen treten aber auch unendliche Graphen auf, und auch für unendliche Graphen lassen sich viele interessante Aussagen beweisen.

Wir zeigen einige Beispiele von Graphen. Sei $n > 0$ eine natürliche Zahl.

- K_n steht für den Graphen (V, E) mit $V := \{1, 2, \ldots, n\}$ und $E := \binom{V}{2}$, und wird die n-elementige *Clique* genannt.

[38] In diesem Abschnitt haben die Graphen auch keine *Schlingen*; Graphen ohne Mehrfachkanten und Schlingen werden *schlicht* genannt.

- I_n bezeichnet den Graphen $(\{1, 2, \ldots, n\}, \emptyset)$, und wird *stabile Menge* der Größe n genannt.

- P_n, für $n \geq 2$, bezeichnet den *Pfad der Länge* n, das heißt, den Graphen (V, E) mit $V := \{1, \ldots, n\}$ und $E := \{\{1, 2\}, \{2, 3\}, \ldots, \{n-1, n\}\}$. Achtung: in manchen Büchern und Artikeln ist P_n der Pfad mit n Kanten, und nicht, wie hier, der Pfad mit n Knoten. Das macht natürlich einen Unterschied, aber keinen großen.

- C_n, für $n \geq 3$, bezeichnet den Graphen

$$\big(\{0, 1, 2, \ldots, n-1\}, \big\{\{i, j\} \mid (i - j) \equiv 1 \pmod{n}\big\}\big),$$

  genannt *Kreis* (mit n Knoten und n Kanten).

Das *Komplement* eines Graphen $G = (V, E)$ ist der Graph $\overline{G} = (V, \binom{V}{2} \setminus E)$. Zum Beispiel ist I_n das Komplement von K_n. Selbstverständlich gilt $\overline{(\overline{G})} = G$.

Definition 53 (Isomorphie)**.** *Zwei Graphen G und H sind* isomorph*, wenn es eine Bijektion $f\colon V(G) \to V(H)$ gibt, so dass $\{u, v\} \in E(G)$ genau dann, wenn $\{f(u), f(v)\} \in E(H)$; intuitiv bedeutet das, dass man H aus G durch Umbenennen der Knoten von G erhält. Wir schreiben in diesem Fall $G \cong H$, und nennen f einen* Isomorphismus *von G nach H.*

Zum Beispiel ist das Komplement von C_5 isomorph zu C_5.

Definition 54 (Subgraph)**.** *Ein Graph H ist ein* Subgraph von G *falls gilt $V(H) \subseteq V(G)$ und $E(H) \subseteq E(G) \cap \binom{V(H)}{2}$. Ein* induzierter Subgraph von G *ist ein Graph H mit $V(H) \subseteq V(G)$, und $E(H) = E(G) \cap \binom{V(H)}{2}$.*

Die induzierten Subgraphen von G werden eindeutig durch ihre Knotenmenge bestimmt. Für $V \subset V(G)$ ist der *durch V induzierte Subgraph von G* der (eindeutige) induzierte Subgraph von G mit Knotenmenge V; wir schreiben auch $G[V]$ für diesen Graphen.

Eine Folge $(u_1, u_2, \ldots, u_l)$ von Knoten eines Graphen G heißt *Kantenzug* (oder *Streckenzug*) von u_1 nach u_l in G falls $\{u_i, u_{i+1}\} \in E(G)$ für alle $i \in \{1, \ldots, l-1\}$. Wir lassen hier explizit den Fall $l = 1$ zu; in diesem Fall hat der Kantenzug nur einen Knoten und keine Kanten. Ein Kantenzug $(u_1, u_2, \ldots, u_l)$ ist *geschlossen* falls $u_1 = u_l$, und ansonsten *offen*. Ein Kantenzug $(u_1, u_2, \ldots, u_l)$ ist ein *Weg* (oder *Pfad*) von u_1 nach u_l falls für verschiedene i, j aus $\{1, \ldots, l\}$ gilt, dass $u_i \neq u_j$. Wenn es einen Kantenzug von u nach v gibt, dann gibt es klarerweise auch einen Weg von u nach v (einfach die Umwege weglassen).

Ein *Kreis* ist ein Kantenzug $(u_0, u_1, \ldots, u_{l-1}, u_l)$ mit $u_0 = u_l$, $l \geq 3$ und $u_i \neq u_j$ für alle unterschiedlichen i, j aus $\{1, \ldots, l-1\}$. Mit anderen Worten: ein Graph enthält einen Kreis genau dann, wenn er einen Subgraphen enthält der isomorph ist zu C_n für ein $n \geq 3$.

7.1 Knotenzusammenhang

Sei G ein Graph.

Definition 55 (Zusammenhang)**.** *Ein Graph G heißt* zusammenhängend *falls es für alle $s, t \in V(G)$ einen Kantenzug von s nach t in G gibt.*

Es seien $G = (U, E)$ und $H = (V, F)$ zwei Graphen mit disjunkten Knotenmengen. Dann schreiben wir $G \uplus H$ für den Graphen $(U \cup V, E \cup F)$, die *disjunkte Vereinigung* von G und H.

Lemma 56. *Ein Graph G ist genau dann zusammenhängend, wenn er nicht geschrieben werden kann als $H_1 \uplus H_2$ für Graphen H_1, H_2 mit mindestens einem Knoten.*

Beweis. '$\Leftarrow$': Wir zeigen die Kontraposition. Es seien H_1 und $H + 2$ zwei Graphen mit $V(H_1) \cap V(H_2) = \emptyset$, $x \in V(H_1)$, und $y \in V(H_2)$. Man zeigt leicht mit vollständiger Induktion, dass $t \in V(H_1)$ falls es einen Kantenzug von x nach t in $H_1 \uplus H_2$ gibt, da es in $V(H_1 \uplus H_2)$ keine Kanten $\{u, v\}$ gibt mit $u \in V(H_1)$ und $v \in V(H_2)$. Da $y \notin V(H_1)$ gibt es also keinen Kantenzug von x nach y in $H_1 \uplus H_2$, und $H_1 \uplus H_2$ ist nicht zusammenhängend.

'$\Rightarrow$': Wieder zeigen wir die Kontraposition. Angenommen, es gibt $x, y \in V(G)$ ohne Kantenzug von x nach y in G. Es sei $V' \subseteq V(G)$ die Menge aller Knoten $v \in V(G)$ mit einem Kantenzug von x nach v. Dann gibt es in G keine Kanten zwischen Knoten aus V' und Knoten in $V(G) \setminus V'$. Also gilt $G = G[V'] \uplus G[V(G) \setminus V']$ und da $x \in V'$ und $y \in V(G) \setminus V'$ haben wir G als disjunkte Vereinigung von Graphen mit mindestens einem Element geschrieben. $\square$

Eine *Zusammenhangskomponente* ist ein maximaler zusammenhängender induzierter Subgraph H von G; *maximal* bedeutet hier, dass jeder induzierte Subgraph H' von G mit $V(H) \subsetneq V(H')$ nicht mehr zusammenhängend ist. Klarerweise haben zwei verschiedene Zusammenhangskomponenten von G disjunkte Knotenmengen, und jeder Knoten von G liegt in einer Zusammenhangskomponente. Also lässt sich jeder Graph schreiben als disjunkte Vereinigung seiner Zusammenhangskomponenten.

7.2 Färbbarkeit

Ein Graph G heißt *k-färbbar* (oder *k-partit*) falls es eine Funktion

$$f \colon V(G) \to \{0, 1, \ldots, k-1\}$$

gibt, so dass für alle $\{u, v\} \in E(G)$ gilt, dass $f(u) \neq f(v)$. In anderen Worten: falls wir die Knoten von G so mit k Farben einfärben können, dass benachbarte Knoten unterschiedliche Farben bekommen. Im folgenden sprechen wir daher von f als einer *Färbung* (mit den *Farben* $0, 1, \ldots, k-1$). Wann ist ein Graph zweifärbbar? Darauf gibt es eine elegante Antwort.

Proposition 57. *Ein endlicher Graph G ist genau dann zweifärbbar, wenn er keine* ungeraden Kreise *enthält.*

Beweis. Ungerade Kreise sind sicherlich nicht zweifärbbar; also brauchen wir nur eine Richtung der Äquivalenz zu zeigen. Sei also $G = (V, E)$ ein Graph ohne ungerade Kreise. Sicherlich ist ein Graph genau dann zweifärbbar wenn alle seine Zusammenhangskomponenten zweifärbbar sind. Wir färben eine Zusammenhangskomponente C von G wie folgt.

1. Zunächst wählen wir einen beliebigen Knoten u aus C, und definieren $f(u) := 0$.
2. Für alle Nachbarn v von u definieren wir $f(v) := 1$. Das ist möglich, da keine zwei verbunden sein können; sonst gäbe es einen Kreis der Länge 3.
3. Wenn wir bereits alle Knoten von C gefärbt haben, sind wir fertig mit dem Beweis.
4. Ansonsten erweitern wir für einen noch nicht gefärbten Nachbarn w eines bereits mit Farbe i gefärbten Knoten w' die Färbung mit $f(w) := 1 - i$. Alle bereits gefärbten Nachbarn von w tragen die Farbe i, denn ansonsten hätten wir einen ungeraden Kreis gefunden!
5. Fahre fort mit Schritt 3.

Da C endlich ist, bricht dieses Verfahren nach endlich vielen Schritten ab, und wir haben die gewünschte Färbung gefunden. □

Aus dem Beweis wird ersichtlich, dass es einen effizienten Algorithmus gibt, der für einen gegebenen Graphen feststellt, ob er zweifärbbar ist.

Zweifärbbare Graphen heißen auch *bipartit*. Mit anderen Worten, ein Graph G ist bipartit, wenn sich seine Knoten so in zwei disjunkte Teilmengen A und B aufteilen lassen, dass zwischen den Knoten innerhalb beider Teilmengen keine Kanten verlaufen. Die Menge $\{A, B\}$ heißt *Bipartition* von G, und A und B heißen *Partitionsklassen* (oder auch *Farbklassen*). In anderen Worten, $\{A, B\}$ ist eine Partition von V, und $E \cap \binom{A}{2} = E \cap \binom{B}{2} = \emptyset$. Wir zeigen nun einige Beispiele für bipartite Graphen G.

Beispiel 27. *Sei* ST *die Menge der Student:innen der TU Dresden, und* VL *sei die Menge der Vorlesungen im Wintersemester 2021/22. Sei E die Menge aller Mengen der Gestalt $\{s, v\}$ mit $s \in \mathrm{ST}$ und $v \in \mathrm{VL}$, so dass s die Vorlesung v belegt. Dann ist* $(\mathrm{ST} \cup \mathrm{VL}, E)$ *ein bipartiter Graph mit den Partitionsklassen* ST *und* VL.

Beispiel 28. *$K_{n,m}$ bezeichnet den* maximalen bipartiten Graphen *mit Partitionsklassen $P_1 := \{1, \ldots, n\}$ und $P_2 := \{n + 1, \ldots, m + n\}$, d.h.,*

$$K_{n,m} = (P_1 \cup P_2, \big\{\{u, v\} \mid u \in P_1, v \in P_2\big\}).$$

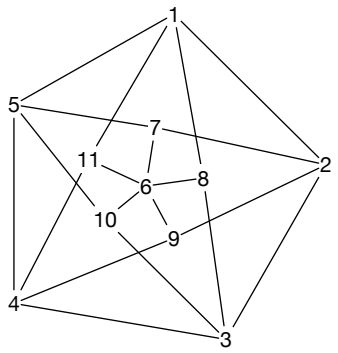


Abbildung 14: Der Grötzsch Graph (siehe Übung 25).

Wann ist ein Graph 3-färbbar? Dafür gibt es im allgemeinen keine so elegante Beschreibung wie in Proposition 57. Es ist auch kein effizienter Algorithmus bekannt, der für einen gegebenen Graphen testet, ob er 3-färbbar ist. Man kann zeigen, dass wenn es einen effizienten Algorithmus zum Testen von 3-Färbbarkeit gäbe, es dann auch einen effizienten Algorithmus für das aussagenlogische Erfüllbarkeitsproblem gäbe, mit all den Konsequenzen, die wir in Abschnitt 3.3 diskutiert haben.

Übungen.

24. Zeigen Sie: der Grötzsch Graph (siehe Abbildung 14) ist nicht 3-färbbar, aber 4-färbbar.

25. Gilt Proposition 57 auch für unendliche Graphen?

7.3 Bäume

Gleich zu Beginn eine Warnung: in der Informatik und Mathematik werden ganz verschiedene Dinge als Bäume bezeichnet. In der Graphentheorie jedoch ist die unumstößliche Definition von Bäumen die folgende.

Definition 58 (Baum)**.** *Ein* Baum *ist ein zusammenhängender Graph ohne Kreise.*

Nicht notwendigerweise zusammenhängende Graphen ohne Kreise haben auch einen Namen: aus naheliegendem Grund heißen solche Graphen *Wälder*. Nach dem, was wir im Abschnitt 7.1 gelernt haben, sind Wälder disjunkte Vereinigungen von Bäumen. Proposition 57 impliziert, dass Bäume und Wälder zweifärbbar sind.

Ein Knoten in einem Baum vom Grad eins heißt *Blatt*.

Lemma 59. *Jeder endliche Baum mit mindestens zwei Knoten enthält ein Blatt.*

Beweis. Es sei (V, E) unser endlicher Baum. Wähle $x \in V$ beliebig. Da V noch weitere Knoten besitzt, und (V, E) zusammenhängend ist, muss x einen Nachbarn y haben. Wenn y ein Blatt ist, sind wir fertig. Ansonsten hat y einen Nachbarn ungleich x. Wieder sind wir fertig, wenn der Nachbar ein Blatt ist. Also hat der Nachbar wieder einen Nachbarn, und so weiter. Da V endlich ist, schließt sich auf diese Weise irgendwann ein Kreis, was nicht sein kann, oder wir geraten irgendwann in eine Sackgasse, und haben damit das gesuchte Blatt gefunden. □

Wir schreiben $G - x$ für den Graph, den man aus G durch Löschen des Knotens x erhält. Formal ist $G - x$ definiert als der von $V(G) \setminus \{x\}$ induzierte Subgraph von G.

Lemma 60. *Sei $G = (V, E)$ ein endlicher zusammenhängender Graph mit mindestens einem Knoten. Dann sind äquivalent:*

1. *G ist ein Baum.*
2. *$|E| = |V| - 1$.*
3. *$|E| \leq |V| - 1$.*

Beweis. Die Implikation von 1. nach 2. zeigen wir per vollständiger Induktion über die Knotenanzahl $n := |V|$ von G. Die Aussage ist klarerweise korrekt für $n = 1$, da in diesem Fall G keine Kanten besitzen kann. Sein nun G ein Baum mit mehr als einem Knoten. Dann enthält G nach Lemma 59 ein Blatt x. Für $G' := G - x$ gilt nach Induktionsvoraussetzung, dass $|E(G')| = |V(G')| - 1$. Da G genau einen Knoten und eine Kante mehr als G' hat, folgt die gewünschte Aussage.

Die Implikation von 2. nach 3. ist trivial. Wir zeigen die Implikation von 3. nach 1. durch Kontraposition: angekommen, (V, E) besitzt einen Kreis. Dann entfernen wir eine Kante von diesem Kreis aus $E(G)$. So fahren wir fort, bis der resultierende Graph H keine Kreise mehr enthält. Auch H ist zusammenhängend, da wir nur Kanten aus Kreisen entfernt haben. Also ist H ein Baum, und

$$\begin{aligned} |E| &> |E(H)| && \text{(nur Kanten entfernt)} \\ &= |V(H)| - 1 = |V| - 1 && \text{(wegen 1. } \Rightarrow \text{ 2., bereits gezeigt)} \end{aligned}$$ □

Lemma 61. *Es sei G ein endlicher Graph. Die folgenden Aussagen sind äquivalent.*

1. *G ist ein Baum.*
2. *G hat maximal viele Kanten ohne einen Kreis zu enthalten.*
3. *G hat minimal viele Kanten mit der Eigenschaft, zusammenhängend zu sein.*
4. *Zwischen zwei Knoten in einem Baum gibt es einen* eindeutigen *Weg.*

Beweis. Die Charakterisierung von Bäumen in 2. und 3. folgt leicht aus Lemma 60. Daher gleich zu 4.: Wenn es zwischen zwei Knoten x, y eines Graphen zwei verschiedene Wege $x = z_1, z_2, \ldots, z_n = y$ und $x = z'_1, z'_2, \ldots, z'_m = y$ gibt, und $i \leq n$ kleinstmöglich, so dass $z_i = z'_j$ für ein $j \leq m$, dann ist $(z_2, z_3, \ldots, z_i, z'_{j-1}, \ldots, z'_2, z'_1)$ ein Kreis in G. Umgekehrt gibt es zwischen zwei Knoten eines Kreises zwei verschiedene Wege, also ist 4. äquivalent dazu, dass G ein Baum ist. □

7.4 Zweifacher Zusammenhang

Sei G ein Graph, $x \in V(G)$, und H die Zusammenhangskomponente von x. Dann ist x ein *Gelenkpunkt* von G falls $H - x$ nicht zusammenhängend ist.

Definition 62 (Zweifacher Zusammenhang)**.** *Ein Graph G heißt* zweifach (Knoten-) zusammenhängend*, falls er zusammenhängend ist, mindestens drei Knoten hat, und keine Gelenkpunkte besitzt.*

Ein Graph ist also genau dann zweifach zusammenhängend, wenn er mindestens drei Knoten hat, und wenn $G - x$ für alle $x \in V(G)$ zusammenhängend ist.

Ein maximaler zusammenhängender Subgraph von $G = (V, E)$, der selbst keine Gelenkpunkte besitzt, heißt *Block*. Das bedeutet, jeder Block von G ist entweder

- im Fall $|V(B)| = 1$: ein isolierter Knoten, das heißt, ein Knoten in G vom Grad null;
- im Fall $|V(B)| = 2$: eine *Brücke* in G, das heißt, eine Subgraph von G mit zwei Knoten und genau einer Kante $\{u, v\}$ so dass $(V, E \setminus \{\{u, v\}\})$ nicht zusammenhängend ist; oder
- im Fall $|V(B)| \geq 3$: ein maximaler zweifach zusammenhängender Subgraph von G.

Anders als bei der Zerlegung eines Graphen in seine Zusammenhangskomponenten sind die Blöcke im allgemeinen nicht knotendisjunkt. Zwei Blöcke können jedoch höchstens einen gemeinsamen Knoten haben, und dieser Knoten ist dann ein Gelenkpunkt in G. Wie sich die Blöcke überlappen, kann durch einen Wald beschrieben werden.

Definition 63 (Der Blockgraph)**.** *Sei G ein Graph, sei $A \subset V(G)$ die Menge der Gelenkpunkte von G, und $\mathcal{B}$ die Menge der Blöcke von G. Dann ist der* Blockgraph *von G der Graph mit Knotenmenge $A \cup \mathcal{B}$ und genau den Kanten $\{a, B\}$ mit $a \in A$, $B \in \mathcal{B}$, und $a \in V(B)$.*

Der Beweis des folgenden Satzes ist nicht schwer.

Proposition 64. *Der Blockgraph eines Graphen ist ein Wald. Der Blockgraph eines zusammenhängenden Graphen ist ein Baum.*

Beweis. Wir nehmen an, dass der Blockgraph des Graphen G einen Kreis besitzt; dieser Kreis ist von der Gestalt $(U_1, v_1, U_2, v_2, \ldots, U_k)$ für $k \geq 2$, wobei

- $U_1, \ldots, U_k$ Blöcke von G sind mit $U_1 = U_k$, und
- $v_1, \ldots, v_k$ Gelenkpunkte von G sind.

Für $i \in \{1, \ldots, k-1\}$ gibt es in U_i per Definition einen Pfad von v_i nach v_{i+1}. Wenn wir alle diese Pfade zusammensetzen, erhalten wir einen Kreis in G, im Widerspruch dazu, dass die v_i Gelenkpunkte sind.

Sei nun G ein zusammenhängender Graph. Wir zeigen zunächst per Induktion über k, dass wenn es einen Pfad $(v_1, \ldots, v_k)$ der Länge k zwischen zwei Gelenkpunkten v_1, v_k in G gibt, dann auch einen Pfad im Blockgraphen. Diese Aussage ist sicherlich richtig für $k = 1$. Sei i größtmöglich, so dass alle Knoten $v_1, v_2, \ldots, v_i$ im gleichen Block B von G liegen. Wir bemerken, dass v_i wieder ein Gelenkpunkt von G sein muss. Nach Induktionsvoraussetzung existiert ein Pfad $\bar{w}$ von v_i nach v_k im Blockgraph. Der Pfad im Blockgraph von v_1 nach v_k ist dann $(v_1, B, \bar{w})$. Wir überlassen es Ihnen, den Beweis zu vervollständigen. □

Sei $G = (V, E)$ ein endlicher Graph. Der Blockgraph eines Graphen läßt sich in Zeit $a + b|V| + c|E|$ berechnen (mit einer Variation einer Technik, die wir in Abschnitt 12.2 kennenlernen werden), wobei a, b, c kleine Konstanten sind, die vom Maschinenmodell abhängen, und die nicht von Interesse sind. Das ist sehr schnell! Naive Algorithmen benötigen hier typischerweise mindestens quadratische Zeit in der Größe der Eingabe.

Ohrendekompositionen. Seien $G = (V, E)$ ein Graph, $n \in \mathbb{N}$, $a_0, a_1, \ldots, a_n, a_{n+1}$ paarweise verschieden mit $V(G) \cap \{a_0, a_1, \ldots, a_n, a_{n+1}\} = \{a_0, a_{n+1}\}$. Dann heißt der Graph mit Knotenmenge $V \cup \{a_1, \ldots, a_n\}$ und Kantenmenge

$$E \cup \big\{\{a_0, a_1\}, \{a_1, a_2\}, \{a_2, a_3\}, \ldots, \{a_n, a_{n+1}\}\big\}$$

der aus G durch Anhängen des Pfades $(a_0, a_1, \ldots, a_n, a_{n+1})$ entstandene Graph. Für $n = 0$ erhalten wir also den Graphen $(V, E \cup \{a_0, a_1\})$. Die angehängten Pfade nennt man dann die *Ohren*.

Proposition 65. *Ein Graph ist genau dann zweifach zusammenhängend, wenn er aus einem Kreis durch sukzessives Anhängen von Pfaden konstruiert werden kann.*

Beweis. Die Rückrichtung der Proposition zeigt man mittels vollständiger Induktion über die Anzahl der Ohren der Dekomposition: Kreise sind zweifach zusammenhängend, und wenn G zweifach zusammenhängend ist, und G' aus G durch Anhängen eines Pfades entstanden ist, dann ist auch G' zweifach zusammenhängend, wie man sich leicht überzeugt.

Umgekehrt sei $G = (V, E)$ zweifach zusammenhängend. Sei H ein Subgraph von G den man durch sukzessives Anhängen von Pfaden aus einem Kreis gewinnen kann und der maximal ist bezüglich $|V| + |E|$. Wegen der Maximalität von H ist H bereits ein induzierter Subgraph von G, denn einzelne Kanten können als Ohren mit nur zwei Knoten aufgenommen werden. Weiterhin enthält G und also auch H einen Kreis, daher $V(H) \geq 3$. Wenn also $H \neq G$ ist, dann gibt es eine Kante $\{u, v\} \in E$ mit $u \in V(H)$ und $v \in$

$V \setminus V(H)$, da G zusammenhängend ist. Da G zweifach zusammenhängend ist, enthält $G-u$ einen Pfad $(v, p_1, \ldots, p_n, w)$ von v zu einem Knoten $w \in V(H)$. Dann erhalten wir aus H durch Anhängen des Pfades $(u, v, p_1, \ldots, p_n, w)$ einen größeren Subgraphen von G als H, im Widerspruch zur Maximalität von H. □

Man kann Proposition 65 auch als die Aussage betrachten, dass sich beliebige zweifach zusammenhängende Graphen G dekomponieren lassen, indem man sukzessive Ohren abschneidet, und schließlich einen Kreis erhält. Dieser Kreis, zusammen mit der gesamten Folge an angehängten Pfaden nennt man daher auch die *Ohrendekomposition* (oder *Ohrenzerlegung*) von G.

7.5 Der Satz von Menger

Der Begriff des Zusammenhangs und zweifachen Zusammenhangs lässt sich unschwer auf k-fachen Zusammenhang verallgemeinern.

Sei $G = (V, E)$ ein Graph und $X \subseteq V$. Wir schreiben $G - X$ für den Graphen $G[V \setminus X]$. Sei $k \geq 1$. Dann heißt G *k-fach (Knoten-) zusammenhängend* wenn $|V| > k$ und wenn für alle $x_1, \ldots, x_{k-1} \in V$ der Graph $G - \{x_1, \ldots, x_{k-1}\}$ zusammenhängend ist.

Bemerkung 66. *Die Motivation für die Forderung, dass k-fach zusammenhängende Graphen mindestens $k+1$ Knoten haben müssen, ist, dass sonst k-zusammenhängende Graphen nicht unbedingt $k-1$ zusammenhängend wären. Zum Beispiel wäre dann der Graph I_2, der aus zwei isolierten Knoten besteht, 2-fach zusammenhängend – und das wäre unschön und unpraktisch. Achtung: ein Graph mit nur einem Knoten ist zwar zusammenhängend im Sinne von Definition 55, nicht aber 1-fach zusammenhängend. Für Graphen mit mindestens zwei Knoten jedoch sind Zusammenhang und 1-facher Zusammenhang äquivalent.*

Zwei Pfade $(u_1, \ldots, u_k)$ und $(v_1, \ldots, v_l)$ von $a := u_1 = v_1$ nach $b := u_k = v_l$ in G heißen *unabhängig* falls $\{u_1, \ldots, u_k\} \cap \{v_1, \ldots, v_l\} = \{a, b\}$.

Satz 67. *Sei $k \geq 1$. Ein endlicher Graph $G = (V, E)$ mit mehr als k Knoten ist genau dann k-fach zusammenhängend, wenn er für alle $a, b \in V$ mindestens k paarweise unabhängige Pfade von a nach b enthält.*

Um diese wichtige Charakterisierung von k-fachem Zusammenhang zu zeigen, beweisen wir eine stärkere Aussage, nämlich den Satz von Menger, einen der Eckpfeiler der Graphentheorie. Für $A, B \subseteq V$ ist ein *A-B Pfad* ein Pfad $(v_1, \ldots, v_n)$ mit $\{v_1, \ldots, v_n\} \cap A = \{v_1\}$ und $\{v_1, \ldots, v_n\} \cap B = \{v_n\}$. Zwei Pfade heißen *disjunkt* falls die Knotenmengen der Pfade disjunkt sind. Eine Menge $X \subseteq V$ *trennt* A von B in G falls jeder A-B Pfad einen Knoten aus X enthält.

Definition 68 (Kantenkontraktionen)**.** *Ist G ein Graph und $e = \{a, b\}$ eine Kante in G, dann bezeichne G/e den Graphen, der aus G folgendermaßen entsteht:*

1. *Die Knoten a und b werden entfernt und durch einen neuen Knoten $v_{\{a,b\}}$ ersetzt.*

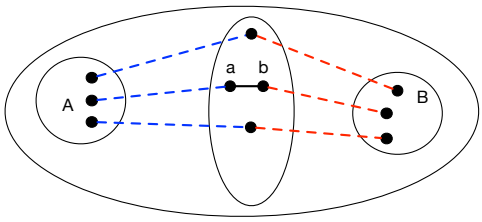
2. *Kanten, die weder a noch b enthalten, bleiben unverändert.*

3. *Für jede Kante, die in G den Knoten a oder den Knoten b mit einem Knoten $c \in V(G) \setminus \{a, b\}$ verbindet, enthält der neue Graph G/e eine Kante, die c mit dem neuen Knoten $v_{\{a,b\}}$ verbindet. Weitere Kanten gibt es nicht.*

Satz 69 (von Menger[39])**.** *Sei $G = (V, E)$ ein endlicher Graph, und $A, B \subseteq V$. Dann ist die maximale Anzahl von paarweise disjunkten A-B Pfaden in G gleich der Größe k der kleinsten Knotenmenge, die A von B in G trennt.*

Beweis. Klarerweise kann es nicht mehr als k disjunkte A-B Pfade in G geben. Es bleibt also zu zeigen, dass es k paarweise disjunkte A-B Pfade in G gibt, falls jede Menge, die A von B trennt, mindestens k Knoten hat. Wir zeigen dies per Induktion über $|E|$. Falls $E = \emptyset$ dann ist $k = |A \cap B|$ und es gibt k (einelementige) A-B Pfade.

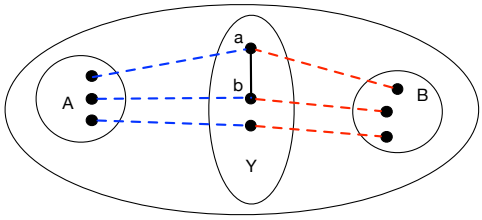
Sei also $\{a, b\} \in E$. Wir betrachten $G/\{a, b\}$ und zählen den neuen Knoten $v_{\{a,b\}}$ in $G/\{a, b\}$ als Element von A bzw. B wenn einer der beiden Knoten a, b ein Element von A bzw. B ist. Wenn es k disjunkte A-B Pfade in $G/\{a, b\}$ gibt, so auch in G, und wir sind fertig (siehe Abbildung).



Ansonsten gibt es nach Induktionsvoraussetzung in $G/\{a, b\}$ eine Knotenmenge X mit höchstens $k - 1$ Elementen, die A von B trennt. Jeder A-B Pfad in G ist entweder ein A-B-Pfad in G/e, oder wird nach Kontraktion von e zu einem A-B-Pfad in G/e, und muss also einen Knoten aus X enthalten. Also muss die Menge X den Knoten $v_{\{a,b\}}$ enthalten, denn ansonsten würde X auch A von B in G trennen, was der Wahl von k widerspricht. Die Menge $Y := (X \setminus \{v_{\{a,b\}}\}) \cup \{a, b\}$ trennt A und B in G. Nach Voraussetzung gilt also $|Y| \geq k$. Da $|X| \leq k - 1$ gilt $|Y| \leq k$. Also $|Y| = k$.

Wir betrachten nun den Graphen $G' := (V, E \setminus \{\{a, b\}\})$. Jede Menge S, die A von Y in G' trennt, trennt auch A von B in G und hat daher mindestens k Elemente. Die Induktionsannahme gibt uns nun k disjunkte A-Y Pfade in G'. Analog gibt es k disjunkte Y-B Pfade in G'.

[39] Karl Menger; geboren am 13. Januar 1902 in Wien; gestorben am 5. Oktober 1985 in Chicago.



Zusammengesetzt ergeben diese Pfade k disjunkte A-B Pfade in G (siehe zweite Abbildung). □

Korollar 70. *Sei $G = (V, E)$ ein endlicher Graph und $a, b \in V(G)$ so dass $\{a, b\} \notin E$. Dann ist die maximale Anzahl paarweise unabhängiger Pfade von a nach b gleich der Größe der kleinsten Teilmenge von $V \setminus \{a, b\}$, die $\{a\}$ von $\{b\}$ in G trennt.*

Beweis. Wenn sich $\{a\}$ und $\{b\}$ in G mit k Knoten aus $V \setminus \{a, b\}$ trennen lassen, dann kann es höchstens k unabhängigen Pfade von a nach b geben.

Angenommen, $\{a\}$ und $\{b\}$ lassen sich in G nicht mit k Knoten aus $V \setminus \{a, b\}$ trennen. Es sei $A \subseteq V$ die Menge der Nachbarn von a, und $B \subseteq V$ die Menge der Nachbarn von b in G. Da a und b nicht benachbart sind, lassen sich auch A und B nicht mit k Knoten trennen. Also gibt es nach dem Satz von Menger k disjunkte A-B Pfade. Zusammen mit Startpunkt a und Endpunkt b erhalten wir k unabhängige Pfade von a nach b in G. □

Beweis von Satz 67. Es seien $a, b \in V$ beliebige Knoten des Graphen G. Wenn es k unabhängige Pfade von a nach b in G gibt, dann gibt es auch nach Entfernen von $k - 1$ von a und b verschiedenen Knoten von G einen Pfad von a nach b.

Umgekehrt nehmen wir k-fachen Zusammenhang von G an. Im Fall, dass a und b nicht benachbart sind, folgt die Aussage aus Korollar 70. Ansonsten betrachten wir $G' := (V, E \setminus \{\{a, b\}\})$. Wenn es in G' bereits $k - 1$ unabhängige Pfade gibt, dann gibt es in G sogar k unabhängige Pfade. Also betrachten wir im folgenden die Situation, dass dem nicht so ist. Wieder nach Korollar 70 schließen wir, dass wir $\{a\}$ von $\{b\}$ in G' mit einer Knotenmenge $X \subseteq V \setminus \{a, b\}$ der Größe $k-2$ trennen können. Da $|V| > k$, gibt es ein $v \in V \setminus (X \cup \{a, b\})$. Dann trennt X entweder $\{v\}$ von $\{a\}$ oder $\{v\}$ von $\{b\}$ in G'. Nehmen wir ersteres an, der andere Fall geht analog. Dann gibt es in $G - (X \cup \{b\})$ keinen Pfad von a nach v, im Widerspruch zur Annahme, dass G k-fach zusammenhängend ist. □

Bemerkung 71. *Analog zur Blockdekomposition eines Graphen kann man einen beliebigen 2-fach zusammenhängenden Graphen mit Hilfe von 3-fach zusammenhängenden Graphen beschreiben, und wieder ist die Dekomposition 'baumartig'; diese Beoachtung stammt von Trakhtenbrot*[40].

[40] Boris (Boaz) Avraamovich Trakhtenbrot; geboren am 19. Februar 1921 in Tîrnova, Rajon Donduşeni; gestorben am 19. September 2016.

7.6 Kantenzusammenhang und Kantengraphen

Der *Kantengraph* (oder englisch *line graph*) $L(G)$ eines Graphen $G = (V, E)$ ist der Graph (V', E') wobei $V' := E(G)$ und $E' := \big\{\{\{u_1, v_1\}, \{u_2, v_2\}\} \mid |\{u_1, v_1\} \cap \{u_2, v_2\}| = 1\big\}$. Die Kanten von G werden also zu den Knoten des Kantengraphen, und zwei verschiedene Kanten von G werden benachbart im Kantengraph, wenn sie sich einen Knoten teilen; siehe Abbildung 15. Eine interessante Übung zu Kantengraphen ist Übung 30.

Kantengraphen haben unzählige Anwendungen in der Graphentheorie. Wir können mit ihrer Hilfe zum Beispiel ganz leicht eine Variante von Menger's Satz für Kanten anstatt Knoten zeigen. Hierfür zunächst eine Definition.

Definition 72. *Ein Graph $G = (V, E)$ mit mehr als einem Knoten heißt* k-fach kantenzusammenhängend*, falls für jede Kantenmenge $F \subseteq E$ mit $|F| < k$ der Graph $(V(G), E \setminus F)$ zusammenhängend ist.*

Eine Menge an Kanten $F \subseteq E$ *trennt* die Knoten a und b, falls es in $(V(G); E \setminus F)$ keinen Kantenzug von a nach b gibt. Zwei Pfade $(u_1, \ldots, u_k)$ und $(v_1, \ldots, v_l)$ heissen *kantendisjunkt* falls $\big\{\{u_1, u_2\}, \ldots, \{u_{k-1}, u_k\}\big\} \cap \big\{\{v_1, v_2\}, \ldots, \{v_{l-1}, v_l\}\big\} = \emptyset$.

Proposition 73. *Es seien a und b zwei verschiedene Knoten eines endlichen Graphen G. Dann ist die maximale Anzahl von kantendisjunkten Pfaden von a nach b in G gleich der Größe der kleinsten Kantenmenge, die a von b in G trennt.*

Beweis. Wenn es k kantendisjunkte Pfade von a nach b in G gibt, so kann man a von b sicher nicht durch Entfernen von weniger als k Kanten trennen. Für die umgekehrte Richtung wenden wir den Satz von Menger (Satz 69) auf $L(G)$ an mit

$$A := \{u \in E(G) \mid a \in u\} \subseteq V(L(G)) \quad \text{und} \quad B := \{u \in E(G) \mid b \in u\} \subseteq V(L(G))$$

(Siehe Abbildung 15.) Wenn wir mindestens k Kanten in G entfernen müssen, um a von b zu trennen, dann müssen wir mindestens k Knoten in $L(G)$ entfernen, um A von B zu trennen. Also gibt es nach dem Satz von Menger mindestens k paarweise disjunkte A-B Pfade in $L(G)$. Diese Pfade entsprechen kantendisjunkten Pfaden von a nach b in G. □

Korollar 74 (Globale Version)**.** *Ein endlicher Graph ist genau dann k-fach kantenzusammenhängend, wenn er zwischen je zwei Knoten mindestens k kantendisjunkte Pfade besitzt.*

Beweis. Wenn es zwischen zwei Knoten a und b mindestens k kantendisjunkte Pfade gibt, so gibt es auch nach dem Entfernen von $k - 1$ Kanten einen Pfad von a nach b. Wenn G also zwischen je zwei Knoten mindestens k kantendisjunkte Pfade besitzt, so ist G k-fach kantenzusammenhängend. Umgekehrt sei G k-fach kantenzusammenhängend und $a, b \in V(G)$. Dann muss man mindestens k Kanten entfernen, um a von b in G zu trennen. Nach Proposition 73 gibt es also mindestens k kantendisjunkte Pfade von a nach b in G. □

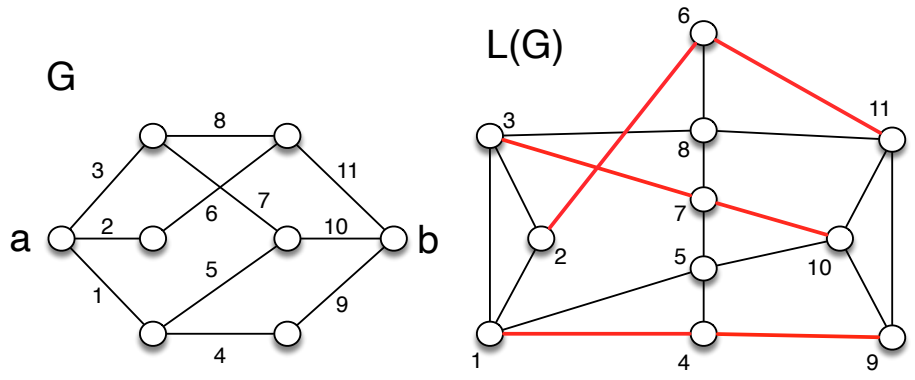


Abbildung 15: Ein Graph und sein Kantengraph.

Übungen.

26. Geben Sie einen alternativen Beweis von Proposition 67 an für den Spezialfall $k = 2$ mit Hilfe einer vollständigen Induktion über die Anzahl der Ohren einer Ohrendekomposition.

27. Zeigen Sie: wenn ein Graph k-fach (knoten-) zusammenhängend ist, dann auch k-fach kantenzusammenhängend.

7.7 Eulersche Graphen

Sei G ein Graph. Ein geschlossener Kantenzug, in dem jede Kante von G genau einmal auftaucht, heißt *Eulerzug* (auch *Eulertour* oder *Eulersche Linie*). Ein solcher Graph heißt *eulersch*. Ein *offener Eulerzug* ist ein offener Kantenzug in G, der alle Kanten von G genau einmal durchläuft.

Welche Graphen besitzen einen Eulerzug? Diese Frage wurde als das *Königsberger Brückenproblem* bekannt und 1736 von Leonhard Euler beantwortet. Der Beweis unten stammt von Hierholzer[41] von 1873.

Satz 75. *Sei G ein zusammenhängender endlicher Graph. Dann besitzt G genau dann einen Eulerzug, wenn jeder seiner Knoten geraden Grad hat.*

Beweis. Klarerweise müssen in einem Graph mit einem Eulerzug alle Knoten geraden Grad haben: wir verlassen ja einen Knoten im Eulerzug über genausoviel Kanten, wie wir ihn betreten.

[41] Carl Hierholzer; geboren am 2. Oktober 1840 in Freiburg im Breisgau; gestorben am 13. September 1871 in Karlsruhe.

Umgekehrt nehmen wir an, dass alle Knoten von G geraden Grad haben. Es sei $Z = (u_0, u_1, \ldots, u_l)$ der längstmögliche Kantenzug in G, der keine Kante zweimal verwendet. Da Z nach Annahme nicht verlängert werden kann, sind alle Kanten an u_l bereits von Z durchlaufen. Da es aber eine gerade Anzahl solcher Kanten gibt, muss $u_l = u_0$ gelten. Angenommen, Z ist kein Eulerzug. Dann hat G eine Kante $\{v, w\}$, die nicht vom Eulerzug durchlaufen wird. Da G zusammenhängend ist, können wir v, w so wählen, dass $w = u_i$ für ein $i \in \{0, \ldots, l\}$. Dann ist $(v, u_i, \ldots, u_{l-1}, u_l, u_1, \ldots, u_i)$ ein Kantenzug in G, der länger ist als Z, ein Widerspruch. □

Satz 76. *Ein zusammenhängender endlicher Graph $G = (V, E)$ besitzt genau dann einen offenen Eulerzug, wenn er genau zwei Knoten ungeraden Grades hat.*

Beweis. Wenn G einen offenen Eulerzug $(u_0, \ldots, u_l)$ besitzt, dann haben genau u_0 und u_l einen ungeraden Grad.

Umgekehrt sei G so, dass nur die Knoten u und v ungeraden Grad haben. Sei $w \notin V$ beliebig. Betrachte den Graphen $G' := \big(V \cup \{w\}, E \cup \{\{u, w\}, \{w, v\}\}\big)$. In G' haben *alle* Knoten geraden Grad. Da G' zusammenhängend ist, hat G' einen Eulerzug nach Satz 75. Sicherlich können wir diesen Eulerzug so wählen, dass er in w beginnt. Streicht man nun in diesem Eulerzug den Knoten w, so erhält man einen offenen Eulerzug für G. □

Übung.

28. Warum kann man im Beweis von Satz 76 anstatt des Graphen G' nicht einfach den Graphen $G'' := (V, E \cup \{u, v\})$ verwenden?

Wie findet man einen Eulerzug? Einfach losmarschieren führt nicht sicher zum Ziel. Aber es ist nicht schwer, aus dem Existenzbeweis für einen Eulerzug in Satz 75 auch einen effizienten Algorithmus zu gewinnen, der diesen Zug berechnet. Mit geeigneten Datenstrukturen kann man sogar eine Rechenzeit erreichen, die linear in der Größe der Eingabe ist. Wenn wir das Problem ändern, und fordern, dass der Kantenzug jeden *Knoten* genau einmal besuchen soll (man spricht dann von einem *Hamiltonkreis*), erhalten wir ein sehr viel schwereres Berechnungsproblem. Von diesem Problem vermutet man, dass es von keinem Algorithmus mit polynomieller Rechenzeit gelöst werden kann. Denn wenn es einen solchen Algorithmus gäbe, dann auch einen für das aussagenlogische Erfüllbarkeitsproblem, mit den bereits diskutierten Konsequenzen.

7.8 Paarungen

Sei G ein Graph. Eine *Paarung*, auch oft englisch *Matching*, ist eine Teilmenge M von $E(G)$ paarweise disjunkter Kanten. Eine *perfekte Paarung* ist eine Paarung M mit $2|M| = |V|$. Falls $\{x, y\} \in M$, so nennen wir y den *Partner* von x (und x den Partner von y). Für $S \subseteq V(G)$ ist M eine *Paarung von S* falls jedes Element von S in einer Kante aus M auftaucht.

Wenn A, B Mengen sind, dann ist die *symmetrische Differenz* von A und B die Menge

$$A \Delta B \ := \ \{x \in A \cup B \mid x \notin A \cap B\} \ .$$

Lemma 77. *Es seien M_1 und M_2 Paarungen in $G = (V, E)$. Dann besteht der Graph $(V, M_1 \Delta M_2)$ aus einer disjunkten Vereinigung von Kreisen gerader Länge und Pfaden.*

Beweis. Idee: jeder Knoten $v \in V$ hat höchstens zwei Nachbarn in $(V, M_1 \Delta M_2)$. □

Wie können wir in G eine Paarung maximaler Größe finden? Betrachten wir dazu eine beliebige Paarung M. Ein Pfad in G, der abwechselnd über Kanten aus $E \setminus M$ und Kanten aus M verläuft, heißt *alternierender Pfad*. Ein alternierender Pfad P heißt *augmentierend* bezüglich M falls sowohl der Startpunkt als auch der Endpunkt von P keinen Partner haben. Augmentierende Pfade können wir verwenden, um eine größere Paarung als M zu finden: Sei dazu M' die symmetrische Differenz von M und P. Falls P ein augmentierender Pfad ist, so ist M' wieder eine Paarung und $|M'| > |M|$.

Lemma 78 (Lemma von Berge[42])**.** *Sei G ein endlicher Graph. Eine Paarung M in G ist genau dann größtmöglich, wenn es keine augmentierenden Pfade bezüglich M in G gibt.*

Beweis. Wir haben bereits gesehen, dass eine Paarung mit einem augmentierenden Pfad nicht größtmöglich sein kann. Für die andere Richtung der Aussage nehmen wir nun an, dass es eine Paarung M' in G gibt mit $|M'| > |M|$. Sei D die symmetrische Differenz von M und M'. Da $|M'| > |M|$ hat der Graph (V, D) eine Zusammenhangskomponente mit mehr Kanten aus M' als aus M. Nach Lemma 77 muss eine solche Komponente ein augmentierender Pfad bezüglich M sein. □

Wenn $S \subseteq V$, so schreiben wir $N(S)$ für

$$\{n \in V \mid \text{ es gibt } s \in S \text{ so dass } \{n, s\} \in E\},$$

die *Nachbarschaft* von S in G. Eine klarerweise notwendige Bedingung für die Existenz einer Paarung von $S \subseteq V$ in G ist $|N(S)| \geq |S|$. Diese Bedingung ist im allgemeinen natürlich nicht hinreichend.

Im Folgenden sei $G = (V, E)$ ein bipartiter Graph mit fester Bipartition A, B (siehe Abschnitt 7.2).

Satz 79 (Heiratssatz von Hall[43])**.** *Ein bipartiter Graph G erlaubt genau dann eine Paarung von A, wenn $|N(S)| \geq |S|$ für alle $S \subseteq A$.*

[42] Claude Berge; geboren am 5. Juni 1926; gestorben am 30. Juni 2002.

[43] Philip Hall; geboren am 11. April 1904 in Hampstead, London; gestorben am 30. Dezember 1982 in Cambridge.

Beweis. Nach obiger Bemerkung müssen wir nur die Rückrichtung zeigen. Es sei M eine Paarung von G die einen Knoten a_0 aus A ohne Partner lässt. Wir werden einen augmentierenden Pfad bezüglich M konstruieren. Es sei $a_0, b_1, a_1, b_2, a_2, \ldots$ eine maximale Folge unterschiedlicher Knoten $a_i \in A$ und $b_i \in B$, so dass

1. $\{b_i, a_i\} \in M$, und
2. b_i eine Kante hat zu einem Knoten $a_{f(i)} \in \{a_0, \ldots, a_{i-1}\}$.

Nach Voraussetzung kann diese Folge nicht in einem Knoten aus A enden: denn für $S = \{a_0, \ldots, a_{i-1}\}$ gilt $|N(S)| \geq |S|$, also können wir stets eine Kante $\{a, b\} \in E$ finden mit $a \in \{a_0, \ldots, a_{i-1}\}$ und $b \in B \setminus \{b_1, \ldots, b_{i-1}\}$. Sei $b_k \in B$ der letzte Knoten der Folge. Wegen der zwei Eigenschaften ist $P := b_k a_{f(k)} b_{f(k)} a_{f^2(k)} b_{f^2(k)} \ldots a_{f^r(k)}$ mit $f^r(k) = 0$ ein alternierender Pfad.

Wir behaupten, dass b_k in M keinen Partner hat. Denn wenn a ein Partner von b_k wäre und $a = a_i$ für ein $i \in \{1, \ldots, k-1\}$, dann muss gelten $b_k = b_i$, da M eine Paarung ist, ein Widerspruch. Wenn $a \neq a_i$ für alle $i \in \{1, \ldots, k-1\}$, dann könnten wir mit $a_k := a$ die Folge verlängern, im Widerspruch zu deren Maximalität. Also ist b_k ohne Partner und P ein augmentierender Pfad. □

Ein Graph G heißt *k-regulär* falls jeder Knoten in G den Grad k hat.

Korollar 80. *Jeder bipartite k-reguläre Graph, für $k \geq 1$, hat eine perfekte Paarung.*

Beweis. Jede Teilmenge $S \subseteq A$ hat genau $k|S|$ Kanten zu Knoten in $N(S)$. Insgesamt gibt es $k|N(S)|$ Kanten zu Knoten aus $N(S)$. Also gilt $k|S| \leq k|N(S)|$, und damit $|S| \leq |N(S)|$. Der Heiratssatz liefert uns dann eine Paarung von A in G. Klarerweise gilt in regulären Graphen $|A| = |B|$. Also haben wir eine perfekte Paarung in G gefunden. □

Eine weitere Konsequenz des Heiratssatzes ist der Satz von Kőnig. Sei $G = (V, E)$ ein Graph. Eine Menge $U \subseteq V$ wird *Überdeckung* von G genannt, falls jede Kante aus G mindestens einen Knoten aus U enthält.

Satz 81 (Kőnig[44])**.** *Sei G ein bipartiter endlicher Graph. Dann ist die Größe einer größten Paarung in G gleich der Größe einer minimalen Überdeckung von G.*

Beweis. Sei U eine Überdeckung von G minimaler Größe. Sei M eine Paarung in G. Dann benötigt man mindestens $|M|$ Knoten, um M zu überdecken. Also gilt $|U| \geq |M|$. Wir zeigen, dass es eine Paarung der Größe $|U|$ in G gibt. Es sei $U_1 := U \cap A$ die Teilmenge der Überdeckung im einen, und $U_2 := U \cap B$ die Teilmenge der Überdeckung im anderen Teil des bipartiten Graphen G. Wir zeigen nun die Heiratsbedingung aus dem Heiratssatz für die Menge U_1 im bipartiten Graphen $G_1 := G[U_1 \cup B \setminus U_2]$. Dazu müssen wir für beliebiges $S \subseteq U_1$ zeigen, dass $|S| \leq |N(S)|$. Wenn dies nicht so wäre, dann könnten wir

[44] Dénes Kőnig; geboren am 21. September 1884 in Budapest; gestorben am 19. Oktober 1944 in Budapest.

in U die Menge S durch die kleinere Menge $N(S)$ ersetzen, und hätten immer noch eine Überdeckung von G, ein Widerspruch zur Minimalität von U. Der Heiratssatz (Satz 79) liefert uns nun eine Paarung M_1 von U_1 in G_1. Analog erhalten wir eine Paarung M_2 von U_2 im Graphen $G_2 := G[U_2 \cup A \setminus U_1]$. Dann ist $M_1 \cup M_2$ eine Paarung in G, und $|M_1 \cup M_2| = |U_1| + |U_2| = |U|$. $\square$

Eigenschaft	$\leq$	$=$	$<$	$\equiv \bmod n$	$\{(x, x+1) \mid x \in \mathbb{Z}\}$	$\{(x, y) : \lvert x - y \rvert \leq 1\}$
reflexiv	✓	✓		✓		✓
irreflexiv			✓		✓	
symmetrisch		✓		✓		✓
antisymmetrisch	✓	✓	✓		✓	
transitiv	✓	✓	✓	✓		

Abbildung 16: Beispiele zu fundamentalen Eigenschaften von Relationen auf $\mathbb{Z}$.

8 Äquivalenzrelationen

Es sei $R \subseteq A^2 := A \times A$ eine zweistellige (oder *binäre*) Relation auf der Menge A. Statt $(a, b) \in R$ schreibt man oft auch aRb (oder auch $R(a, b)$), und man sagt, dass a und b *in Relation R stehen*. Entsprechend ist eine *n-stellige Relation*, für $n \in \mathbb{N}$, eine Teilmenge von A^n, der Menge aller n-Tupel. Wenn nichts dazu gesagt wird, ist $n = 2$ gemeint.

Es folgen einige wichtige Eigenschaften, die Relationen haben können; Beispiele dazu sind in Abbildung 16 aufgelistet. Eine Relation $R \subseteq A^2$ heißt

- *reflexiv* falls $(a, a) \in R$ für alle $a \in A$;
- *irreflexiv* falls $(a, a) \in R$ für kein $a \in A$ gilt;
- *symmetrisch* falls für alle $(a, b) \in R$ auch $(b, a) \in R$;
- *antisymmetrisch* falls aus $(a, b) \in R$ und $(b, a) \in R$ folgt dass $a = b$;
- *transitiv* falls aus $(a, b) \in R$ und $(b, c) \in R$ folgt dass $(a, c) \in R$.

Definition 82 (Äquivalenzrelation)**.** *Eine Relation $R \subseteq A^2$ heißt* Äquivalenzrelation *auf A, falls R reflexiv, transitiv und symmetrisch ist.*

Das einfachste Beispiel einer Äquivalenzrelation ist die *Gleichheitsrelation*

$$\{(a, a) \mid a \in A\}$$

auf A, die wir auch mit $=_A$ bezeichnen. Äquivalenzrelationen tauchen in fast jedem Kapitel und Abschnitt dieser Vorlesung auf. Bisherige Beispiele sind etwa:

- Gleichmächtigkeit von Mengen. Zwei Mengen A und B stehen in dieser Relation, wenn es eine Bijektion zwischen A und B gibt.
- Die Relation, im selben Zyklus einer Permutation π von $\{1, \ldots, n\}$ zu sein. Zwei Elemente $i, j \in \{1, \ldots, n\}$ stehen in Relation, wenn es ein $k \in \mathbb{N}$ mit $\pi^k(i) = j$ gibt.

- Die Relation $f \sim g$ für Funktionen von $\mathbb{N} \to \mathbb{R}$ mit asymptotisch gleichem Wachstum (diese wurde in der Stirlingschen Formel verwendet, am Ende von Kapitel 2.6).
- Äquivalenz von aussagenlogischen Ausdrücken (Definition 14): zwei Aussagenlogische Ausdrucke A_1 und A_2 über den gleichen Variablen $X_1, \ldots, X_n$ stehen in Relation, wenn sie die gleichen erfüllenden Belegungen haben.
- Kongruenz von Elementen aus $\mathbb{Z}$ modulo n. Zwei ganze Zahlen a, b stehen in Relation, wenn sie kongruent modulo n sind.
- Die Relation auf den Elementen einer Gruppe G, für eine gegebene Untergruppe U von G zur gleichen Nebenklasse von U zu gehören. Zwei Elemente x, y der Gruppe stehen in Relation, wenn $x \circ U = y \circ U$.
- Isomorphie auf der Menge der Graphen. Zwei Graphen stehen in Relation, wenn sie isomorph sind.
- Zusammenhang in einem Graphen. Zwei Knoten a, b stehen in Relation, wenn es einen Pfad von a nach b im Graphen gibt.

Übungen.

29. Zeigen Sie, dass die obigen Relationen wie behauptet Äquivalenzrelationen bilden.

8.1 Äquivalenz und Partition

In diesem Abschnitt werden wir über den Begriff der Partition eine alternative Sichtweise aus Äquivalenzrelationen kennenlernen.

Definition 83 (Äquivalenzklassen)**.** *Ist R eine Äquivalenzrelation auf A, so definiert man für $a \in A$*

$$a/R := \{b \in A \mid (a, b) \in R\}$$

und nennt diese Menge die Äquivalenzklasse *von a bezüglich R. Für die Menge aller Äquivalenzklassen von Elementen aus A bezüglich R schreiben wir A/R, genannt die* Faktormenge *von A nach R. Also $A/R := \{a/R \mid a \in A\}$.*

Eine weitere übliche Schreibweise für die Äquivalenzklasse von a bezüglich R ist $[a]_R$. Es gibt eine 'kanonische' surjektive Abbildung $f\colon A \to A/R$, die jedem Element seine Äquivalenzklasse bezüglich R zuordnet, $f(a) := a/R$. Äquivalenzklassen sind disjunkt oder gleich: Für je zwei Elemente $a, b \in A$ gilt entweder $a/R \cap b/R = \emptyset$ oder $a/R = b/R$.

Definition 84 (Partition)**.** *Eine* Partition *einer Menge A ist eine Menge $\mathcal{P}$ nicht leerer Teilmengen von A die paarweise disjunkt sind und deren Vereinigung gleich A ist.*

Man nennt die Elemente von $\mathcal{P}$ die *Klassen* der Partition $\mathcal{P}$.

Lemma 85 (Äquivalenz und Partition)**.** *Die Faktormenge A/R einer Äquivalenzrelation R auf einer Menge A ist stets eine Partition. Umgekehrt gilt: ist $\mathcal{P}$ eine Partition von A, dann ist $R_{\mathcal{P}} := \bigcup_{A_i \in \mathcal{P}} A_i \times A_i$ eine Äquivalenzrelation. Es gilt $R = R_{A/R}$ und $A/R_{\mathcal{P}} = \mathcal{P}$.*

Eine Äquivalenzrelation R auf A ist *feiner* als eine Äquivalenzrelation S auf A falls $R \subseteq S$. Gleichbedeutend dazu ist, dass S *gröber* ist als R. Man nennt dann R eine *Verfeinerung* von S und S eine *Vergröberung* von R. Wir schließen hier den Fall mit ein, dass $R = S$. Im Fall, dass R feiner ist als S und $R \neq S$, sagen wir, dass R *echt feiner* ist als S, und entsprechend definieren wir *echt gröber*. Die Relation R ist genau dann feiner als S, wenn jede Äquivalenzklasse von R Teilmenge einer Äquivalenzklasse von S ist. Wir nennen dann auch die zu R gehörende Partition feiner als die zu S gehörige.

Der Schnitt beliebig vieler Äquivalenzrelationen auf A ist wieder eine Äquivalenzrelation auf A. Die Vereinigung von zwei Äquivalenzrelationen R, S auf A ist jedoch im allgemeinen keine Äquivalenzrelation. Es gibt aber stets eine feinste Äquivalenzrelation, die $R \cup S$ enthält (die *transitive Hülle* von $R \cup S$; siehe Abschnitt 9.3).

Übungen.

30. Sei G ein Graph und $L(G)$ sein Kantengraph. Zeigen Sie, dass die folgende Relation $\sim$ auf den Kanten von $L(G)$ eine Äquivalenzrelation bildet: falls $\{e_1, e_2\}$ und $\{f_1, f_2\}$ zwei solche Kanten von $L(G)$ sind, dann definieren wir $\{e_1, e_2\} \sim \{f_1, f_2\}$ falls $\{e_1, f_1\}$, $\{e_1, f_2\}$, $\{e_2, f_1\}$, $\{e_2, f_2\}$ ebenfalls Kanten von $L(G)$ sind. Angenommen, jeder Knoten von G hat Grad 2; wie viele Äquivalenzklassen hat $\sim$?

8.2 Partitionen zählen

Wie viele Möglichkeiten gibt es, n Dinge in k Klassen aufzuteilen? Präziser gefragt: wie viele k-elementige Partitionen hat eine n-elementige Menge? Beispielsweise hat die Menge $\{1, 2, 3\}$ genau drei zweielementige Partitionen, nämlich

$$\{\{1, 2\}, \{3\}\},\ \{\{1, 3\}, \{2\}\},\ \{\{2, 3\}, \{1\}\}.$$

Die Partitionen $\{\{1, 2, 3\}\}$ und $\{\{1\}, \{2\}, \{3\}\}$ dagegen sind nicht zweielementig und werden nicht mitgezählt.

Für die Anzahl der Partitionen einer n-elementigen Menge in k (nicht-leere) Klassen schreiben wir $S_{n,k}$. Wie wir oben gesehen haben, ist beispielsweise $S_{3,2} = 3$. Die Zahlen $S_{n,k}$ nennt man die *Stirling-Zahlen zweiter Art*[45], oder besser die *Stirlingschen Partitionszahlen*. Man berechnet sie auf effiziente Weise rekursiv mit folgender Proposition.

Proposition 86. *Für alle $n, k \in \mathbb{N}$ gilt*

1. $S_{n,n} = 1$;

[45] James Stirling; geboren im Mai 1692 in Garden bei Stirling; gestorben am 5. Dezember 1770 in Edinburgh.

2. $S_{n,k} = 0$ *falls* $n < k$*; und*

3. $S_{n,k} = S_{n-1,k-1} + k \cdot S_{n-1,k}$ *falls* $n \geq 1$.

Beweis. Es gibt nur eine Möglichkeit, n Elemente in n Klassen aufzuteilen. Das gilt auch für $n = 0$, denn die leere Menge ist eine Partition der leeren Menge. Also gilt $S_{n,n} = 1$. Und wenn wir k nicht-leere Klassen bilden wollen, muss n mindestens so groß sein wie k. Es bleibt also bloß die dritte Aussage zu zeigen.

Ohne Beschränkung der Allgemeinheit nehmen wir dass, dass es sich um die n Elemente $\{1, \ldots, n\}$ handelt. Wir betrachten gesondert die Partitionen, in denen die Klasse der 1 nur ein Element hat. Davon gibt es $S_{n-1,k-1}$, denn wir müssen $k-1$ Klassen mit den übrigen Elementen bilden. Alle weiteren Partitionen erhält man, indem man zuerst eine Partition von $\{2, \ldots, n\}$ mit k Klassen bildet, und dann die 1 einer dieser Klassen zuteilt. □

Die Anzahl aller Partitionen einer n-elementigen Menge wird mit B_n bezeichnet; die Zahlen B_n nennt man die *Bellzahlen* (oder auch die *Bellschen Zahlen*)[46]. Sicherlich gilt

$$B_n = \sum_{k=0}^{n} S_{n,k} \,,$$

denn um alle Partitionen einer n-elementigen Menge zu zählen, können wir separat die Partitionen mit genau $k \in \{1, \ldots, n\}$ Klassen zählen, und danach aufsummieren.

Man hat $B_0 = 1$, $B_1 = 1$, $B_2 = 2$, $B_3 = 5$, $B_4 = 15$, $B_5 = 52$. Die Bellschen Zahlen tragen die Nummer A000110 in *Sloane's Oline-Enzyklopädie der Zahlenfolgen (OEIS)*: ein sehr praktisches Hilfsmittel. Denn oft will man kombinatorische Objekte mit n Elementen zählen, und kann dies auch für kleine n, so wie wir eben für die Bellschen Zahlen. Was man aber oft kennen will ist eine geschlossene Formel für die Zahlenfolge, oder eine Rekursionsgleichung zur effizienten Berechnung, oder eine gute Näherungsformel für das asymptotische Wachstum. Dies sind typische Informationen, die in der OEIS Datenbank zu finden sind. Über hinreichend viele Glieder der Folge, oder bisweilen auch über Stichwortsuche kann man häufig die richtige Folge ausfindig machen, und findet dann meist auch alles, was Forscher zu dieser Folge bereits herausgefunden haben.

Für die Bellschen Zahlen kennt man (wie für die Stirlingschen Zahlen) keine einfache explizite Darstellung, aber eine schöne Rekursionsformel.

Proposition 87. *Für* $n \in \mathbb{N}$ *gilt*

$$B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k \,.$$

[46] Eric Temple Bell; geboren am 7. Februar 1883 in Peterhead, Aberdeenshire; gestorben am 21. Dezember 1960 in Watsonville, Santa Cruz County.

Beweis. Um die Anzahl aller Partitionen der Menge $\{1, \ldots, n\}$ zu bestimmen, stellvertretend für eine beliebige Menge mit n Elementen, zählen wir zunächst diejenigen Partitionen, in denen die Klasse der 1 genau $n-k$ Elemente enthält. Wir wählen also zuerst diejenigen Elemente aus, die in der gleichen Klasse liegen wie die 0. Dafür gibt es $\binom{n}{n-k} = \binom{n}{k}$ Möglichkeiten. Danach partitionieren wir die übrigen k Elemente: dafür gibt es B_k Möglichheiten. Jede Partition von $\{1, \ldots, n\}$ wird auf diese Weise für genau ein $0 \in \{1, \ldots, n\}$ mitgezählt, und dies erklärt die Summe in der Rekursionsformel. □

Beispielsweise ist

$$\begin{aligned} B_4 &= \binom{3}{0} B_0 + \binom{3}{1} B_1 + \binom{3}{2} B_2 + \binom{3}{3} B_3 \\ &= 1 \cdot 1 + 3 \cdot 1 + 3 \cdot 2 + 1 \cdot 5 \\ &= 15 \end{aligned}$$

8.3 Der Kern einer Abbildung

Sei $f\colon A \to B$ eine Abbildung. Dann ist der *Kern* von f die Relation auf A die definiert ist durch

$$\operatorname{Kern} f := \{(a_1, a_2) \in A^2 \mid f(a_1) = f(a_2)\}\,.$$

Anders formuliert: zwei Elemente aus A stehen in Relation $\operatorname{Kern} f$, wenn sie von f auf das gleiche Element abgebildet werden. Man rechnet leicht nach, dass $\operatorname{Kern} f$ eine Äquivalenzrelation ist.

Der Begriff des Kerns einer Abbildung ist nicht exakt der gleiche wie der Begriff des Kerns einer *linearen* Abbildung, wie er in der linearen Algebra verwendet wird; beide Begriffe sind aber für solche Abbildungen eng verwandt.

8.4 Funktionen zählen

Seien A und B endliche Mengen. Wir wissen bereits, dass es $|B|^{|A|}$ viele Funktionen von A nach B gibt (Proposition 5). Falls $|A| = |B|$, so wissen wir ebenfalls, dass es $n!$ viele Bijektionen zwischen A und B gibt (Abschnitt 2.6). Für $|B| \leq |A|$ wollen wir nun das eben Gelernte anwenden, um die Anzahl der *surjektiven* Funktionen von A nach B zu berechnen.

Proposition 88. *Seien A, B endlich, $n := |A|$, $m := |B|$, so dass $n \geq m$. Dann ist die Anzahl der surjektiven Funktionen von A nach B gleich $m! \cdot S_{n,m}$.*

Beweis. Alle zu zählenden Funktionen haben einen Kern mit genau m verschiedenen Äquivalenzklassen; diese Klassen sind von der Gestalt $a/(\operatorname{Kern} f) = \{x \in A \mid f(x) = f(a)\}$. Es gibt $m!$ Surjektionen mit dem gleichen Kern R, denn wir müssen alle Bijektionen zwischen A/R und B zählen, und davon gibt es $m!$ viele, da $m = |B| = |A/R|$. Die Anzahl aller Kerne mit m Äquivalenzklassen ist $S_{n,m}$ per Definition. Die angegebene Formel folgt. □

Der Vollständigkeit halber wollen wir nun auch noch die Menge der Injektionen von A nach B zählen.

Proposition 89. *Die Anzahl der injektiven Funktionen von A nach B ist gleich Null, falls $n := |A|$ größer ist als $m := |B|$, und ansonsten gleich*

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

Beweis. Um die Anzahl aller Injektionen von $\{1, \ldots, n\}$ nach $\{1, \ldots, m\}$ zu zählen, bemerken wir zunächst, dass alle solche Funktionen genau $\binom{m}{n}$ viele verschiedene Bilder haben können. Für jedes festgehaltene Bild $C \subseteq B$ gibt es $n!$ viele verschiedene Injektionen, da wir dafür die Bijektionen zwischen A und C zählen müssen. $\square$

9 Ordnungsrelationen

Eine *partielle Ordnung* (oder *Halbordnung*) auf einer Menge A ist eine binäre Relation, die transitiv, reflexiv und antisymmetrisch ist. Wie auch die Äquivalenzrelationen tauchen Ordnungsrelationen in fast jedem Abschnitt der Vorlesung auf. Beispiele sind:

1. Die Teilmengenbeziehung $\subseteq$ (Abschnitt 1).
2. Die Ordnungsrelation $\leq$ der natürlichen Zahlen (Abschnitt 4.1).
3. Die Teilbarkeitsrelation $|$ auf den natürlichen Zahlen (Abschnitt 4.4).
4. Die Untergruppenrelation (Abschnitt 6.5).
5. Die Gleichheitsrelation $=_A$ (Abschnitt 8).

Wenn wir abstrakt über eine Ordnungsrelation schreiben, so verwenden wir typischerweise das Symbol $\leq$. Falls $x \leq y$ und $x \neq y$ gelten, so schreiben wir $x < y$. Die Relation $<$ ist irreflexiv, und immer noch antisymmetrisch und transitiv; solche Relationen werden *strikte Halbordnungen* genannt. Wenn $x \leq y$ oder $y \leq x$ gilt, so sagen wir, dass x und y *vergleichbar* sind, und ansonsten nennen wir x und y *unvergleichbar*.

Hasse-Diagramme. Wenn man partielle Ordnungen bildlich veranschaulichen will, so kann man sich die Transitivität der Ordnung zu Nutze machen, und Hasse-Diagramme verwenden. Wir tragen in solchen Diagramm eine Linie zwischen x und y ein, wenn $x < y$ und es kein z gibt mit $x < z$ und $z < y$. In diesem Fall zeichnen wir y weiter *oben* im Bild ein als x. Es gilt also $x \leq y$ genau dann, wenn es einen aufsteigenden Pfad von x zu y im Diagramm gibt. Diese Diagramme werden *Hasse-Diagramme* genannt[47]. Falls die Grundmenge der partiellen Ordnung endlich ist, dann ist die partielle Ordnung durch das Hasse-Diagramm eindeutig bestimmt.

Betrachten wir zum Beispiel die Teilbarkeitsrelation auf den Teilern der Zahl 60. Es handelt sich hier um eine Menge mit 12 Elementen, nämlich

$$\{1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60\}.$$

Das entsprechende Hasse-Diagramm ist in Abbildung 17 gezeichnet.

Maximale und minimale Elemente. Ein *maximales Element* einer durch $\leq$ partiell geordneten Menge A ist ein Element $x \in A$, so dass für alle $y \in A$ mit $x \leq y$ gilt, dass $x = y$. Wir verlangen nicht, dass $x \geq y$ für alle $y \in A$; es kann also mehr als ein maximales Element geben. Minimale Elemente sind *dual* dazu definiert: man vertauscht dazu in der Definition oben die Symbole $\leq$ und $\geq$.

[47] Helmut Hasse; geboren am 25. August 1898 in Kassel; gestorben am 26. Dezember 1979 in Ahrensburg bei Hamburg.

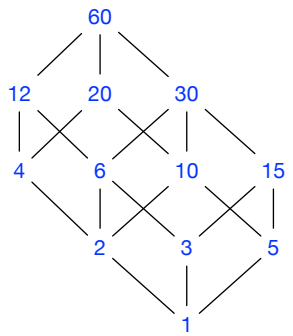


Abbildung 17: Das Hasse-Diagramm der Teilbarkeitsrelation auf den Teilern der Zahl 60.

Lemma 90. *Jede partielle Ordnung einer nicht-leeren endlichen Menge A besitzt ein maximales Element.*

Beweis. Da A nicht leer ist, finden wir ein Element $a_1 \in A$. Wenn a_1 nicht maximal ist, so gibt es ein $a_2 \in A$ mit $a_1 < a_2$. Wenn a_2 nicht maximal ist, so wiederholen wir diesen Schritt, immer wieder. Dieser Prozess muss irgendwann mit einem maximalen Element abbrechen, da A endlich ist. □

9.1 Lineare Erweiterungen

Ein wichtiger Spezialfall von partiellen Ordnungen sind die linearen Ordnungen (diese werden auch *totale* Ordnungen genannt). Sie erfüllen zusätzlich die Bedingung, dass für alle Elemente der Grundmenge x und y entweder $x < y$, $y < x$, oder $x = y$ gilt. Lineare Ordnungen sind also partielle Ordnungen ohne unvergleichbare Elemente. Die Ordnung der natürlichen Zahlen ist natürlich ein Beispiel einer linearen Ordnung.

Jede partielle Ordnung kann zu einer linearen Ordnung erweitert werden; diese Tatsache wird oft als *Satz von Szpilrajn*[48] bezeichnet. Wir zeigen dies hier für partielle Ordnungen auf endlichen Mengen.

Lemma 91. *Sei $R \subseteq A^2$ eine partielle Ordnung auf einer endlichen Menge A. Dann existiert eine lineare Ordnung $R' \subseteq A^2$ mit $R \subseteq R'$.*

Beweis. Wenn $R_0 := R$ keine lineare Ordnung ist, so gibt es unvergleichbare Elemente $x, y \in A$. Es sei $R_1 \subseteq A^2$ eine minimale Menge die $R_0 \cup \{(x, y)\}$ enthält, und die transitiv ist. Die Relation R_1 ist eindeutig und kann explizit beschrieben werden als

$$R_1 \;=\; R_0 \cup \{(u, v) \mid (u, x) \in R_0 \text{ und } (y, v) \in R_0\}\ .$$

[48] Edward Marczewski; geboren am 15. November 1907 in Warschau; gestorben am 17. Oktober 1976 in Breslau; trug bis 1940 den Nachnamen Szpilrajn.

Wenn R_1 eine lineare Ordnung ist, so haben wir die gewünschte Erweiterung gefunden, ansonsten fahren wir mit R_1 anstatt R_0 fort. Da A endlich ist, bricht dieses Verfahren nach endlich vielen Schritten mit der gewünschten Relation R' ab. $\square$

Die Aussage aus Lemma 91 gilt allerdings auch für unendliche Mengen (interessanterweise lässt sich die Aussage in ZFC, nicht aber in ZF zeigen). Zum Beispiel ist die bereits eingeführte lineare Ordnung der natürlichen Zahlen eine lineare Erweiterung der Teilbarkeitsrelation $|$ auf den natürlichen Zahlen, wenn man sie wie folgt abändert: anstatt $0 < x$ für alle $x \geq 1$ soll gelten $x < 0$ für alle $x \geq 1$.

9.2 Quasiordnungen

Oft haben wir es auch mit Relationen zu tun, die zwar transitiv und reflexiv sind, aber nicht antisymmetrisch; auch für solche Relationen gibt es einen speziellen Namen.

Definition 92 (Quasiordnung)**.** *Eine* Quasiordnung *(oft auch* Präordnung*) auf einer Menge A ist eine binäre Relation auf A, die transitiv und reflexiv ist.*

Unser generisches Symbol für eine Quasiordnung ist $\preceq$. Wir schreiben $x \prec y$ falls gilt: $x \preceq y$ und $x \neq y$. Anstatt $x \prec y$ schreiben wir auch $y \succ x$.

Für Quasiordnungen gibt es eine wichtige Konstruktion, die auf eine (antisymmetrische) partielle Ordnung führt. Zunächst stellen wir fest, dass für jede Quasiordnung $\preceq$ die Relation $\sim$ definiert als

$$\{(a, b) \in A^2 \mid a \preceq b, b \preceq a\}$$

eine Äquivalenzrelation ist. Wenn $\preceq$ eine Quasiordnung ist, definieren wir $\succeq$ als die Relation $\{(a, b) \mid b \preceq a\}$; auch die Relation $\succeq$ ist eine Quasiordnung. Es gilt $\sim = (\preceq \cap \succeq)$. Wenn $\preceq$ eine partielle Ordnung wäre, müßte $a \sim b$ genau dann gelten, wenn $a = b$ (wegen der Bedingung der Antisymmetrie). Wir werden nun auf der Menge der Äquivalenzklassen von $\sim$ eine neue Ordnung definieren.

Definition 93 (Faktorordnung)**.** *Es sei $\preceq$ eine Quasiordnung auf der Menge A und $\sim :=$ $(\preceq \cap \succeq)$. Dann definieren wir die Relation $\leq$ auf $A/\!\sim$ wie folgt: es gelte $(a/\!\sim) \leq (b/\!\sim)$ genau dann, wenn $a \preceq b$.*

Dies ist *wohldefiniert*: dazu müssen wir zeigen, dass die Definition von $\leq$ auf den Klassen aus $A/\!\sim$ nicht von der Wahl der Repräsentanten aus diesen Klassen abhängt. Das heißt, wir müssen zeigen, dass für alle $a, b, a', b' \in A$ mit $a \sim a'$ und $b \sim b'$ gilt, dass

$$(a/\!\sim) \leq (b/\!\sim) \text{ genau dann, wenn } (a'/\!\sim) \leq (b'/\!\sim).$$

Nehmen wir also zunächst an, dass $(a/\!\sim) \leq (b/\!\sim)$, das bedeutet per Definition dass $a \preceq b$. Da $a' \preceq a$ und $b \preceq b'$ nach Definition von $\sim$, so gilt $a' \preceq b'$ wegen der Transitivität von $\preceq$, and damit auch $(a'/\!\sim) \leq (b'/\!\sim)$. Der Beweis der umgekehrten Implikation geht analog. Man überzeugt sich leicht, dass $\leq$ eine partielle Ordnung auf $A/\!\sim$ definiert.

9.3 Transitive Hülle

Es sei R eine binäre Relation auf einer Menge X. Wann gibt es eine partielle Ordnung R' auf X, die R enthält? Wenn es in X für $n \geq 2$ eine Folge $x_0, \ldots, x_{n-1}$ unterschiedlicher Elemente gibt mit $(x_i, x_j) \in R$ falls $j = i + 1 \mod n$, dann kann es so ein R' sicher nicht geben. Denn wegen der Transitivität von R' wäre dann neben (x_0, x_1) auch (x_1, x_0) in R', was aber wegen der Antisymmetrie von R' nicht sein kann.

Sicher aber gibt es immer eine Quasiordnung R' auf X, die R enthält: zum Beispiel die volle Relation X^2 hat diese Eigenschaft. Für Relationen $R, S \subseteq X^2$ definieren wir $R \circ S$ als die binäre Relation

$$R \circ S := \{(a, b) \mid \text{ es gibt ein } c \in X \text{ mit } (a, c) \in R \text{ und } (c, b) \in S\}.$$

Die Relation $R \circ S$ heißt auch die *Komposition* von R und S. Für $R \circ R$ schreiben wir auch R^2, und für $n \geq 2$ ist R^n induktiv definiert als $R^{n-1} \circ R$. Wir vereinbaren außerdem $R^0 := \{(x, x) \mid x \in X\}$.

Lemma 94. *Sei X eine endliche Menge, und $R \subseteq X^2$ eine Relation. Dann sind äquivalent:*

1. *$R' \subseteq X^2$ ist die* kleinstmögliche *Quasiordnung, die R enthält.*
2. *$R' = \bigcap\{S \subseteq X^2 \mid S$ Quasiordnung mit $R \subseteq S\}$.*
3. *$R' = \bigcup_{i \in \mathbb{N}} R^i$.*

Beweis. 1. $\Leftrightarrow$ 3.: Es sei R' wie in 1. Jede Quasiordnung, die R enthält, muss auch R^i enthalten, für alle $i \in \mathbb{N}$, da sie transitiv ist. Also gilt $\bigcup_{i \in \mathbb{N}} R^i \subseteq R'$. Zum anderen ist $\bigcup_{i \in \mathbb{N}} R^i$ sicher eine Quasiordnung, die R enthält. Da R' die kleinste solche Quasiordnung ist, gilt $R' \subseteq \bigcup_{i \in \mathbb{N}} R^i$. Damit ist insgesamt die Gleichheit bewiesen.

1. $\Leftrightarrow$ 2.: Übung. □

Die Relation R' wird auch die *transitive reflexive Hülle* von R genannt. Die transitive Hülle von R ist analog definiert, und es gibt ein analoges Lemma dafür (Aufforderung: wie genau ist die Formulierung der entsprechenden Variante von Lemma 94?).

9.4 Ketten, Antiketten, Dilworth

Eine *Kette* in einer partiellen Ordnung $\leq$ auf einer Menge X ist eine Menge $K \subseteq X$ paarweise *vergleichbarer* Elemente aus X; damit meinen wir, dass $a \leq b$ oder $b \leq a$ für alle $a, b \in K$ gilt. Eine *Antikette* ist eine Teilmenge $A \subseteq X$ paarweise *unvergleichbarer* Elemente aus X; damit meinen wir, dass für alle verschiedenen $a, b \in K$ weder $a \leq b$ noch $b \leq a$ gilt; wir schreiben in diesem Fall $a \perp b$ (die Relation $\perp$ ist symmetrisch).

Satz 95. *Sei $\leq$ eine partielle Ordnung einer unendlichen Menge A. Dann gibt es in A eine unendliche Kette oder eine unendliche Antikette.*

Beweis. Sei $x_0 \in A$ beliebig. Da A unendlich ist, so ist eine der drei Mengen unendlich (das sogenannte *unendliche Schubfachprinzip*; aus dem Englischen kommend wird mittlerweile auch manchmal von *Taubenschlagprinzip* gesprochen):

$$\begin{aligned} X_{0,<} &:= \{a \in A \mid a < x_0\} \\ X_{0,>} &:= \{a \in A \mid a > x_0\} \\ X_{0,\perp} &:= \{a \in A \mid a \perp x_0\}. \end{aligned}$$

Sei $i_0 \in \{<, >, \perp\}$ so, dass X_{0,i_0} unendlich ist. Wir wiederholen dieses Argument nun mit X_{0,i_0} statt A, und erhalten auf die gleiche Weise ein $i_1 \in \{<, >, \perp\}$ und eine unendliche Menge $X_{1,i_1} \subseteq X_{0,i_0}$. Im Allgemeinen wählen wir für $j \in \mathbb{N}$ ein x_{j+1} aus X_{j,i_j}, und definieren

- $X_{j+1,<} := \{a \in X_{j,i_j} \mid a < x_j\}$,
- $X_{j+1,>} := \{a \in X_{j,i_j} \mid a > x_j\}$, und
- $X_{j+1,\perp} := \{a \in X_{j,i_j} \mid a \perp x_j\}$

Dann ist $X_{j+1,i_{j+1}}$ unendlich für mindestens ein $i_{j+1} \in \{1, 2, 3\}$. Auf diese Weise erhalten wir also eine unendliche Folge $x_0, x_1, x_2, \dots$ von verschiedenen Elementen aus X und eine unendliche Folge $i_0, i_1, i_2, \dots$ von Elementen aus $\{<, >, \perp\}$, so dass

$$X_{0,i_0} \supset X_{1,i_1} \supset X_{2,i_2} \supset \cdots$$

Von den Werten $\{<, >, \perp\}$ muss mindestens einer unendlich oft in der Folge $i_0, i_1, i_2, \dots$ auftauchen (wieder das unendliche Schubfachprinzip!). Seien also $j_1, j_2, \dots \in \mathbb{N}$ so dass $i_{j_1} = i_{j_2} = \cdots$. Dann ist $K := \{x_{j_1}, x_{j_2}, \dots\}$ entweder eine Kette oder eine Antikette:

- Falls $i_{j_1} = i_{j_2} = \cdots = \perp$, so gilt $x_{j_a} \perp x_{j_b}$ für alle $a, b \in \mathbb{N}$ mit $a < b$: Denn dann ist i_{j_a} gleich $\perp$, und daher ist $x_{j_b} \in X_{j_a,\perp}$, also gilt $x_{j_b} \perp x_{j_a}$ und K ist eine Antikette.
- Falls $i_{j_1} = i_{j_2} = \cdots$ gleich $<$ ist, so zeigt man analog, dass $x_{j_a} < x_{j_b}$ für alle $a, b \in \mathbb{N}$ mit $a < b$. Also ist K eine Kette.
- Falls $i_{j_1} = i_{j_2} = \cdots$ gleich $>$ ist, so zeigt man analog, dass $x_{j_a} > x_{j_b}$ für alle $a, b \in \mathbb{N}$ mit $a < b$. Es ist K also wieder eine Kette. □

Ein weiterer schöner Satz für partielle Ordnungen ist der Satz von Dilworth[49].

Satz 96 (Dilworth)**.** *Sei $\leq$ eine partielle Ordnung auf einer endlichen Menge X und k die Größe der größten Antikette in X. Dann kann X in k Ketten partitioniert werden.*

[49] Robert Palmer Dilworth; geboren am 2. Dezember 1914 in Hemet in Kalifornien; gestorben am 29. Oktober 1993 in Kalifornien.

Beweis. Wir verwenden den Satz von König (Satz 81), und definieren dazu den folgenden bipartiten Graphen G. Für jedes Element $x \in X$ schaffen wir in G zwei Knoten, x^- und x^+. Wir fügen in G eine Kante zwischen x^- und y^+ ein falls $x < y$.

Sei U eine minimale Überdeckung von G. Dann ist $A := \{x \in X \mid x^+ \notin U \text{ und } x^- \notin U\}$ eine Antikette in X: denn wenn für $x, y \in A$ gilt, dass $x \leq y$, dann wäre $\{x^-, y^+\} \in E(G)$, und damit entweder x^- oder y^+ in U. Nach Voraussetzung gilt $|A| \leq k$. Außerdem haben wir $|A| \geq |X| - |U|$. Also gilt $|X| - |U| \leq k$ und daher $|U| \geq |X| - k$.

Nach dem Satz von König gibt es also eine Paarung $M \subseteq E(G)$ der Größe $|X| - k$. Betrachte die Familie aus Ketten, in welcher zwei Elemente $x_1, x_2 \in X$ in der gleichen Kette sind, falls $\{x_1^+, x_2^-\} \in M$ oder $\{x_1^-, x_2^+\} \in M$. Auf diese Weise erhalten wir $|X| - |M| = |X| - (|X| - k) = k$ disjunkte Ketten. □

Korollar 97 (Dilworth)**.** *Jede partielle Ordnung auf einer Grundmenge der Größe $a(b+1)$ enthält eine Antikette der Länge $a+1$ oder eine Kette der Länge $b+1$.*

Beweis. Es sei A die längste Antikette der partiellen Ordnung. Wenn $|A| > a$, so gibt es nichts zu zeigen; im folgenden nehmen wir also an, dass $|A| \leq a$ ist. Dann gibt es nach dem Satz von Dilworth eine Partition der Grundmenge in $|A|$ Ketten. Da es $a(b+1)$ Elemente gibt, muss eine der Ketten mindestens $b+1$ Elemente besitzen (diesen Beweisschluss nennt man das *endliche Schubfachprinzip*). □

Übungen.

31. Sei $\leq$ eine partielle Ordnung auf einer Menge X und sei $f\colon \mathbb{N} \to X$ eine Funktion. Zeigen Sie: es gibt eine unendliche Teilmenge $S \subseteq \mathbb{N}$ so dass eine der folgenden vier Möglichkeiten zutrifft:
    - $f(i) < f(j)$ für alle $i, j \in S$ mit $i < j$;
    - $f(j) < f(i)$ für alle $i, j \in S$ mit $i < j$;
    - $f(i) = f(j)$ für alle $i, j \in S$;
    - $f(i)$ und $f(j)$ sind unvergleichbar für alle $i \neq j$.

9.5 Wohlquasiordnungen

Eine Quasiordnung $\preceq$ einer Menge X ist eine *Wohlquasiordnung*, falls es zu jeder unendlichen Folge $x_1, x_2, \ldots$ von Elementen aus X Indizes i, j gibt mit $i < j$ und $x_i \preceq x_j$. Wenn $\preceq$ eine Wohlquasiordnung von X ist, dann gibt es insbesondere keine unendlichen Antiketten in X. Weiterhin darf es keine *unendlichen absteigenden Ketten* geben, das heißt, keine Folgen von Elementen $x_0, x_1, x_2, \ldots$ mit $x_i \succ x_j$ für alle $i, j \in \mathbb{N}$ mit $i < j$.[50]

Beispiele:

[50] Formal ist eine *Folge von Elementen aus X* eine Abbildung $f\colon \mathbb{N} \to X$; solche Funktionen gibt man dann häufig mit der Pünktchenschreibweise als $f(0), f(1), f(2), \ldots$ an.

- Jede Quasiordnung einer endlichen Menge ist eine Wohlquasiordnung.
- $(\mathbb{N}; \leq)$, und allgemeiner jede (lineare) Wohlordnung (Abschnitt 4.1) ist eine Wohlquasiordnung.

Gegenbeispiele:

- $(\mathbb{Z}; \leq)$ ist keine Wohlquasiordnung: die Folge $-1, -2, -3, \ldots$ ist eine unendliche absteigende Kette.
- $(\mathbb{N}; |)$ ist keine Wohlquasiordnung: die Teilmenge der Primzahlen bildet eine unendliche Antikette.

Lemma 98. *Eine Quasiordnung ist genau dann eine Wohlquasiordnung, wenn sie keine unendlichen Antiketten und keine unendlichen absteigenden Ketten besitzt.*

Beweis. $\Rightarrow$: trivial.
$\Leftarrow$: Folgt aus dem Beweis von Satz 95. Weil der so schön war, geben wir ihn hier nochmal explizit für unsere Situation an. Sei $x_1, x_2, \ldots$ eine unendliche Folge, so dass es keine Indizes i, j gibt mit $i < j$ und $x_i \preceq x_j$. Das heißt, für alle i, j mit $i < j$ gilt entweder dass $x_i \succ x_j$ oder dass x_i und x_j unvergleichbar sind. Wir unterscheiden zwei Fälle:

1. x_1 ist unvergleichbar zu x_j für unendlich viele $j = j_1, j_2, \ldots$ In diesem Fall färbe x_1 blau.
2. Ansonsten muss gelten: $x_1 \succ x_j$ für unendlich viele $j = j_1, j_2, \ldots$. In diesem Fall färbe x_1 rot.

Ersetze nun x_1 durch x_{j_1}, und die Folge $x_1, x_2, \ldots$ durch die (unendliche!) Folge $x_{j_1}, x_{j_2}, \ldots$, und wiederhole die gleiche Fallunterscheidung. So fahren wir ohne Ende fort, und erhalten auf diese Weise eine Folge eingefärbter Elemente $z_1, z_2, \ldots$.

Nun muss es in der Folge $z_1, z_2, \ldots$ unendlich viele blaue oder unendlich viele rote Elemente geben. Im ersten Fall haben wir eine unendliche Antikette, im zweiten Fall eine unendliche Kette gefunden. $\square$

Sei $n \in \mathbb{N}$ beliebig. Wir betrachten nun die folgende partielle Ordnung $\preceq$ von $\mathbb{N}^n$, der Menge der n-Tupel von natürlichen Zahlen: setze $(a_1, \ldots, a_n) \preceq (b_1, \ldots, b_n)$ falls $a_i \leq b_i$ für alle $i \in \{1, \ldots, n\}$.

Lemma 99 (Dickson[51])**.** *Die Ordnung $\preceq$ auf $\mathbb{N}^n$ ist eine Wohlquasiordnung.*

[51] Leonard Eugene Dickson; geboren am 22. Januar 1874 in Independence, Iowa; gestorben am 17. Januar 1954 in Harlingen, Texas.

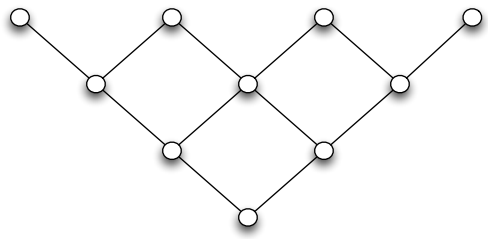


Abbildung 18: Das Hasse-Diagramm eines unteren Ausschnitts unserer Ordnung $\preceq$ von $\mathbb{N}^2$.

Beweis. Wir beweisen hier nur den Spezialfall für $n = 2$. Zu zeigen ist, dass für alle unendlichen Folgen $(x_1, y_1), (x_2, y_2), \ldots$ aus $\mathbb{N}^2$ Indizes i, j existieren mit $i < j$ und $(x_i, y_i) \preceq (x_j, y_j)$, d.h., $x_i \leq y_i$ und $y_i \leq y_j$. Es sei nun eine beliebige Folge $(x_1, y_1), (x_2, y_2), \ldots$ gegeben. Für eine übersichtlichere Notation verwenden wir die Funktionen $f, g\colon \mathbb{N} \to \mathbb{N}$ mit $f(i) := x_i$ und $g(i) := y_i$. Sei $i_0 \in \mathbb{N}$ so gewählt, dass $f(i_0) = \min\{f(\ell) \mid \ell \in \mathbb{N}\}$. Induktiv definieren wir $i_{n+1} \in \mathbb{N}$ als eine Zahl größer als i_n so dass $f(i_{n+1}) = \min\{f(\ell) \mid \ell > i_n\}$. Dann gilt $i_0 < i_1 < \cdots$ und $f(i_0) \leq f(i_1) \leq \cdots$. Die Folge $g(i_0), g(i_1), \ldots$ in $\mathbb{N}$ kann nicht streng absteigend sein. Also gibt es ein $k \in \mathbb{N}$ mit $g(i_k) \leq g(i_{k+1})$. Die Indizes $i := i_k$ und $j := i_{k+1}$ leisten dann das Gewünschte. $\square$

Für manche Berechnungsprobleme kann man zeigen, dass sie von einem effizienten Algorithmus gelöst werden können, ohne dass man einen solchen Algorithmus tatsächlich angeben könnte. Manche solcher Existenzbeweise für Algorithmen lassen sich mit Hilfe von Wohlquasiordnungen führen.

Sei $\preceq$ eine Wohlquasiordnung einer Menge X, und $S \subseteq X$. Dann schreiben wir $\uparrow S$ für $\{y \in X \mid \text{es existiert } x \in S \text{ mit } x \preceq y\}$.

Proposition 100. *Sei $\preceq$ eine Wohlquasiordnung einer Menge X, und $S \subseteq X$. Dann gibt es eine endliche Menge $B \subset X$ so dass $\uparrow B = \uparrow S$.*

Beweis. Sei B minimal bezüglich Inklusion mit der Eigenschaft, dass $\uparrow B = \uparrow S$. Dann kann B keine Elemente a, b enthalten mit $a \preceq b$, da ansonsten $\uparrow(B \setminus \{b\}) = \uparrow S$, im Widerspruch zur Minimalität von B. Also ist B eine Antikette. Da $\preceq$ eine Wohlquasiordnung ist, kann B also nicht unendlich sein. $\square$

Wir bemerken, dass wir in diesem Beweis nur die Existenz einer endlichen Menge B von Objekten in X gezeigt haben, ohne eine wirkliche Vorschrift zu kennen, wie man diese Menge in konkreten Situationen findet.

Ordnungen auf Wörtern. Sei A eine beliebige endliche Menge. Ein *Wort* der Länge n über dem Alphabet A ist eine Abbildung von $\{1, \ldots, n\}$ nach A. Wenn w ein Wort der

Länge n ist, dann schreiben wir auch oft $w(1)w(2)\dots w(n)$ für dieses Wort, und $|w|$ für n. Diese Schreibweise ist praktisch, um Wörter anzugeben. Zum Beispiel steht $w := abba$ für die Funktion $w\colon \{1,2,3,4\} \to A$ mit $w(1) = a$, $w(2) = b$, $w(3) = b$, $w(4) = a$, und $|abba| = 4$. Für die Menge aller Wörter endlicher Länge über dem Alphabet A schreiben wir A^*.

- Die *Präfixordnung* ist definiert wie folgt: wenn $u, v \in A^*$, dann ist u ein *Präfix* von v falls $|u| \leq |v|$ und falls $u(i) = v(i)$ für alle $i \in \{1, \dots, |u|\}$. Siehe Abbildung 19. Bezüglich dieser Ordnung ist (7) eine Antikette. Offensichtlich gibt es keine unendlichen absteigenden Ketten.

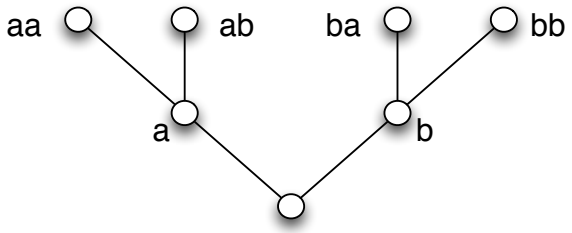


Abbildung 19: Das Hasse-Diagramm eines unteren Ausschnitts der Präfixordnung auf $\{a, b\}^*$.

- Es sei $\leq$ eine lineare Ordnung von A. Für die *lexikographische Ordnung* $\leq_{\text{lex}}$ von A^* (die Ordnung im Wörterbuch) nehmen wir an, dass wir bereits eine lineare Ordnung $\leq$ auf dem Alphabet A festgelegt haben. Diese Ordnung ist wie folgt definiert: für $u, v \in A^*$ gilt $u \leq_{\text{lex}} v$ falls u ein Päfix ist von v, oder falls ein $i \in \{1, \dots, |u|\}$ existiert so dass $u_j = v_j$ für alle $j < i$, und falls $u_i < v_i$. Die lexikographische Ordnung ist sogar eine lineare Ordnung, ist aber für $|A| \geq 2$ keine Wohlquasiordnung: betrachte dazu die unendliche absteigende Folge von Wörtern

$$b, ab, aab, aaab, \dots \tag{7}$$

- Die *Teilfolgenordnung* (oder *Teilwortordnung*) $\preceq$ auf A^* ist wie folgt definiert. Für zwei Wörter $w_1, w_2 \in A^*$ gilt $w_1 \preceq w_2$ falls w_1 aus w_2 durch 'Wegstreichen von beliebig vielen Buchstaben' (an beliebigen Stellen) entsteht. Formal heißt das, dass es $i_1, \dots, i_k \in \{1, \dots, k\}$ mit $k = |w_2|$ gibt, so dass $i_1 < \dots < i_k$ und $w_1(j) = w_2(i_j)$ für alle $j \in \{1, \dots, |w_1|\}$. Offensichtlich gibt es wieder keine unendlichen absteigenden Ketten.

Wir geben den folgenden Satz ohne Beweis an.

Satz 101. *Sei A eine endliche Menge. Dann ist die Teilfolgenordnung eine Wohlquasiordnung von A^*.*

Wir betrachten nun folgende Anwendung. Sei $L \subseteq A^*$ eine beliebige Menge von Wörtern. Wir zeigen nun die Existenz eines Algorithmus, der als Eingabe ein endliches Wort $w \in A^*$ erhält, und der entscheidet, ob $w \in \uparrow L$. Das bedeutet, dass der Algorithmus nach endlicher Zeit anhält, und 'Ja' ausgibt, falls $w \in \uparrow L$, und 'Nein' ausgibt, falls $w \notin \uparrow L$. Wir sagen in diesem Fall auch, dass der Algorithmus $\uparrow L$ *entscheidet*.

Da die Teilfolgenrelation von A^* eine Wohlquasiordnung ist, so gibt es nach Proposition 100 eine endliche Menge B von Wörtern aus A^* mit $\uparrow B = \uparrow L$. Unser Algorithmus muss also für ein gegebenes Wort nur testen, ob w ein Wort aus B als Teilfolge enthält. Das kann man sogar effizient und in polynomieller Zeit tun.

Wir wissen also, dass es den gewünschten Algorithmus für $\uparrow L$ *gibt*. Aber wir kennen ihn im allgemeinen nicht! Selbst wenn wir einen Algorithmus kennen, der L entscheidet, so kennen wir damit noch keinen Algorithmus, der $\uparrow L$ entscheidet, obwohl wir wissen, dass es sogar einen effizienten Algorithmus dafür gibt.

Übungen.

32. Wir verallgemeinern den Begriff der lexikographischen Ordnung auf A^* wie folgt: wir gehen nun aus von einer *Quasiordnung* auf A, und belassen sonst die Definition. Zeigen Sie:
    - die resultierende Ordnung $\leq_{\text{lex}}$ ist selbst wieder eine Quasiordnung.
    - falls die Ordnung auf A sogar eine partielle Ordnung ist, dann ist $\leq_{\text{lex}}$ ebenfalls eine partielle Ordnung.

9.6 Semilineare Ordnungen

Eine *semilineare Ordnung* $\leq$ einer Menge X ist eine partielle Ordnung von X, so dass

- für alle $x, y \in X$ existiert ein $z \in X$ mit $z \leq x$ und $z \leq y$;
- für jedes $x \in X$ ist die Menge $\{y \in X \mid y \leq x\}$ linear geordnet.

Ein Beispiel für eine semilineare Ordnung ist die Präfixordnung auf A^* (Abschnitt 9.5). Offensichtlicherweise sind auch alle linearen Ordnungen semilinear. Die Teilwortrelation dagegen ist nicht semilinear: Wir haben $a \preceq ab$, $a \preceq ba$, $ab \preceq aba$, $ba \preceq aba$, und ba und ab sind unvergleichbar.

Sei (V, E) ein Baum, und $r \in V$. Den Baum zusammen mit diesem ausgezeichneten Knoten r nennt man einen *gewurzelten* Baum, und r die *Wurzel*. Für einen gewurzelten Baum definieren wir die Nachfahrenrelation $\leq$ wie folgt. Für $x, y \in V$ definieren wir $x \leq y$ falls x auf dem (eindeutigen; siehe Lemma 61) Pfad von r nach y liegt. In diesem Fall heißt x ein *Vorfahre* von y und y ein *Nachfahre* von x.

Lemma 102. *Die Nachfahrenrelation $\leq$ eines gewurzelten Baumes ist semilinear.*

Beweis. Es gilt $r \leq x$ für alle $x \in V$, und also gilt die erste Bedingung. Um die zweite Bedingung nachzuweisen fixieren wir $x \in V$ beliebig, und betrachten zwei Elemente $y_1, y_2 \in V$ mit der Eigenschaft dass $y_1 \leq x$ und $y_2 \leq x$. Das bedeutet, sowohl y_1 als auch y_2 liegen auf dem Pfad P von r nach x. Auf P kommt entweder zuerst y_1 und dann y_2, oder zuerst y_1 und dann y_2, oder es gilt $y_1 = y_2$. Also wird die Menge $\{y \in V \mid y \leq x\}$ tatsächlich von $\leq$ linear geordnet. □

Umgekehrt gilt folgendes.

Lemma 103. *Das Hasse-Diagramm einer endlichen semilinearen Ordnung ist ein Baum.*

Beweis. Sei $\leq$ eine semilineare Ordnung auf einer endlichen Menge X, und (X, E) der Graph des Hasse-Diagramms von $\leq$. Dann ist (X, E) zusammenhängend: denn mit $x, y \in X$ gibt es ein $z \in X$ mit $z \leq x$ und $z \leq y$. Mit $z \leq x$ gibt es einen Pfad von z nach x im HasseDiagramm, und ebenso gibt es einen Pfad von z nach y. Also gibt es einen Streckenzug von x nach y. Nehmen wir nun an, es gäbe einen Kreis in (X, E). Wähle einen kürzesten Kreis C aus. Für jede Kante $\{u, v\} \in E$ gilt entweder $u \leq v$ oder $v \leq u$. Also muss es Knoten x, y, z auf dem Kreis C geben mit $x \leq y$ und $z \leq y$. Da $\leq$ semilinear ist, muss entweder $x \leq z$ oder $z \leq x$ gelten. Das widerspricht der Minimalität von C. □

Wir kommen nun zu einem wichtigen Lemma, das eine Verbindung herstellt zwischen der Existenz von unendlichen Mengen und der Existenz von endlichen Mengen. Sei G ein Baum. Dann heißt G *endlich verzweigend* falls alle Knoten von G endlichen Grad haben.

Lemma 104 (Kőnig's Baum Lemma)**.** *Sei $G = (V, E)$ ein endlich verzweigender unendlicher Baum und $x_0 \in V$. Dann gibt es einen unendlichen Pfad in G, der in x_0 startet.*

Beweis. Da G endlich verzweigend ist, hat x_0 nur endlich viele Nachbarn, und $G - x_0$ hat endlich viele Zusammenhangskomponenten. Da V unendlich ist, so muss es einen Nachbarn x_1 von x_0 geben, der in einer unendlichen Zusammenhangskomponente K von $G - x_0$ liegt. Wir wiederholen dieses Verfahren mit x_1 an der Stelle von x_0, und mit dem Baum $G[K]$ an der Stelle von G, und erhalten einen Nachbarn x_2 von x_1, etc. Auf diese Weise erhalten wir einen unendlichen Pfad $x_0, x_1, x_2, \ldots$ in G. □

Die Annahme in Lemma 104, dass G endlich verzweigend ist, ist notwendig, wie man in Abbildung 20 sieht: der gezeigte Baum enthält *beliebig lange* endliche Pfade, aber keinen *unendlichen* Pfad, der im Knoten x_0 ganz unten startet.

Wir betrachten nun eine der vielen Anwendungen vom Baumlemma von König. Es sei (V, E) der Einheitsabstandsgraph auf $\mathbb{R}^2$: das ist der Graph mit Knotenmenge $V := \mathbb{R}^2$ (wir stellen uns die Knoten als die Punkte der Ebene vor) und Kantenmenge

$$E := \left\{(x, y) \in V^2 \;\middle|\; |x - y| = 1\right\} .$$

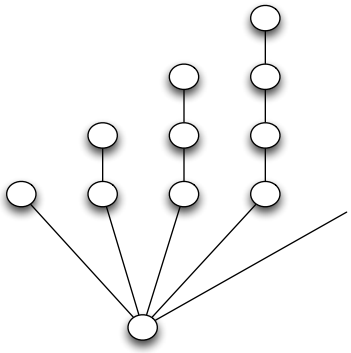


Abbildung 20: Ein unendlich verzweigender Baum.

Zwei Punkte sind also durch eine Kante verbunden wenn sie Abstand eins haben. Wie viele Farben brauchen wir, um diesen Graphen zu färben?

Wir behaupten, dass wir mindestens vier Farben brauchen. Das folgt ganz leicht aus den folgenden drei Beoachtungen.

- Der Graph in Abbildung 21, *Graph von Golomb*[52] genannt, ist ein Subgraph von (V, E).
- Um diesen Subgraphen zu färben, brauchen wir mindestens vier Farben.
- Wenn wir bereits für einen Subgraphen vier Farben brauchen, dann auch für (V, E): und das ist trivial.

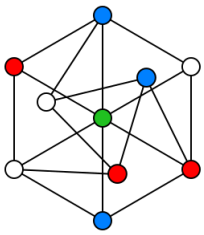


Abbildung 21: Der Zehn-Knoten Graph mit Einheitskanten von Golomb.

Proposition 105. *Es sei G ein unendlicher Graph, dessen endliche Subgraphen allesamt k-färbbar sind. Dann besitzt auch G eine k-Färbung.*

[52] Solomon Wolf Golomb; geboren am 30. Mai 1932 in Baltimore.

Beweis. Wir zeigen die Aussage nur für abzählbare Graphen G (sie gilt aber allgemein). Sei $v_1, v_2, \ldots$ eine Aufzählung von $V(G)$. Es sei X_n die Menge aller Funktionen von $\{v_1, \ldots, v_n\}$ nach $\{1, \ldots, k\}$. Wir betrachten die folgende Ordnung auf $X := \bigcup_{n \in \mathbb{N}} X_n$. Setze $f \leq g$ falls f eine Einschränkung von g ist. Dann ist $\leq$ eine semilineare Ordnung ohne unendlich absteigende Ketten. Außerdem ist das Hasse-Diagramm von $\leq$ endlich verzweigend, da es nur endlich viele Abbildungen von $\{v_1, \ldots, v_{n+1}\}$ nach $\{1, \ldots, k\}$ gibt. Es gibt also nach König's Baum Lemma eine unendliche Kette $g_1, g_2, \ldots$ in X. Definieren $f \colon V(G) \to \{1, \ldots, k\}$ durch $f(v_i) := g_i(v_i)$. Dann ist f eine k-Färbung von G. $\square$

Die Frage nach der kleinsten Zahl an Farben, die man benötigt, um den Einheitsabstandsgraphen auf $\mathbb{R}^2$ zu färben, ist als das Problem von Hadwiger[53]-Nelson[54] bekannt. Es ist bekannt, dass man mit sieben Farben auskommt. Allerdings weiss man nicht, ob es auch mit weniger Farben geht. Nachtrag: Im April 2018 wurde von de Grey[55] ein nicht 4-färbbarer Einheitsabstandsgraph gefunden (mit mehreren Millionen Knoten!). Dass dieser Graph wirklich ein Einheitsabstandsgraph ist und keine 4-Färbung besitzt wurde mit Hilfe von Computern verifiziert. Es bleibt also die Frage: Ist die Antwort 5, 6 oder 7?

Übungen.

33. Verwenden Sie Kőnig's Baumlemma und Satz 95, um folgendes zu zeigen: für jedes $k \in \mathbb{N}$ gibt es ein $n \in \mathbb{N}$, so dass jede partielle Ordnung auf einer n-elementigen Menge eine Kette oder eine Antikette der Größe k enthält.

[53] Hugo Hadwiger; geboren am 23. Dezember 1908 in Karlsruhe, gestorben am 29. Oktober 1981 in Bern.

[54] Edward Nelson; geboren am 4. Mai 1932 in Decatur in Georgia, gestorben am 10. September 2014 in Princeton.

[55] Aubrey de Grey; geboren am 20.4.1963 in London.

10 Bäume zählen

Wie viele Bäume mit n Knoten gibt es? Wenn eine solche Frage beantwortet werden soll, muss man zunächst klären, was genau gemeint ist.

10.1 Beschriftete und unbeschriftete Graphen

Die eingangs gestellte Frage ergibt wörtlich keinen Sinn, da bei einem Baum mit n Knoten ja nicht vorgegeben ist, was genau die Knotenmenge sein soll; es gibt also sicherlich unendlich viele Bäume mit n Knoten. Eine Art, dieses Problem zu umgehen, besteht darin, isomorphe Bäume nur einmal zu zählen. Der Isomorphiebegriff für Graphen wurde in Definition 53 eingeführt; es handelt sich bei $\cong$ um eine Äquivalenzrelation auf der Klasse aller Graphen. Formal zählen wir also die Äquivalenzklassen der Äquivalenzrelation $\cong$ auf der Menge der Bäume mit n Knoten.

Beispiel. Es gibt, wie man durch das Aufstellen einer Liste findet, genau 16 verschiedene Bäume auf der Knotenmenge $V := \{0, 1, 2, 3\}$. Davon sind zwölf isomorph zu P_4, und vier sind isomorph zu einem sogenannten *Stern* mit drei Knoten vom Grad eins, und einem Knoten vom Grad drei. Bis auf Isomorphie gibt es also genau zwei Bäume mit vier Knoten.

Wenn man Graphen bis auf Isomorphie betrachtet, spricht man auch häufig von *unbeschrifteten Graphen*. Oben haben wir zum Beispiel die Anzahl der *unbeschrifteten Bäume* mit vier Knoten gezählt, was einfach die Anzahl der Bäume mit 4 Knoten bis auf Isomorphie bedeuten soll. Formal kann man also einen unbeschrifteten Baum als die Äquivalenzklasse des Baumes bezüglich der Äquivalenzrelation $\cong$ betrachten. Das Problem, die Anzahl der unbeschrifteten Bäume mit n Knoten zu bestimmen, ist ein interessantes (und gelöstes! [2]) mathematisches Problem, das wir aber in dieser Vorlesung nicht lösen werden.

Eine weitere interessante Art, Graphen zu zählen, besteht darin, sich auf Graphen mit der Knotenmenge $\{0, 1, \ldots, n-1\}$ zu beschränken. Man spricht dann davon, *beschriftete Graphen* zu zählen. Im nächsten Kapitel soll es beispielsweise darum gehen, beschriftete Bäume zu zählen.

10.2 Die Formel von Cayley

Wir betrachten hier die Frage, wie viele Bäume (V, E) mit der Knotenmenge $\{0, 1, \ldots, n-1\}$ es gibt. Wir schreiben $t(n)$ für diese Anzahl. Klarerweise ist t eine *monotone* Funktion, das heißt, $t(n) \leq t(n+1)$ für alle $n \in \mathbb{N}$. Für $n > 2$ ist sie sogar *streng monoton*, das heißt, $t(n) < t(n+1)$ (warum?). Zunächst berechnen wir $t(n)$ für kleine n von Hand.

n	1	2	3	4	5
t(n)	1	1	3	16	125

Satz 106 (Formel von Cayley[56]). *Für alle $n \in \mathbb{N} \setminus \{0\}$ gilt $t(n) = n^{n-2}$.*

[56] Arthus Cayley; geboren am 16. August 1821 in Richmond; gestorben am 26. Januar 1895 in Cambridge.

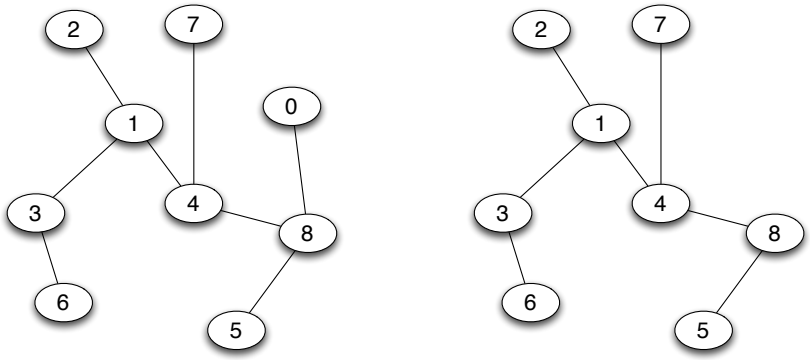


Abbildung 22: Ein Beispiel zur Berechnung des Prüfercodes.

Der Satz besagt, dass es gleich viele Bäume mit der Knotenmenge $\mathbb{Z}_n = \{0, \ldots, n-1\}$ gibt wie Wörter der Länge $n-2$ mit den Buchstaben $\{0, 1, \ldots, n-1\}$. Ein Beweis kann also geführt werden, indem eine bijektive Abbildung zwischen diesen beiden Mengen angegeben wird. Dazu werden wir eine Vorschrift angeben, die jedem Baum mit Knotenmenge $\mathbb{Z}_n$ auf eindeutige und umkehrbare Weise ein Wort der Länge $n-2$ zuordnet, den sogenannten *Prüfercode*[57].

Prüfercodes. Sei $n \in \mathbb{N} \setminus \{0, 1\}$ beliebig. Der Prüfercode ordnet jedem Baum (V, E) mit linear geordneter n-elementiger Knotenmenge V ein Wort der Länge $n-2$ mit Buchstaben aus V zu, und zwar wie folgt:

- Falls $n = 2$ ist, wird dem Baum das leere Wort (das Wort der Länge 0) zugeordnet.
- Falls $n > 2$ ist, sei b das kleinste Blatt des Baums und a_0 der zu b adjazente Knoten. Es sei w der Prüfercode des Baumes $(V, E) - b$. Dieser Baum ist kleiner, wir nehmen also induktiv an, dass wir w bereits kennen. Dem Baum (V, E) ordnen wir dann den Prüfercode $a_0 w$ zu.

Beispiel. Betrachten wir den Baum in Abbildung 22. Das kleinste Blatt ist das mit der Nummer 0. Der dazu adjazente Knoten ist 8. Der Prüfercode beginnt also mit 8. Für den nächsten Schritt wird das Blatt 0 entfernt. Nun ist 2 das kleinste Blatt. Der dazu adjazente Knoten ist 1. Der Prüfercode beginnt also mit 81. Für den nächsten Schritt wird das Blatt 2 entfernt, und so weiter. Als Prüfercode des Baumes aus Abbildung 22 erhält man so 8183144.

[57] Ernst Paul Heinz Prüfer; geboren am 10. November 1896 in Wilhelmshaven; gestorben am 7. April 1934 in Münster.

Beobachtungen.

- Der Prüfercode ist ein Wort der Länge $n-2$ mit Buchstaben aus V.
- Die Blätter des Baumes kommen im Prüfercode nicht vor. Das ist aufgrund der Konstruktion offensichtlich: im Tupel werden ja nur Knoten notiert, die zu einem Blatt adjazent sind. Solche Knoten können aber für $n > 2$ keine Blätter sein, und für $n = 2$ werden sie nicht notiert.
- Alle Knoten, die nicht Blätter sind, tauchen im Prüfercode auf, da irgendwann einer der Nachbarn aus dem Baum entfernt wird.

Also gilt folgendes.

Lemma 107. *Sei (V, E) ein Baum. Die Blätter von (V, E) sind genau diejenigen Knoten, die im Prüfercode nicht vorkommen.*

Dekodieren. Wie gelangt man nun vom Prüfercode zurück zum Baum? Gegeben sei ein Wort $a_0 a_1 \dots a_{n-3}$ der Länge $n-2$ mit Buchstaben aus V, und $|V| = n$. Dem leeren Wort (im Falle $n = 2$) ordnen wir den eindeutigen Baum auf den zwei Knoten aus V zu. Im Falle $n > 2$ bestimmen wir ein Wort $b_0 b_1 \dots b_{n-3}$ wie folgt:

- b_0 sei das kleinste Element von $V \setminus \{a_0, a_1, \dots, a_{n-3}\}$.
- Für $i \in \{0, \dots, n-4\}$, sei b_{i+1} das kleinste Element aus $V \setminus \{b_0, \dots, b_i, a_{i+1}, \dots, a_{n-3}\}$.

Außerdem sei $e := V \setminus \{b_0, b_1, \dots, b_{n-3}\}$. Der gesuchte Baum hat dann die Kantenmenge

$$E := \big\{\{a_0, b_0\}, \dots, \{a_{n-3}, b_{n-3}\}, e\big\} .$$

Beispiel. Gegeben sei $V := \{0, 1, \dots, 8\}$, $n = 9$, und $a_0 a_1 \dots a_{n-3} := 8183144$. Wir berechnen $b_0 b_1 \dots b_{n-3}$ nach der angegebenen Regel.

0. $b_0 := \min(\{0, 1, \dots, 8\} \setminus \{8, 1, 3, 4\}) = 0$;
1. $b_1 := \min(\{0, 1, \dots, 8\} \setminus \{0\} \setminus \{1, 8, 3, 4\}) = 2$;
2. $b_2 := \min(\{0, 1, \dots, 8\} \setminus \{0, 2\} \setminus \{8, 3, 1, 4\}) = 5$;
3. $b_3 := \min(\{0, 1, \dots, 8\} \setminus \{0, 2, 5\} \setminus \{3, 1, 4\}) = 6$;
4. $b_4 := \min(\{0, 1, \dots, 8\} \setminus \{0, 2, 5, 6\} \setminus \{1, 4\}) = 3$;
5. $b_5 := \min(\{0, 1, \dots, 8\} \setminus \{0, 2, 5, 6, 3\} \setminus \{4\}) = 1$;
6. $b_6 := \min(\{0, 1, \dots, 8\} \setminus \{0, 2, 5, 6, 3, 1\} \setminus \{4\}) = 7$.

Wir erhalten

$$\begin{aligned} a_0a_1\dots a_{n-3} &= 8183144 \\ b_0b_1\dots b_{n-3} &= 0256317 \end{aligned}$$

Dann ist $V \setminus \{b_0, b_1, \dots, b_{n-3}\} = \{4, 8\}$, und

$$E := \big\{\{8,0\}, \{1,2\}, \{8,5\}, \{3,6\}, \{1,3\}, \{4,1\}, \{4,7\}, \{4,8\}\big\}\,.$$

Satz 108. *Jedes Wort der Länge $n-2$ mit Buchstaben aus V ist der Prüfercode von genau einem Baum mit Knotenmenge V.*

Beweis. Per Induktion über n. Für den kleinstmöglichen Fall, $n = 2$, ist die Behauptung offenbar richtig, sei also nun $n > 2$. Sei $a_0a_1\dots a_{n-3}$ ein Wort mit Buchstaben aus V. Nach der Induktionsvoraussetzung ist $a_1a_2\dots a_{n-3}$ Prüfercode genau eines Baumes T mit der Knotenmenge $V \setminus \{b_0\}$. Die Blätter von T sind nach Lemma 107 genau die Elemente aus $V \setminus \{b_0\}$, die nicht in $\{a_1, a_2, \dots, a_{n-3}\}$ vorkommen. Dann gilt:

$$\begin{aligned} & b_0 = \min(V \setminus \{a_0, a_1, \dots, a_{n-3}\}) \\ \Leftrightarrow\ & b_0 \text{ ist kleiner als alle Blätter in } T \\ \Leftrightarrow\ & \text{der Baum } (V, E(T) \cup \{a_0, b_0\}) \text{ hat Prüfercode } a_0a_1\dots a_{n-3}\,. \end{aligned}$$

Es gibt also genau einen Baum mit Prüfercode $a_0a_1\dots a_{n-3}$. □

Es ist damit auch Satz 106 bewiesen, denn wir haben eine Bijektion konstruiert zwischen den Bäumen mit einer Knotenmenge V der Größe n auf der einen Seite, und den Wörten der Länge $n-2$ mit Buchstaben aus V auf der anderen Seite. Von solchen Wörtern gibt es klarerweise n^{n-2} viele.

In Abschnitt 9.6 hatten wir *gewurzelte* Bäume kennengelernt: das waren Bäume im graphentheoretischen Sinn zusammen mit einem ausgezeichneten Knoten, der *Wurzel*. Also folgt aus Satz 106, dass es n^{n-1} gewurzelte Bäume mit Knotenmenge V gibt.

Interessanterweise kannte man die Formel von Cayley bereits vor dem bijektiven Beweis von Prüfer; es existieren rein algebraische Beweise. In der enumerativen Kombinatorik ist der Zusammenhang zwischen Bäumen mit Knotenmenge $\mathbb{Z}_n$ und Wörtern der Länge $n-2$ mit Buchstabe $\{0, 1, \dots, n-1\}$ nur ein Beispiel von unzähligen weiteren bijektiven Zusammenhängen; wir verweisen auf Stanley [8] für weiterführende Literatur.

10.3 Der Satz von Kirchhoff

Ein *Spannbaum* (zu Deutsch auch *Gerüst* oder *aufspannender Baum*) eines Graphen (V, E) ist ein Baum (V, F) mit $F \subseteq E$. Ein Spannbaum von G ist also ein Baum, der die gleiche Knotenmenge und eine Teilmenge der Kantenmenge von G hat. In Abbildung 23 findet

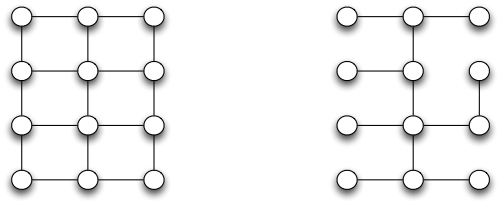


Abbildung 23: Ein Graph und einer seiner Spannbäume.

sich ein Beispiel von einem Graph G und einem Spannbaum von G. Offensichtlich hat ein endlicher Graph genau dann einen Spannbaum, wenn er zusammenhängend ist.

Wir wollen nun untersuchen, wie man die *Anzahl* der Spannbäume für einen gegebenen Graphen berechnen kann. Das ist der Gegenstand vom Satz von Kirchhoff[58], auch bekannt als der *Matrix-Gerüst-Satz*.

Zunächst eine wichtige Bemerkung. Ein Graph kann sehr viele Spannbäume haben. Der Graph K_n zum Beispiel hat, wie wir wissen, n^{n-2} Spannbäume. Wir können im allgemeinen also sicher nicht die Anzahl ermitteln, indem wir der Reihe nach alle Spannbäume ausrechnen und mitzählen – selbst wenn wir dafür einen Computer programmieren, wird das für wachsende Knotenmenge ganz schnell unpraktikabel. Der Satz von Kirchhoff verwendet auf wundervolle Weise Begriffe aus der linearen Algebra, um dieses Zählproblem effizient und elegant zu lösen.

Definition 109. *Sei $G = (\{1, \ldots, n\}, E)$ ein Graph. Die* Adjazenzmatrix *von G ist die $(n \times n)$-Matrix $A := (a_{i,j})_{i,j \in \{1,\ldots,n\}}$ mit $a_{i,j} := 1$ falls $\{i, j\} \in E$, und $a_{i,j} := 0$ sonst.*

Zum Beispiel hat der Pfad der Länge vier, P_4, die folgende Adjazenzmatrix.

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Die Adjazenzmatrix

- ist symmetrisch, d.h., $a_{ij} = a_{ji}$ für alle $i, j \in \{1, \ldots, n\}$;
- hat Nullen in der Hauptdiagonalen, d.h., $a_{ii} = 0$ für alle $i \in \{1, \ldots, n\}$.

Die Spalten- und Zeilensummen sind gleich dem jeweiligen Knotengrad:

$$\sum_{i \in \{1,\ldots,n\}} a_{ij} = \sum_{i \in \{1,\ldots,n\}} a_{ji} = \mathrm{Grad}(j)$$

[58] Gustav Robert Kirchhoff; geboren am 12. März 1824 in Königsberg; gestorben am 17. Oktober 1887 in Berlin.

Definition 110. *Als* Gradmatrix *von* $(\{1,\dots,n\},E)$ *bezeichnen wir die Diagonalmatrix* $D := (d_{i,j})_{i,j\in\{1,\dots,n\}}$ *mit* $d_{i,j} := 0$ *falls* $i \neq j$ *und* $d_{i,i} = \mathrm{Grad}(i)$ *sonst.*

Für den Graphen P_4 beispielsweise ist die Gradmatrix wie folgt.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Wenn B eine Matrix ist, dann schreiben wir $(B)_i$ für die Matrix, die aus B entsteht durch Streichen der i-ten Zeile und der i-ten Spalte.

Satz 111 (Kirchhoff)**.** *Für* $n \geq 2$*, sei* $G := (\{1,\dots,n\},E)$ *ein Graph,* A *seine Adjazenzmatrix,* D *seine Gradmatrix, und* $i \in \{1,\dots,n\}$ *beliebig. Dann ist die Anzahl der Spannbäume von* G *gleich der Determinante der Matrix* $(D-A)_i$*.*

Das ist eine ausgezeichnete Nachricht, denn Determinanten von Matrizen lassen sich schnell ausrechnen. Für unser Beispiel $G = P_4$ erhalten wir:

$$D - A = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \text{ und } (D-A)_1 = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}$$

und $\det(D-A)_1$ berechnet sich wie folgt

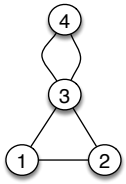
$$\begin{aligned} & 2\cdot\big(2\cdot 1-(-1)(-1)\big)-(-1)\cdot\big((-1)\cdot 1-(0\cdot(-1))\big) \\ = \;& 2\cdot(2-1)+1\cdot(-1+0) \\ = \;& 1\,. \end{aligned}$$

Tatsächlich hat P_4 genau einen Spannbaum.

Beweisstrategie

Der Beweis argumentiert per Induktion über die Anzahl der Kanten. Wir benutzen dazu zwei Konstruktionen, die aus einem Graphen einen neuen Graphen mit weniger Kanten machen: nämlich *Löschen* beziehungsweise *Kontrahieren* einer Kante. Wenn G ein Graph ist, und $e \in E(G)$, dann bezeichnet $G - e$ den Graph $(V(G), E(G) \setminus \{e\})$, der also aus G durch Löschen der Kanten e entsteht. Kantenkontraktionen wurden bereits in Definition 68 eingeführt.

Weil es leichter ist, beweisen wir einen etwas allgemeineren Satz. Wir beweisen den Satz nicht nur für Graphen, sondern allgemeiner für Multigraphen, die nun definiert werden.



$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

Abbildung 24: Ein Multigraph und seine Adjazentmatrix und Gradmatrix.

Multigraphen

Wir führen den Beweis für verallgemeinerte Graphen, in denen es zwischen zwei Knoten mehrere Kanten geben darf; wir verbieten aber weiterhin Schleifen (Kanten die einen Knoten mit sich selbst verbinden). Man spricht dann von *Graphen mit Mehrfachkanten* oder, synonym, von *Multigraphen*. Die Graphik in Abbildung 24 zeigt ein Diagramm eines Multigraphen und seine Adjazenz- und Gradmatrix.

Definition 112. *Ein Multigraph G ist ein Tripel $G = (V, E, i)$ wobei*

- *V eine Menge ist,* Knotenmenge *genannt,*
- *E eine Menge ist,* Kantenmenge *genannt, und*
- *$i \colon E \to \binom{V}{2}$ eine Funktion ist.*

Die Idee dieser Definition ist, dass eine Kante $e \in E$ die beiden Knoten in $i(e)$ verbindet; es kann nun allerdings im Gegensatz zu gewöhnlichen Graphen zwei verschiedene Kanten geben, die die gleichen Endpunkte verbinden.

Adjazenz- und Gradmatrix eines Multigraphen sind analog definiert sind wie bei gewöhnlichen Graphen; allerdings erlauben wir nun beliebige natürliche Zahlen als Einträge der Adjazenzmatrix. Ein gewöhnlicher Graph $G = (V, E)$ kann als Multigraph (V, E, id_E) aufgefasst werden. Umgekehrt betrachten wir Multigraphen (V, E, i) bei denen die Funktion i injektiv ist als gewöhnliche Graphen (V, E') wobei E' das Bild ist von i. Multigraphen können daher als Verallgemeinerung von gewöhnlichen Graphen betrachtet werden.

Ein *Spannbaum eines Multigraphen* (V, E, i) ist ein Multigraph mit der gleichen Knotenmenge und einer Teilmenge der Kantenmenge, der wie oben beschrieben als gewöhnlicher Baum betrachtet werden kann. Insbesondere bedeutet das, dass wir beim Zählen von Spannbäumen von Multigraphen *die Kanten unterscheiden.* Zum Beispiel besitzt der Multigraph mit Knotenmenge $\{a, b\}$ und zwei Kanten von a nach b zwei Spannbäume.

Kantenkontraktion

Ist $G = (V, E, i)$ ein Multigraph und e eine Kante, dann bezeichne G/e den Multigraphen, der aus G durch Kantenkontraktion entsteht; das ist analog definiert wie für gewöhnliche

Graphen (Definition 68). Wir nehmen der Einfachheit halber an, dass $V \cap E = \emptyset$. Dann ist G/e der Multigraph (V', E', i') mit

- Knotenmenge $V' := (V \setminus i(e)) \cup \{e\}\}$.
- Kantenmenge $E' := E \setminus \{f \in E \mid i(f) = i(e)\}$.
- Für $f \in E'$ definiere

$$i'(f) := \begin{cases} i(f) & \text{if } i(f) \cap i(e) = \emptyset \\ i(f) \setminus i(e) \cup \{e\} & \text{sonst.} \end{cases}$$

Beispiel: In Abbildung 25 findet sich rechts der Multigraph, der aus dem Graph links durch Kontraktion der Kante e entsteht.

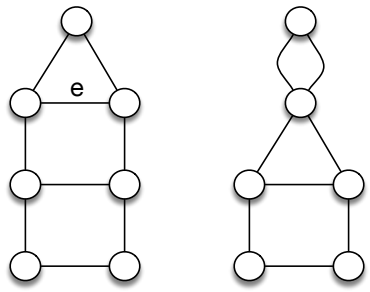


Abbildung 25: Ein Beispiel zur Kantenkontraktion.

Beachte:

- Aus jeder Kante von G, die nicht die Knoten u und v verbindet, entsteht genau eine Kante von G/e, und das sind alle Kanten von G/e.
- Ist T ein Spannbaum von G, welcher die Kante e enthält, dann ist T/e ein Spannbaum von G/e.
- Umgekehrt erhalten wir für jeden Spannbaum T von G/e einen Spannbaum von G durch Hinzunahme der Kante e.
- Die Anzahl der Spannbäume von G, die die Kante e enthalten, ist also gleich der Anzahl der Spannbäume von G/e.
- Die Anzahl der Spannbäume von G, die die Kante e *nicht* enthalten, ist offensichtlich gleich der Anzahl der Spannbäume von $G - e$.

Mit diesen Beobachtungen erhalten wir folgenden Hilfssatz.

Lemma 113. *Ist G ein Multigraph und e eine Kante in G von u nach v, so gilt für die Anzahl $\tau(G)$ der Spannbäume von G:*

$$\tau(G) = \tau(G - e) + \tau(G/e)$$

Weil sowohl $G - e$ als auch G/e weniger Kanten haben als G, ermöglicht der Hilfssatz die rekursive Bestimmung von $\tau(G)$.

Beweis des Satzes von Kirchhoff. Hat der Multigraph G keine Kanten, so ist $D - A$ die Nullmatrix und damit ist $\det(D - A)_i = 0$. Ein kantenloser Multigraph mit mindestens zwei Knoten hat keine Spannbäume, die Behauptung gilt damit in diesem Fall.

Hat G mindestens eine Kante e, so haben wir nach Lemma 113

$$\tau(G) = \tau(G - e) + \tau(G/e) \ .$$

Die Multigraphen $G - e$ und G/e haben beide weniger Kanten als G, und wir können den Satz von Kirchhoff induktiv auf diese Graphen anwenden.

Wir nehmen nun ohne Beschränkung der Allgemeinheit an, dass die Kante e die Knoten 1 und 2 verbindet (ansonsten benennen wir die Knoten um, so dass die Annahme gilt). Weiter sei d_1 der Grad von Knoten 1, und d_2 der Grad von Knoten 2, und d der Grad des Knotens, der bei der Kontraktion von e entsteht. Die Matrix $D - A$ hat dann folgende Gestalt:

$$\begin{pmatrix} d_1 & a & N_1 \\ a & d_2 & N_2 \\ N_1^\top & N_2^\top & M \end{pmatrix} ,$$

die entsprechende Matrix für $G - e$ ist

$$\begin{pmatrix} d_1 - 1 & a + 1 & N_1 \\ a + 1 & d_2 - 1 & N_2 \\ N_1^\top & N_2^\top & M \end{pmatrix}$$

und die für G/e ist

$$\begin{pmatrix} d & N_3 \\ N_3^\top & M \end{pmatrix} .$$

Also haben wir

$$\begin{aligned}
\tau(G) &= \tau(G-e) + \tau(G/e) && \text{Lemma 113} \\
&= \det \begin{pmatrix} d_2 - 1 & N_2 \\ N_2^\top & M \end{pmatrix} + \det M && \text{Induktionsvoraussetzung} \\
&= \det \begin{pmatrix} d_2 & N_2 \\ N_2^\top & M \end{pmatrix} + \det \begin{pmatrix} -1 & 0 \\ N_2^\top & M \end{pmatrix} + \det M && \text{Linearität von det in der 1. Zeile} \\
&= \det \begin{pmatrix} d_2 & N_2 \\ N_2^\top & M \end{pmatrix} && \text{Vereinfachen!} \\
&= \det(D-A)_1 && \text{Nach Definition.}
\end{aligned}$$

Und der Satz von Kirchhoff ist bewiesen! □

Übungen.

34. Finden Sie eine Beschreibung der Anzahl aller Dreiecke eines endlichen Graphen G (ein Dreieck in G ist ein Subgraph von G, der isomorph ist zu K_3) mit Hilfe der Adjazenzmatrix von G.

35. Beschreiben Sie die Graphen, deren Adjazenzmatrix eine der folgenden beiden Gestalten hat:

$$\begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix} \quad \begin{pmatrix} 0 & C^\top \\ C & 0 \end{pmatrix}$$

Hier stehen A, B, C für beliebige $n \times n$ Untermatrizen mit Einträgen aus $\{0, 1\}$ und 0 steht für die $n \times n$ Untermatrix mit allen Einträgen gleich 0. Bei A und B fordern wir zusätzlich, dass jeder Diagonaleneintrag 0 ist.

36. $(*)$ Beweisen Sie die Formel von Cayley (Satz 106) mit Hilfe des Satzes von Kirchhoff 111. **Hinweis:** Berechnen Sie zunächst den Rang von $D - A$ wobei D die Gradmatrix und A die Adjazenzmatrix des K_n.

11 Planare Graphen

Ein *Polygonzug* (oder *Streckenzug*) ist die Vereinigung P von Verbindungsstrecken[59] einer endlichen Folge von Punkten $p_1, \ldots, p_n$; in dieser Vorlesung meist von Punkten aus $\mathbb{R}^2$ (der *Ebene*). Weiterhin wollen wir annehmen, dass der Punkt p_i, für $i \in \{2, \ldots, n-1\}$, nur in der Verbindungsstrecke zwischen p_{i-1} und p_i und der Verbindungsstrecke zwischen p_i und p_{i+1}, nicht aber in den anderen Strecken des Polygonzugs liegen. Alle anderen Punkte des Polygonzuges liegen in nur einer der Verbindungsstrecken des Polygonzugs. Der erste und letzte Punkt in der Folge, p_1 und p_n, heißen *Endpunkte* des Polygonzugs, und das *Innere* des Polygonzugs ist der Polygonzug ohne die Endpunkte, $P \setminus \{p_1, p_n\}$. Wir sagen, dass P die beiden Punkte p_1 und p_2 *verbindet*.

Definition 114. *Ein* ebener Graph *ist ein Paar* (V, E) *aus einer Menge* $V \subseteq \mathbb{R}^2$ *von* Ecken *und einer Menge von Polygonzügen* E*, den* Kanten*, wobei*

1. *die Kanten aus* E *jeweils zwei verschiedene Ecken aus* V *miteinander verbinden,*
2. *verschiedene Kanten verschiedene Endpunkte besitzen,*
3. *und das Innere der Kanten keine Ecken und keine Punkte anderer Kanten enthält.*

Jedem ebenen Graph (V, E) ist ein gewöhnlicher Graph zugeordnet, nämlich der Graph mit Knotenmenge V, der für jeden Polygonzug von a nach b in E eine Kante zwischen a und b hat. In diesem Fall nennen wir (V, E) eine *Zeichnung*, oder *Einbettung*, des Graphen (in die Ebene). Wir können daher alle Begriffe, die wir für Graphen definiert haben, wie zum Beispiel 'Zusammenhang', auch für ebene Graphen anwenden.

Wenn man die zweite Bedingung in der Definition von ebenen Graphen weglässt, und die erste dahingehend abändert, dass eine Kante auch aus einem einzigen Knoten bestehen darf (in welchem Fall sie eine Ecke mit eine Ecke mit sich selbst verbindet), so kann man auf diese Weise auch Multigraphen darstellen, und man spricht dann von einem *ebenen Multigraphen*.

Definition 115. *Ein Graph heißt* planar*, wenn er eine Zeichnung besitzt.*

Beispielsweise ist der Graph K_4 planar, denn er *besitzt* eine Zeichnung (siehe Abbildung 26). Man kann sich auch davon überzeugen, dass der Graph K_5 *nicht* planar ist – darauf werden wir noch zurückkommen.

Bemerkung 116. *Man kann zeigen, dass jeder planare Graph eine Zeichnung besitzt, in der jeder Polygonzug aus einer einzigen Strecke besteht (der Satz von Wagner und Fáry[60]; siehe erneut Abbildung 26).*

[59] Wenn $a, b \in \mathbb{R}^2$, dann ist die *Verbindungsstrecke* von a und b ist die kürzeste Verbindung von a und b, das heißt, die Menge $\{x \in \mathbb{R}^2 \mid x = a + \lambda(b-a), \lambda \in \mathbb{R}, 0 \leq \lambda \leq 1\}$.

[60] István Fáry; geboren am 30. Juni 1922 in Gyula; gestorben am 2. November 1984 in El Cerrito, Kalifornien.

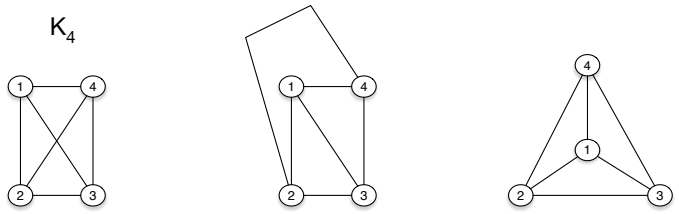


Abbildung 26: Eine Illustration des K_4, die keine Einbettung ist (links), und eine Einbettung des K_4 (mittig), und eine Einbettung, in der alle Polygonzüge aus einzelnen Strecken bestehen (rechts)

.

11.1 Die eulersche Polyederformel

Für eine beliebige Menge $M \subseteq \mathbb{R}^2$ definieren wir folgende Äquivalenzrelation R: ein Paar $(a, b) \in M^2$ sei genau dann in R, wenn es einen Polygonzug in M gibt, der a und b verbindet. Die Äquivalenzklassen von R werden die *Gebiete* von M genannt.

Definition 117. *Es sei (V, E) ein ebener Graph. Dann sind die* Flächen *von (V, E) die Gebiete der Menge $\mathbb{R}^2 \setminus \bigcup_{e \in E} e$.*

Beispielsweise hat ein ebener Baum, oder allgemeiner ein ebener Wald, nur eine Fläche. Der Kreis C_n besitzt ebenfalls eine Zeichnung, und jede Zeichnung von C_n hat genau zwei Flächen. Wir sagen, dass eine Kante $e \in E$ eines ebenen Graphen (V, E) eine Fläche f *begrenzt*, wenn f keine Fläche des ebenen Graphen $(V, E \setminus \{e\})$ mehr ist (da f verschmolzen ist mit der Fläche, die in (V, E) durch die Kante e von f getrennt wurde).

Der erste Satz in diesem Kapitel trägt den Namen *Polyederformel*. Er ist nach einem wichtigen Spezialfall benannt worden, den wir später genauer betrachten; wir werden den Satz aber hier ganz allgemein für ebene Graphen zeigen.

Satz 118 (Eulersche Polyederformel)**.** *Es sei (V, E) ein zusammenhängender ebener Graph mit n Ecken, m Kanten, und l Flächen. Dann gilt*

$$n - m + l = 2 \ .$$

Beweis. Wir halten n fest, und beweisen die Aussage per Induktion nach m. Falls $m \leq n-1$, so zeigt Lemma 60, dass (V, E) ein Baum ist, und $m = n - 1$ gilt. Bäume haben genau eine Fläche, also gilt die Polyederformel, denn $n - (n - 1) + 1 = 2$.

Wir betrachen nun den Fall, dass $m \geq n$ ist. Nach Lemma 60 ist (V, E) nun kein Baum, und muss einen Kreis enthalten. Wähle eine Kante e auf diesem Kreis. Es seien f_1 und f_2 die an diese Kante angrenzenden Flächen; da wir eine Kante auf einem Kreis gewählt haben, sind diese beiden Flächen verschieden[61]. Dann hat der ebene Graph $(V, E \setminus \{e\})$ genau eine

[61] Für einen ganz ausführlichen, strengen Beweis wäre hier weitere Arbeit notwendig.

Kante und eine Fläche weniger als (V, E). Die Polyederformel gilt also für $(V, E \setminus \{e\})$ nach Induktionsvoraussetzung, und damit auch für (V, E). □

Man kann sich die Polyederformel auch wie folgt merken (das Vorzeichen *alterniert*):

$$-1 + n - m + l - 1 = 0$$

Bemerkung 119. *Die eulersche Polyederformel gilt auch für ebene* Multigraphen. *Die Beweisidee ist die gleiche, wir überlassen die Details der Leser:in.*

11.2 Triangulationen

Ein ebener Graph (V, E) heißt eine *Triangulation* (oder *Triangulierung*) wenn jede Fläche von (V, E) (inklusive der äußeren) von einem K_3 begrenzt wird.

Proposition 120. *Es sei (V, E) ein ebener Graph mit mindestens drei Ecken. Dann ist (V, E) genau dann eine Triangulation, wenn (V, E) ein* maximaler ebener Graph *ist, das bedeutet, dass man keine weiteren Kanten zu (V, E) hinzufügen kann.*

Beweis. Wenn jede Fläche bereits von drei Flächen begrenzt wird, so kann man keine Kante hinzufügen: denn jede weitere Kante müsste innerhalb einer Fläche von (V, E) verlaufen, mit den Endpunkten auf dem Rand dieser Fläche; da diese Fläche ein Dreieck ist, waren diese Endpunkte aber schon in (V, E) verbunden, ein Widerspruch zur dritten Forderung in der Definition von ebenen Graphen.

Wenn umgekehrt (V, E) ein maximaler ebener Graph ist, dann muss jede Fläche von drei Kanten begrenzt sein: denn ansonsten grenzt die Fläche an mindestens vier Knoten, von denen zwei noch nicht durch eine Kante verbunden sind, und wir könnten eine Kante zwischen diesen beiden Knoten durch diese Fläche zeichnen. □

Korollar 121. *Ein ebener Graph mit $n \geq 3$ Ecken hat höchstens $3n - 6$ Kanten.*

Beweis. Nach Proposition 120 genügt es zu zeigen, dass Triangulationen maximal $3n - 6$ Kanten haben können. In Triangulationen grenzt jede Fläche an drei Kanten. Jede Kante begrenzt genau zwei verschiedene Flächen. Wenn f die Anzahl der Flächen von (V, E) bezeichnet, und m die Anzahl an Kanten, dann folgt daraus, dass $2m = 3f$ gilt. Wir ersetzen also f in der eulerschen Polyederformel durch $2m/3$, und erhalten $n - m + 2m/3 = 2$, was äquivalent ist zu $m = 3n - 6$. □

Diese Formel kann man dazu verwenden, um zu zeigen, dass manche Graphen nicht planar sind. Der Graph K_5, zum Beispiel, hat $10 > 3 \cdot 5 - 6 = 9$ Kanten, kann also nach Korollar 121 keine Zeichnung besitzen.

Übungen.

37. Geben Sie einen informellen Beweis dafür, dass der $K_{3,3}$ aus Beispiel 28 nicht planar ist (bzw. geben Sie einen formalen Beweis, in dem Sie die geometrische Aussage nutzen dürfen, die wir auch schon im Beweis der Eulerschen Polyederformel verwendet haben). Hinweis: der $K_{3,3}$ enthält den C_6 als Subgraphen.

38. Geben Sie einen formalen Beweis dafür, dass der $K_{3,3}$ nicht planar ist unter Verwendung der Eulersche Polyederformel. Tipp: wie viele Flächen müsste eine Zeichnung des $K_{3,3}$ haben? An wie viele Kanten grenzt eine Fläche dieser Zeichnung mindestens?

39. Zeigen Sie, dass eine Triangulation mit mindestens vier Knoten dreifach zusammenhängend ist.

40. Zeigen Sie, dass ein planarer Graph nicht 6-fach zusammenhängend sein kann.

41. Finden Sie einen 5-fach zusammenhängenden planaren Graphen.

42. Zeigen Sie, dass jeder Graph eine kreuzungsfreie Zeichnung im $\mathbb{R}^3$ besitzt.

43. Verallgemeinern Sie den Begriff der Zeichnung für $\mathbb{R}^3$ statt $\mathbb{R}^2$. Zeigen Sie, dass jeder endliche Graph eine Zeichnung im $\mathbb{R}^3$ besitzt. Gilt das auch für unendliche Graphen?

11.3 Zeichnungen

Sei (V, E) eine Zeichnung. Wenn V endlich ist, so ist die Menge $\bigcup_{e \in E} e$ *beschränkt*. Das bedeutet, dass es ein $s \in \mathbb{R}$ gibt, so dass $M \subseteq \{(x_1, x_2) \in \mathbb{R}^2 \mid -s < x_1, \ldots, x_n < s\}$. Man sieht leicht, dass es genau eine Fläche von (V, E) gibt, die unbeschränkt ist. Diese Fläche nennen wir die *äußere Fläche* von (V, E). In Situationen, in denen alle Flächen gleichberechtigt sein sollen, kann man anstelle von Einbettungen in $\mathbb{R}^2$ auch Einbettungen auf die *Einheitssphäre* S^2 betrachten, die wie folgt definiert ist:

$$S^2 := \left\{p \in \mathbb{R}^3 \;\middle|\; |p| = 1\right\} = \left\{(x, y, z) \in \mathbb{R}^3 \mid x^2 + y^2 + z^2 = 1\right\} .$$

In anderen Worten, S^2 besteht aus allen Punkten im Raum mit Abstand 1 zum *Ursprung*, dem Punkt $(0, 0, 0)$. Polygonzüge auf S^2 sind Streckenzüge mit verallgemeinerten Strecken, bei denen wir zwei Punkte nicht mit einer Geraden, sondern mit der kürzesten Verbindung in S^2 verbinden. Entsprechend kann man dann auch Einbettungen in S^2 analog zu Einbettungen in $\mathbb{R}^2$ definieren.

Ein Graph hat genau dann eine Zeichnung in der Ebene, wenn er eine Zeichnung auf S^2 besitzt. Dazu bemerken wir, dass wenn wir einen Punkt aus S^2 entfernen, die resultierende Menge topologisch so aussieht wie $\mathbb{R}^2$. Der Begriff, der diese Intuition formal fasst, ist der Begriff des *Homöomorphismus* (auch *Homeomorphismus*): dies sind Bijektionen φ, die in beiden Richtungen stetig sind. Wir wollen und können diese Begriffe in dieser Vorlesung

nicht formal definieren, sondern betrachten stattdessen ein Beispiel für einen solchen Homöomorphismus π. Der Einfachheit halber nehmen wir an, dass wir den *Nordpol* $(0,0,1)$ aus S^2 entfernen. Für jeden Punkt p aus $S^2 \setminus \{(0,0,1)\}$ gibt es genau eine Linie durch $(0,0,1)$ und p, und diese Linie schneidet die Ebene, die gegeben ist durch $\{(x,y,z) \in \mathbb{R}^3 \mid z = 0\}$, in genau einem Punkt q. Wir definieren $\pi(p) := q$. Die Abbildung $\pi\colon S^2 \setminus \{(0,0,1)\} \to \mathbb{R}^2$ ist eine Bijektion, und hat tatsächlich die gewünschte Eigenschaft, dass Einbettungen eines Graphen G in $\mathbb{R}^2$ auf Einbettungen von G in S^2 abgebildet werden, und umgekehrt werden Einbettungen von G in S^2, die den Punkt $(0,0,1)$ nicht berühren, durch π^{-1} abgebildet auf Einbettungen von G in $\mathbb{R}^2$.

Zwischen Zeichnungen in der Ebene und Zeichnungen auf der Sphäre gibt es jedoch den Unterschied, dass bei Zeichnungen auf S^2 keine Fläche als äußere Fläche ausgezeichnet wird.

Topologisch isomorphe Zeichnungen

Es gibt unendlich viele Zeichnungen eines Graphen, aber manche Zeichnungen sehen anderen sehr ähnlich, und das wollen wir in diesem Kapitel und dem nächsten genauer betrachten. Unser erster Ähnlichkeitsbegriff ist ein sehr intuitiver: wir würden zwei ebene Graphen (V,E) und (V',E') gerne *topologisch isomorph* nennen, wenn es einen Homöomorphismus φ gibt, der V auf V' und E auf E' abbildet. Allerdings muss dann die äußere Fläche von (V,E) auf die äußere Fläche von (V',E') abgebildet werden, und diese Einschränkung wünscht man sich im allgemeinen nicht. Wir gehen daher in der folgenden Definition einen Umweg über Einbettungen in S^2.

Definition 122. *Es seien (V,E) und (V',E') zwei ebene Graphen. Eine Abbildung*

$$\sigma\colon V \cup E \to V' \cup E'$$

heißt topologischer Isomorphismus *falls $\sigma(V) = V'$, $\sigma(E) = E'$, und es einen Homöomorphismus φ zwischen S^2 und S^2 gibt so dass σ die Einschränkung von $\pi \circ \varphi \circ \pi^{-1}$ auf $V \cup E$ ist.*

In Abbildung 27 finden sich drei Zeichnungen, von denen die ersten beiden topologisch isomorph sind, nicht aber topologisch isomorph sind zur letzten Zeichnung.

Kombinatorisch isomorphe Zeichnungen

Seien (V_1,E_1) und (V_2,E_2) ebene Graphen und sei F_1 die Menge der Flächen von (V_1,E_1) und F_2 die Menge der Flächen von (V_2,E_2). Ein *kombinatorischer Isomorphismus* zwischen (V_1,E_1) und (V_2,E_2) ist eine Bijektion zwischen $V_1 \cup E_1$ und $V_2 \cup E_2$, die sich erweitern lässt zu einer Bijektion $\alpha\colon V_1 \cup E_1 \cup F_1 \to V_2 \cup E_2 \cup F_2$ so dass

- $\alpha(V_1) = V_2$, $\alpha(E_1) = E_2$, und $\alpha(F_1) = F_2$,

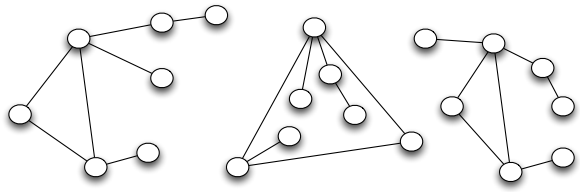


Abbildung 27: Die ersten beiden Zeichnungen sind topologisch isomorph, die letzte Zeichnung nicht.

- $a, b \in V_1$ genau dann durch eine Kante $e \in E_1$ verbunden sind, wenn $\alpha(a)$ und $\alpha(b)$ durch $\alpha(e)$ verbunden sind, und
- eine Ecke oder Kante x aus (V_1, E_1) genau dann auf dem Rand einer Fläche $f \in F_1$ liegt wenn $\alpha(x)$ auf dem Rand von $\alpha(f)$ liegt.

Falls es einen kombinatorischen Isomorphismus zwischen (V_1, E_1) und (V_2, E_2) gibt, so heißen (V_1, E_1) und (V_2, E_2) *kombinatorisch isomorph.* Ein Beispiel von zwei Zeichnungen desselben Graphen, die kombinatorisch nicht isomorph sind, findet sich in Abbildung 28. Auf der anderen Seite sind alle drei Zeichnungen in Abbildung 27 kombinatorisch isomorph. Topologisch isomorphe Zeichnungen sind auch kombinatorisch isomorph; umgekehrt stimmt das nicht, wie Abbildung 27 belegt.

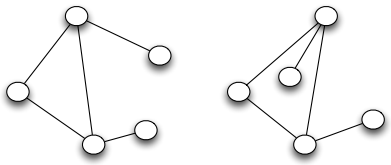


Abbildung 28: Zwei Zeichnungen eines Graphen, die kombinatorisch nicht isomorph sind.

Satz 123. *Jeder kombinatorische Isomorphismus zwischen zweifach zusammenhängenden ebenen Graphen ist bereits ein topologischer Isomorphismus.*

Satz 124 (Whitney[62])**.** *Alle Zeichnungen eines dreifach zusammenhängenden Graphen sind kombinatorisch isomorph.*

Da ein dreifach zusammenhängender Graph insbesondere zweifach zusammenhängend ist, so folgt aus Satz 124 in Kombination mit Satz 123, dass Zeichnungen von dreifach zusammenhängenden Graphen sogar topologisch isomorph sein müssen.

[62] Hassler Whitney; geboren am 23. März 1907 in New York City; gestorben am 10. Mai 1989 in Princeton.

11.4 Polyeder

Polyeder haben in der Mathematik keine einheitliche Definition. Zum Zwecke dieser Vorlesung soll ein Polyeder eine Menge $P \subseteq \mathbb{R}^n$ sein, die durch eine endliche Menge linearer Ungleichungen definiert wird; formal, $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ wobei $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$. Mit dieser Definition sind Polyeder konvexe Mengen: eine Menge $M \subseteq \mathbb{R}^n$ heißt *konvex* wenn für alle $p, q \in M$ auch die Verbindungsstrecke zwischen p und q in M ist.

Wir betrachten nun einen beschränkten Polyeder $P \subseteq \mathbb{R}^3$. Der Rand eines solchen Polyeders besteht aus *Ecken*, *Kanten*, und *Flächen*. Die Ecken und Kanten bilden einen Graphen, den wir im Folgenden den *Kantengraphen von P* nennen wollen.[63] Diese Kantengraphen besitzen im Sinne von Abschnitt 11.3 eine Einbettung in die Sphäre S^2. Ein Polyeder $P \subseteq \mathbb{R}^3$ ist *3-dimensional* wenn er für ein $\epsilon > 0$ einen Würfel mit Seitenlänge ϵ enthält. Ein Polyeder $P \subseteq \mathbb{R}^3$ ist *beschränkt* wenn es ein $\in \mathbb{R}$ gibt so dass

$$P \subseteq \{(x_1, x_2, x_3) \in \mathbb{R}^3 \mid -s < x_1, \ldots, x_3 < s\}.$$

Satz 125 (Steinitz[64])**.** *Ein Graph tritt genau dann als Kantengraph eines beschränkten 3-dimensionalen Polyeders $P \subseteq \mathbb{R}^3$ auf, wenn er 3-fach zusammenhängend und planar ist.*

11.5 Der Dualgraph

Es seien $G = (V, E)$ und $G^* = (V^*, E^*)$ zwei ebene Multigraphen mit jeweils den Flächenmengen F und F^*. Dann nennt man G^* *dual* zu G falls es Bijektionen $\alpha\colon F \to V^*$, $\beta\colon E \to E^*$, und $\gamma\colon V \to F^*$ gibt, so dass

1. für alle $f \in F$ gilt $\alpha(f) \in f$;
2. für alle $e \in E$ gilt $|\beta(e) \cap \bigcup_{e \in E} e| = |e \cap \beta(e)| = |e \cap \bigcup_{e \in E^*} e| = 1$;
3. für alle $v \in V$ gilt $v \in \gamma(v)$.

Proposition 126. *Jeder zusammenhängende ebene Multigraph besitzt einen dualen.*

Beweisidee. Sei (V, E) ein ebener, zusammenhängender Multigraph. Wir wählen zunächst aus jeder Fläche f von (V, E) einen Punkt v aus, und setzen $\alpha(f) = v$. Das Bild von α wird die Knotenmenge V^* vom Dualgraphen. Wenn zwei Flächen f_1, f_2 von (V, E) von der gleichen Kante e begrenzt werden, so verbinden wir $\alpha(f_1)$ und $\alpha(f_2)$ durch eine Kante e^* (was sicher möglich ist) und setzen $\beta(e) := e^*$. Das Bild von β wird die Kantenmenge E^* vom Dualgraphen. Es bleibt nur die Verifikation, dass jede Fläche f^* von (V^*, E^*) genau einen Knoten v aus V enthält (hier verwenden wir, dass (V, E) zusammenhängend ist). Wir setzen $\gamma(v) = f^*$. □

[63] Der Kantengraph eines Polyeders ist nicht zu verwechseln mit dem Kantengraphen eines Graphen aus Abschnitt 7.6.

[64] Ernst Steinitz; geboren am 13. Juni 1871 in Laurahütte, Oberschlesien; gestorben am 29. September 1928 in Kiel.

Proposition 127. *Seien G_1 und G_2 topologisch isomorphe Multigraphen mit Dualgraphen G_1^* und G_2^*. Dann sind G_1^* und G_2^* topologisch isomorph.*

Proposition 128. *Wenn G^* ein Dualgraph von G ist, dann ist G ein Dualgraph von G^*.*

Wegen Proposition 127 ergibt es Sinn, von *dem* Dualgraphen G^* von G zu sprechen. Nach Proposition 128 gilt dann für jeden ebenen zusammenhängenden Multigraphen G, dass $(G^*)^*$ topologisch isomorph ist zu G.

Korollar 129. *Sei $n \geq 4$. Dann gibt es gleich viele Triangulationen mit n Knoten, wie es 3-zusammenhängnde 3-reguläre planare Graphen mit $2n-4$ Knoten gibt.*

Beweis. Man kann zeigen, dass eine Triangulation mit mindestens 4 Knoten 3-fach zusammenhängend ist (Übung 39). Nach dem Satz von Whitney (Satz 124) sind also alle Zeichnungen G des Graphen kombinatorisch isomorph. Der Beweis von Korollar 121 zeigt, dass eine Triangulation mit n Knoten *genau* $3n-6$ Kanten besitzt. Nach der eulerschen Polyederformel (Satz 118) gilt $n-(3n-6)+f=2$, also $f=2n-4$. Da G zusammenhängend ist, gibt es einen Dualgraphen G^* (Proposition 126), und dieser ist eindeutig bis auf topologische Isomorphie (Proposition 127). Dieser hat $2n-4$ Knoten, ist klarerweise 3-regulär. Ausserdem kann man zeigen, dass G^* ebenfalls 3-fach zusammenhängend ist und keine Mehrfachkanten oder Schleifen enthält. Nach Proposition 128 ist also die Abbildung $G \mapsto G^*$ eine Bijektion zwischen Triangulationen mit n Knoten und 3-regulären 3-fach zusammenhängenden planaren Graphen mit $2n-4$ Knoten, was die Aussage impliziert. □

Übungen.

44. Zeigen Sie: wenn zwei ebene Graphen kombinatorisch isomorph sind, dann sind auch deren Dualgraphen kombinatorisch isomorph.

11.6 Minoren und der Satz von Kuratowski-Wagner

Es seien G und H endliche Graphen. Wenn H aus G durch Löschen von Kanten und Knoten, Kontrahieren von Kanten (Definition 68), und Umbenennen der Knoten gewonnen werden kann, so nennt man H einen *Minor* von G, und schreibt $H \preceq G$.

Proposition 130. *Wenn G ein planarer Graph ist, und H ein Minor von G, dann ist auch H planar.*

Im folgenden schreiben wir $\mathcal{G}$ für die Menge aller Graphen mit Knotenmenge der Gestalt $\{1,\ldots,n\}$ für ein $n \in \mathbb{N}$. Die Graphminorenrelation ist dann eine Relation auf $\mathcal{G}$, sprich, eine Teilmenge von $\mathcal{G}^2$. Sicherlich ist $\preceq$ transitiv und reflexiv; das bedeutet, die Minorenrelation ist eine Quasiordnung von $\mathcal{G}$. (Und selbstverständlich gewinnen wir über Faktorisierung auf natürliche Weise eine zugehörige partielle Ordnung, siehe Abschnitt 9.2.)

Satz 131 (Minorentheorem; Robertson[65] und Seymour[66])**.** *In jeder unendlichen Menge*

[65] Neil Robertson; geboren am 30. November 1938 in Kanada.
[66] Paul D. Seymour; geboren am 26. Juli 1950 in Plymouth.

von endlichen Graphen gibt es zwei Graphen G und H so dass $G \preceq H$; damit ist die Minorenrelation auf $\mathcal{G}$ eine Wohlquasiordnung.

Der zweite Teil der Aussage von Satz 131 folgt aus dem ersten mit Hilfe von Lemma 98. Der Beweis des ersten Teils des Satzes ist allerdings sehr kompliziert, und kann in dieser Vorlesung sicher nicht gezeigt werden.

Korollar 132. *Es gibt eine endliche Menge von Graphen $\mathcal{F}$, so dass ein endlicher Graph genau dann planar ist, wenn er keinen der Graphen aus $\mathcal{F}$ als Minoren hat.*

Beweis. Die Menge $\mathcal{P}$ der planaren Graphen in $\mathcal{G}$ ist nach Proposition 130 abgeschlossen unter Minoren. Für $\mathcal{N} := \mathcal{G} \setminus \mathcal{P}$ gilt in der Terminologie aus Abschnitt 9.5 also $\uparrow\mathcal{N} = \mathcal{N}$. Da die Graphminorenrelation nach dem vorhergehenden Satz eine Wohlquasiordnung ist, so gibt uns Proposition 100 eine endliche Menge $\mathcal{F}$ von Graphen mit $\uparrow\mathcal{F} = \mathcal{N}$. $\square$

Satz 133 (Robertson-Seymour Nummer 13)**.** *Es sei H ein Graph. Dann gibt es einen Algorithmus, der für einen gegebenen endlichen Graphen G in polynomieller Zeit feststellt, ob H ein Minor von G ist.*

Korollar 134. *Es gibt einen polynomiellen Algorithmus, der von einem gegebenen Graphen feststellt, ob er planar ist.*

Man weiß nicht nur, dass Planarität mit endlich vielen verbotenen Minoren charakterisiert werden kann, sondern man kennt sogar die Minoren einer solchen Charakterisierung.

Satz 135 (Kuratowski[67], Wagner[68])**.** *Ein endlicher Graphen G ist genau dann planar, wenn er weder K_5 noch $K_{3,3}$ als Minoren enthält.*

Dass K_5 nicht planar ist, folgt bereits aus Korollar 121. Dass $K_{3,3}$ nicht planar ist, ist der Gegenstand in Übung 37 und Übung 38. Die interessante Richtung im Satz 135 ist, dass *jeder* nicht-planare Graph einen dieser beiden Graphen als Minor enthalten muss.

Übungen.

45. Zeigen Sie: der Petersengraph (siehe Abbildung 29) ist nicht planar.

46. Finden Sie für jedes $n \in \mathbb{N}$ eine Antikette der Größe n bezüglich der Minorrelation auf $\mathcal{G}$.

47. Zeigen Sie, dass es in $\mathcal{G}$ unendliche Antiketten bezüglich der *Subgraphrelation* gibt (das heißt, für $G, H \in \mathcal{G}$ definieren wir $H \leq G$ falls H ein Subgraph ist von G).

[67] Kazimierz Kuratowski; geboren am 2. Februar 1896 in Warschau, Polen; gestorben am 18. Juni 1980 in Warschau.

[68] Klaus Wagner; geboren am 31. März 1910 in Köln-Klettenberg; gestorben am 6. Februar 2000.

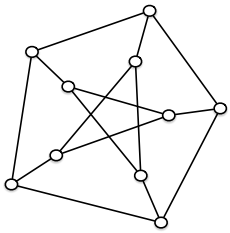


Abbildung 29: Der Petersengraph (siehe Übung 45).

48. Ein Graph heißt *kreisartig planar* (oder *außenplanar* von englisch *outerplanar*) wenn er eine Zeichnung in der Ebene besitzt, in der jeder Knoten an die äußere Fläche grenzt.

    - Zeigen Sie: wenn G kreisartig planar ist, und H ist ein Minor von G, dann ist auch H kreisartig planar.
    - Verwenden Sie einen Satz der Vorlesung, um zu zeigen, dass es endlich viele Graphen $G_1, \ldots, G_k$ gibt, so dass ein Graph genau dann kreisartig planar ist, wenn er keinen dieser Graphen $G_1, \ldots, G_k$ als Minor enthält.
    - Zeigen Sie: der K_4 und der $K_{2,3}$ (siehe Beispiel 28) sind nicht kreisartig planar.
    - Zeigen Sie: ein Graph ist genau dann kreisartig planar, wenn er weder K_4 noch $K_{2,3}$ als Minor besitzt.
    - Zeigen Sie: Der Dualgraph eines kreisartig planar Graphen ohne die Knoten für die äußere Fläche ist ein Baum.
    - Wie viele *maximale* (im Sinne von Proposition 120) kreisartig planar Graphen mit den Knoten $\{1, 2, 3, \ldots, n\}$ gibt es?
    - Zeigen Sie: jeder kreisartig planare Graph ist 3-färbbar.
    - Zeigen Sie, dass jeder zweifach zusammenhängende kreisartig planare Graph einen Hamiltonkreis besitzt.

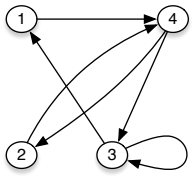


Abbildung 30: Eine Illustration eines Graphen (V, E) mit der Knotenmenge $V = \{1, 2, 3, 4\}$.

12 Gerichtete Graphen

Ein *gerichteter Graph* ist ein Paar $G = (V, E)$ wobei

- V eine beliebige Menge ist, deren Elemente wir *Knoten* nennen, und
- $E \subseteq V^2$ eine Menge von Knotenpaaren ist, die wir *(gerichtete) Kanten* nennen.

Mit anderen Worten, ein gerichteter Graph ist eine Menge zusammen mit einer binären Relation auf dieser Menge. Eine Illustration eines Beispiels für einen gerichteten Graphen mit vier Knoten findet sich in Abbildung 30. Man sieht in diesem Beispiel, dass Kanten der Gestalt (v, v) zugelassen sind. Ausserdem kann es verschiedene Knoten $u, v \in V$ geben, so dass E sowohl (u, v) als auch (v, u) enthält; in anderen Worten, die Kantenrelation muss nicht antisymmetirsch sein. Graphen mit der Eigenschaft, dass die Kantenrelation antisymmetrisch ist, nennt man *orientierte Graphen*.

Ist (v, w) eine Kante im gerichteten Graphen (V, E), so nennen wir w einen *Nachfolger* von v und v einen *Vorgänger* von w. Ein Knoten $v \in V$ ist eine *Quelle*, wenn er keinen Vorgänger hat, und eine *Senke*, wenn er keinen Nachfolger hat. Ein *Kantenzug* ist eine Folge $v_1, v_2, \ldots, v_n$ so dass $(v_i, v_i + 1) \in E$ für alle $i \in \{1, \ldots, n-1\}$. Wir sprechen in diesem Fall auch von einem *Kantenzug von* v_1 *nach* v_n (in (V, E)), und dass v_n von v_1 aus (in (V, E)) *erreichbar* ist, und schreiben $v_1 \preceq v_n$. Der Fall $n = 1$ sei hier explizit erlaubt; in diesem Fall hat der Kantenzug nur einen Knoten und keine Kante. Ein *(gerichteter) Pfad* ist ein Kantenzug mit paarweise verschiedenen Knoten.

Ein *geschlossener gerichteter Kantenzug* ist eine Folge $v_0, \ldots, v_{n-1}$ von $n \geq 1$ Knoten aus V mit $(v_i, v_{i+1}) \in E$ für alle $i \in \mathbb{Z}_n$; auch hier ist der Fall $n = 1$ erlaubt (und der Kantenzug hat genau eine Kante, nämlich (v_0, v_0)). Ein *(gerichteter) Kreis* ist ein geschlossener Kantenzug mit paarweise verschiedenen Knoten. Im Gegensatz zu ungerichteten Graphen kann es also in gerichteten Graphen Kreise mit zwei Kanten und Kreise mit einer Kante geben.

Ein gerichteter Graph G ist ein *Wald*, wenn jeder Knoten in G höchstens einen Vorgänger hat und G keine gerichteten Kreise enthält. Wälder mit nur einer Quelle heißen *Bäume* (und die Quelle heißt die *Wurzel* des Baumes). Wälder sind also disjunkte Vereinigungen von Bäumen.

```
Tiefensuche(G, v).
Eingabe: G = (V, E), v ∈ V.
Ausgabe: Markiere alle von v aus in G erreichbaren Knoten.
---
Markiere v.
Für alle Kanten von v zu einem w in G:
        Falls w noch nicht markiert
                Tiefensuche(G, w).
```

Abbildung 31: Tiefensuche.

Wir definieren die *Erreichbarkeitsrelation* $\preceq$ in (V, E) wie folgt: $x \preceq y$ gelte genau dann, wenn es in (V, E) einen gerichteten Pfad von x nach y gibt. In der Terminologie von Abschnitt 9.3 ist also $\preceq$ der transitive Abschluss der Kantenrelation E.

Proposition 136. *Die Erreichbarkeitsrelation $\preceq$ in einem Graphen G ist eine Quasiordnung. Falls G kreisfrei ist, so ist $\preceq$ eine partielle Ordnung. In Bäumen ist $\preceq$ eine semilineare Ordnung.*

12.1 Tiefensuche

Tiefensuche ist ein Verfahren, um einen gegebenen ungerichteten oder gerichteten Graphen zu durchsuchen. Die Suche kann bei einem beliebigen Knoten v des Graphen begonnen werden. Das besondere bei der Tiefensuche ist, dass für jeden betretenen Knoten zuerst die von diesem Knoten aus erreichbaren Knoten so weit wie möglich erkundet werden, bevor man wieder zurück geht ('backtracking'). Das Verfahren ist in Pseudocode in Abbildung 31 dargestellt. Im Verfahren wird jeder Knoten, der betreten wird, *markiert*. Auf diese Weise weiß man am Ende des Verfahrens, welche Knoten von v aus erreichbar sind, nämlich genau die markierten. Für manche Anwendungen der Tiefensuche werden wir später noch mehr Informationen in den Markierungen abspeichern.

Diese simple Idee ist ausgesprochen mächtig und nützlich. Wir werden von einfachen Dingen, die man mit Tiefensuche bewerkstelligen kann, zu immer trickreicheren Anwendungen der Tiefensuche kommen.

Jede Tiefensuche definiert gewisse lineare Ordnungen auf der Menge der Knoten V. Dazu starten wir die Tiefensuche in einem beliebigen Knoten. Wann immer die Tiefensuche abbricht, aber noch nicht alle Knoten markiert wurden, starten wir die Tiefensuche in einem noch unmarkierten Knoten neu, solange bis alle Knoten des Graphen markiert sind. Nun betrachten wir die folgenden linearen Ordnungen auf V:

- Die *Knotenpräordnung.* Die Ordnung, in der die Knoten *zum ersten Mal* betrachtet werden. Wir nummerieren die Knotenmenge N bezüglich dieser Ordnung mit 1 bis

$n = |V|$ durch, und schreiben $s(v)$ für die Nummer von v in dieser Nummerierung. Die Zahl $s(v) \in \{1, \ldots, n\}$ heißt auch die *Startzeit* von v.

- Die *Knotenpostordnung.* Die Ordnung, in der die Knoten *das letzte Mal* betrachtet werden (für einen Knoten v ist das der Zeitpunkt, zu dem alle rekursiven Aufrufe der Tiefensuche für die Nachfahren von v bereits terminiert wurden, also ganz am Ende von Tiefensuche(G, v)). Erneut nummerieren wir die Knoten gemäß dieser Ordnung, und schreiben $f(v)$ für die Nummer von v in dieser zweiten Nummerierung. Die Zahl $f(v)$ heißt die *Schlusszeit* von v.

Diese Ordnungen sind für einen gegebenen Graphen G selbstverständlich nicht eindeutig, sondern hängen vom Verlauf der Tiefensuche auf G ab. Das oben beschriebene Verfahren (Tiefensuche mit Neustarts bis alle Knoten markiert sind) werden wir *vollständige Tiefensuche* nennen. In der vollständigen Tiefensuche markieren wir jeden Knoten sowohl mit $s(v)$ als auch mit $f(v)$.

Anhand einer vollständigen Tiefensuche kann jede Kante $(u, v) \in E$ des Graphen in **vier** verschiedene Typen eingeteilt werden (siehe auch Abbildung 33):

1. **Baumkanten**. Dies sind die Kanten, entlang derer die Tiefensuche von einem markierten Knoten u zu einem neuen unmarkierten Knoten v schreitet. Wir bemerken, dass der Graph mit Knotenmenge V und genau den Baumkanten einen Wald bildet, den *Tiefensuchwald*.

2. **Vorwärtskanten**. Sind die Kanten, die nicht Baumkanten sind, und die einen Knoten des Baumes mit einem seiner Nachfahren im Tiefensuchwald verbindet.

3. **Rückwärtskanten**. Sind die Kanten, die von einem Knoten zu einem seiner Vorfahren im Tiefensuchwald zeigen.

4. **Querkanten**. Alle übrigen Kanten. Diese können zwischen verschiedenen Bäumen des Tiefensuchwaldes verlaufen, oder im gleichen Baum zwischen Knoten, die unvergleichbar sind bezüglich der Erreichbarkeitsrelation im Tiefensuchwald.

Selbstverständlich ist der Tiefensuchwald im allgemeinen nicht eindeutig, sondern hängt vom Verlauf der Tiefensuche ab. Im Abbildung 32 ist ein Beispiel eines gerichteten Graphen und einer vollständigen Tiefensuche zu finden, im dem alle vier Kantentypen auftreten.

Mit Hilfe dieser Information können wir zum Beispiel einfach feststellen, ob ein gegebener Graph einen Kreis enthält oder kreisfrei ist: ein Graph hat genau dann einen Kreis, wenn er eine Rückwärtskante enthält.

12.2 Starke Zusammenhangskomponenten

Sei (V, E) ein gerichteter Graph. Wir definieren $x \sim y$ falls es in (V, E) einen gerichteten Pfad von x nach y und einen von y nach x gibt. In der obigen Schreibweise gilt also $x \preceq y$ und $y \preceq x$. Die Relation $\sim$ ist eine Äquivalenzrelation auf V (siehe Abschnitt 9.2).

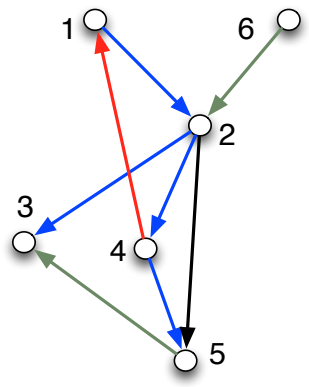


Abbildung 32: Illustration einer vollständigen Tiefensuche und der verschiedenen Kantentypen; die Knoten sind gemäß der Präordnung durchnummeriert.

Definition 137. *Sei G ein gerichteter Graph. Eine* starke Zusammenhangskomponente (SZ-Komponente) *von G ist eine Äquivalenzklasse der oben definierten Relation $\sim$. Der Graph heißt* stark zusammenhängend*, wenn $\sim$ nur eine Äquivalenzklasse hat.*

Das bedeutet, G ist genau dann stark zusammenhängend, wenn es für alle $a, b \in V$ einen gerichteten Pfad von a nach b gibt. Entsprechend sind die SZ-Komponenten eines Graphen genau die maximalen stark zusammenhängenden Subgraphen von G. Ein paar einfache Beobachtungen.

- Wenn $G = (V, E)$ ein gerichteter Graph ist, dann hat der *invertierte* Graph

  $$G^{-1} := (V, \{(v, u) \mid (u, v) \in E\})$$

  die gleichen SZ-Komponenten wie G.

- Jede SZ-Komponente muss enthalten sein in nur einem Baum der Tiefensuche im Tiefensuchwald (denn Kanten aus E, die zwischen verschiedenen Bäumen im Tiefensuchwald verlaufen, gehen immer in die gleiche Richtung; siehe wieder Abbildung 33).

Wie kann man die SZ-Komponenten von G berechnen? Man wünscht sich einen Algorithmus, der die Knoten so mit Zahlen beschriftet, dass zwei Knoten genau dann die gleiche Zahl tragen, wenn sie in der gleichen SZ-Komponente sind.

Wir wollen im folgenden einen einfachen *und* effizienten Algorithmus angeben, der die starken Zusammenhangskomponenten mit einer Laufzeit berechnet, die bloß *linear* ist in der Größe des Graphen (definiert als die Anzahl der Knoten plus die Anzahl der Kanten). Überraschenderweise kommen wir hierfür mit insgesamt nur zwei Tiefensuchen aus. Das

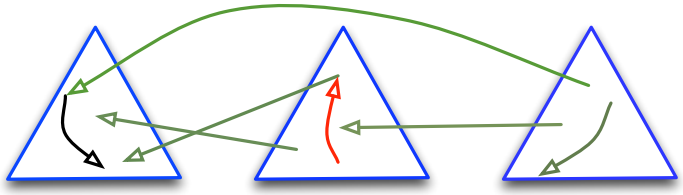


Abbildung 33: Illustration einer vollständigen Tiefensuche und der verschiedenen Kantentypen.

Verfahren ist in Abbildung 34 angegeben, und stammt aus [4]. Die Idee vom eleganten Korrektheitsbeweis dagegen ist von Ingo Wegener[69].

Eingabe: $G = (V, E)$.

0: Lege eine Zählervariable z mit Initialwert 1 an.
1: Führe eine vollständige Tiefensuche auf G durch.
2: Es sei v der Knoten mit der größten Schlusszeit $f(v)$.
3: Führe eine Tiefensuche auf G^{-1} mit Startknoten v aus.
    Alle noch unbeschrifteten Knoten, die von v aus erreichbar sind, liegen in der gleichen SZ-Komponente wie v, und werden mit z beschriftet. Weise z den Wert $z + 1$ zu.
4: Falls alle Knoten von der zweiten Tiefensuche beschriftet wurden: fertig.
5: Ansonsten sei v derjenige in der zweiten Suche unmarkierte Knoten mit der größten Schlusszeit bezüglich der ersten Suche.
6: Fahre mit Schritt 3 fort.

Abbildung 34: Berechnung der starken Zusammenhangskomponenten.

Korrektheitsbeweis für den Algorithmus. Es wird gezeigt, dass der Algorithmus aus Abbildung 34 genau dann zwei Knoten mit der gleichen Zahl beschriftet, wenn sie in der selben SZ-Komponente liegen. Wir führen dazu einen Induktionsbeweis über die Anzahl n der starken Zusammenhangskomponenten. Falls G stark zusammenhängend ist, das heißt, falls $n = 1$, dann sind alle Knoten im gleichen Baum der zweiten Tiefensuche, und die Antwort des Algorithmus ist korrekt.

[69] Ingo Wegener; geboren am 4. Dezember 1950 in Bremen; gestorben am 26. November 2008 in Bielefeld.

Ansonsten betrachten wir den ersten Baum B der zweiten Tiefensuche. Die Wurzel dieses Baumes r ist die Wurzel des letzten Baumes L der ersten Tiefensuche. Da es Pfade gibt von r zu allen Knoten von L, liegt ein Knoten u aus L genau dann in der gleichen SZ-Komponente wie r, wenn es einen Pfad von u zu r gibt. Dies ist genau dann der Fall, wenn es einen Pfad von r zu u in G^{-1} gibt, was wiederum genau dann der Fall ist, wenn u in B liegt. Knoten $u \in V \setminus L$ können von r aus in G^{-1} nicht erreicht werden, und können auch nicht zur gleichen SZ-Komponente gehören wie r.

Wir betrachten nun den Graphen $G - B$; dieser Graph hat eine SZ-Komponente weniger als G. Die Knoten von B sind die Knoten, die von der ersten Tiefensuche zuletzt besucht wurden. Also ist die Einschränkung dieser Tiefensuche auf $G - B$ eine Tiefensuche auf $G - B$. Die Ausgabe des Algorithmus stimmt daher auf den Knoten $V \setminus B$ mit der Ausgabe des Algorithmus überein, der auf $G - B$ ausgeführt wird. Nach Induktionsvoraussetzung aber ist der Algorithmus korrekt auf $G - B$. □

Die Nummerierung der Knoten, die vom Algorithmus aus Abbildung 34 für einen gegebenen gerichteten Graphen G berechnet wird, hat eine weitere interessante Eigenschaft: wenn es einen Pfad von u nach v in G gibt (also $u \preceq v$), dann ist die Nummer von u höchstens so groß wie die für v. Eine solche Nummerierung nennen wir auch eine *topologische Sortierung*.

Beispiel 29. *Wir betrachten den gerichteten Graph G mit der Knotenmenge $\{1, \ldots, 6\}$, der in Abbildung 32 dargestellt ist. Wir nehmen an, dass die Tiefensuche die Knoten in der Reihenfolge $1, 2, 3, 4, 5, 6$ zum ersten Mal besucht; wir haben dann die folgenden Schlusszeiten:*

v	1	2	3	4	5	6
$f(v)$	5	4	1	3	2	6

Die zweite Tiefensuche auf G^{-1} starten wir also auf dem Knoten 6, da der Knoten 6 mit $f(6) = 6$ die größte Schlusszeit besitzt. In G^{-1} ist nur der Knoten 6 selbst von 6 aus erreichbar; nur dieser erhält also die Beschriftung '1' für die erste SCC. Die zweite Tiefensuche auf G^{-1} wählt nun den Knoten 1 als neuen Startpunkt, da dieser mit $f(1) = 5$ die größte Schlusszeit unter den noch unbeschrifteten Knoten besitzt. Von 1 aus sind in G^{-1} lediglich die unmarkierten Knoten 1, 4, und 2 erreichbar; diese erhalten also die Beschriftung '2' für die zweite SCC. Es verbleiben nunmehr die unbeschrifteten Knoten 3 und 5. Die nächste Tiefensuche startet mit dem Knoten 5, da $f(5) = 2 > 1 = f(3)$. Von 5 aus kann man in G^{-1} lediglich die 5 erreichen, also wird nur die 5 mit '3' beschriftet. Die 3 erhält die Beschriftung '4'. Der gerichtete Graph G besitzt also 4 starke Zusammenhangskomponenten.

12.3 Anwendung: 2-SAT

In diesem Kapitel stellen wir eine Anwendung des Begriffes vom starken Zusammenhang im Kontext der propositionalen Logik vor. In Abschnitt 3.4 hatten wir mit Horn-SAT eine natürliche und wichtige Einschränkung des aussagenlogischen Erfüllbarkeitsproblems

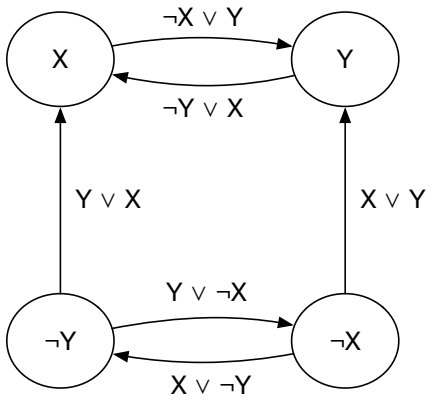


Abbildung 35: Der Implikationsgraph zum aussagenlogischen Ausdruck $(X \vee Y) \wedge (X \vee \neg Y) \wedge (\neg X \vee Y)$.

kennengelernt. Eine weitere solche Einschränkung ist das sogenannte *2-SAT Problem*. Die Einschränkung lautet, dass der gegebene aussagenlogische Ausdruck in konjunktiver Normalform vorliegt, so dass jede Klausel *maximal zwei Literale* enthält. Für solche Ausdrücke A lässt sich ein effizienter Algorithmus angeben, der entscheidet, ob A erfüllbar ist.

Beispiel 30.

$$(X \vee Y) \ \wedge \ (\neg X \vee Z) \ \wedge \ (\neg Z \vee \neg Y) \ \wedge \ (\neg Y \vee Z) \ \wedge \ (\neg Z \vee \neg X)$$

Erfüllbar oder nicht?

Ein effizienter Algorithmus für das 2-SAT Problem, der zuerst von Aspvall, Plass, und Tarjan publiziert wurde, führt das Problem auf die Berechnung der starken Zusammenhangskomponenten in einem gerichteten Graphen zurück. Wir definieren hierzu für einen gegebenen aussagenlogischen Ausdruck A mit höchstens zwei Literalen pro Klausel einen Hilfsgraphen, den sogenannten *Implikationsgraphen*. Zunächst möchten wir gerne mit der technischen Annahme arbeiten, dass alle Klauseln in der Eingabe *genau* zwei Literale besitzen. Dies ist nicht problematisch, da eine Klausel der Gestalt L durch eine Klausel der Gestalt $L \vee L$ ersetzt werden kann, ohne dadurch die Menge der erfüllenden Belegungen von A zu verändern. Weiterhin können wir annehmen, dass A keine Klauseln der Gestalt $X \vee \neg X$ enthält, da wir solche löschen können, ohne dadurch die Menge der erfüllenden Belegungen von A zu verändern.

Der *Implikationsgraph* ist nun der gerichtete Graph G, der für jede Variable X aus A zwei Knoten hat, die wir X und $\neg X$ nennen. Wir fügen in G genau dann eine Kante (L_1, L_2) ein, wenn $\neg L_1 \vee L_2$ eine Klausel in A ist. Siehe Abbildung 35.

Bemerkung 138. *Falls das Literal L_1 von der Form $\neg X$ ist, dann bedeutet $\neg L_1 = \neg(\neg X)$ das gleiche wie X; diese Konvention werden wir fortan stillschweigend verwenden. Weiterhin unterscheiden wir hier (wegen der Kommutativität von $\vee$) nicht zwischen $\neg L_1 \vee L_2$ und $L_2 \vee \neg L_1$. Es wird also für jede Kante von L_1 nach L_2 auch eine von $\neg L_2$ nach $\neg L_1$ eingeführt!*

Es ist klar, dass jede erfüllende Belegung diejenigen Literale, die in der gleichen starken Zusammenhangskomponente von G liegen, entweder allesamt wahr oder allesamt falsch machen muss. Wenn also für eine Variable X aus A die beiden Knoten X und $\neg X$ in der selben SZ-Komponente (siehe Kapitel 12) liegen, dann kann es keine erfüllende Belegung geben für A. Diese offensichtliche notwendige Bedingung für Erfüllbarkeit ist auch hinreichend.

Proposition 139. *Ein aussagenlogischer Ausdruck A mit genau zwei Literalen pro Klausel ist genau dann erfüllbar, wenn für alle Variablen X aus A im zugehörigen Implikationsgraph G die Knoten X und $\neg X$ in unterschiedlichen SZ-Komponenten liegen.*

Beweis. Wir haben bereits gesehen, dass die Existenz von einer Variablen X aus A mit X und $\neg X$ in der gleichen SZ-Komponente von G Unerfüllbarkeit von A bedeutet. Nehmen wir also an, dass es keine solche Variable gibt. Um eine erfüllende Belegung von G zu finden, betrachten wir die Beschriftung der Knoten in der umgekehrten topologischen Ordnung, die vom Algorithmus zur Berechnung der SZ-Komponenten aus Abschnitt 12.2 berechnet wird. Wenn L in dieser Ordnung eine kleinere Zahl trägt, als $\neg L$, so wird L auf 1 gesetzt, und $\neg L$ auf 0. (Nach Annahme können L und $\neg L$ nicht die gleiche Nummer tragen).

- Klauseln der Gestalt $\neg L_1 \vee L_2$, für die die Literale L_1 und L_2 in der gleichen SZ-Komponente liegen, werden erfüllt, da L_1 und L_2 den gleichen Wahrheitswert zugewiesen wird.
- Klauseln der Gestalt $\neg L_1 \vee L_2$, für die die Literale L_1 und L_2 in verschiedenen SZ-Komponenten liegen, werden erfüllt, da L_2 in der topologischen Ordnung kleiner ist als L_1, und damit L_2 auf 1 gesetzt wird und L_1 auf 0.

Also erfüllt die Belegung alle Klauseln. □

Mit dem Algorithmus zur Berechnung der SZ-Komponenten für gerichtete Graphen aus Abschnitt 12.2 gibt es damit einen Algorithmus für das 2-SAT Problem mit linearer Laufzeit!

Beispiel 31. *Im Implikationsgraph aus Abbildung 35 haben wir zwei starke Zusammenhangskomponenten $\{X, Y\}$ und $\{\neg Y, \neg X\}$. Die vom Algorithmus berechnete Beschriftung ist 1 für X und Y, und 2 für $\neg X$ und $\neg Y$. Insbesondere ist die Beschriftung von X kleiner als die von $\neg X$, und X wird daher auf 1 gesetzt. Weiterhin ist die Beschriftung von Y kleiner als die von $\neg Y$, und Y wird daher ebenfalls auf 1 gesetzt. Diese Belegung erfüllt alle drei Konjunkte der aussagenlogischen Formel aus Abbildung 35.*

Übungen.

49. *Gerichtete Eulerzüge* sind analog definiert zum ungerichteten Fall (Abschnitt 7.7): es handelt sich um geschlossene gerichtete Kantenzüge in gerichteten Graphen G, die alle Kanten von G genau einmal durchlaufen. Zeigen Sie, dass ein stark zusammenhängender gerichtete Graph genau dann einen gerichteten Eulerzug besitzt, wenn jeder Knoten gleich viele Vorgänger wie Nachfolger besitzt. Zeigen Sie, dass diese Aussage für nicht stark zusammenhängende Graphen im allgemeinen nicht gilt. Gibt es Graphen, die nicht stark zusammenhängend sind, aber dennoch einen gerichteten Eulerzug besitzen?

50. Ein *Turniergraph* ist ein gerichteter Graph ohne Schleifen, in dem es zwischen je zwei verschiedenen Knoten genau eine Kante gibt. Wir betrachten nun einen endlichen Turniergraphen, in dem jeder Knoten gleich viele Vorgänger wie Nachfolger hat. Zeigen Sie, dass ein solcher Graph einen geschlossenen Eulerzug besitzt.

51. Beweisen Sie: jedes endliche Turnier besitzt einen gerichteten Pfad, der alle Knoten des Graphen genau einmal durchläuft.

52. Jedes endliche Turnier besitzt eine Ecke, *Königin* genannt, von der aus alle weiteren Ecken durch gerichtete Pfade mit höchstens zwei Kanten erreicht werden können.

53. Ein endlicher Turniergraph hat genau dann nur eine Königin, wenn er eine Alleinherrscherin besitzt, d.h., eine Ecke ohne eingehende Kanten. Diese ist dann die einzige Königin.

54. Es existiert kein endliches Turnier mit genau zwei Königinnen.

55. Formulieren und beweisen Sie eine Version von Menger's Satz (Satz 69) für gerichtete Graphen.

12.4 Breitensuche

Die typische Situation, in der man *Breitensuche* einsetzt, und nicht Tiefensuche, ist die Suche nach einem *kürzesten Pfad* von einem gegebenen Knoten s zu einem gegebenen Knoten t in einem gerichteten Graphen. Breitensuche kann auch dazu verwendet werden, um die *kürzesten Kreise* in einem gegebenen Graphen zu berechnen.

Man kann die Breitensuche als Abänderung der Tiefensuche verstehen. Dazu kommen wir auf ein Implementierungsdetail der Tiefensuche. Wie entscheiden wir in der Tiefensuche, welche Kante wir als nächstes betrachten wollen? Dazu kann man einen *Stapel* verwenden. Immer wenn wir einen neuen Knoten v betreten, werden alle Kanten $(v, u) \in E$ in beliebiger Reihenfolge auf den Stapel gelegt. Wenn wir dann entscheiden müssen, welche Kante wir als nächstes betrachten wollen, nehmen wir die *oberste* Kante vom Stapel (also die, die *zuletzt* in den Stapel eingefügt wurde). Auf diese Weise kann man das rekursive Program in

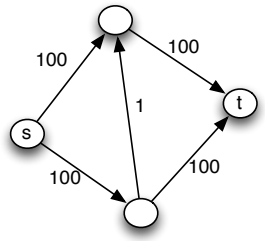


Abbildung 36: Ein kleines Transportnetzwerk.

Abbildung 31 in ein Programm mit einer einzigen Schleife verwandeln, die so oft durchlaufen wird, bis der Stapel leer ist.

Nun zur Breitensuche. Wir ersetzen hierzu einfach den Stapel der Tiefensuche durch eine *Warteschlange*, und anstatt stets die oberste Kante zu wählen, nehmen wir die unterste (also die, die *zuerst* in die Schlange eingefügt wurde). Das ist der ganze Unterschied zwischen Breitensuche und Tiefensuche.

Die Schwäche der Implementierungen der Tiefen- und Breitensuche mit Stapel und mit Schlange, wie sie in diesem Abschnitt vorstellt wurden, ist der Speicherplatzverbrauch, der linear sein kann in der Anzahl der Kanten. Allerdings kann das sowohl bei der Tiefensuche als auch der Breitensuche vermieden werden. Bei der Tiefensuche zum Beispiel durch die Implementierung, wie wir sie in Abschnitt 12.1 vorgestellt haben; hier ist der Speicherbedarf linear in der Anzahl der Knoten. Da es in vielen Graphen viel weniger Knoten als Kanten gibt, ist das besser. Selbiges geht auch für die Breitensuche, indem man sich Knoten statt Kanten merkt, und aufpasst, dass Knoten nicht doppelt in die Schlange aufgenommen werden.

12.5 Transportnetze

Ein *Transportnetz* (oft auch *Netzwerk*) (V, E, s, t, w) besteht aus einem endlichen gerichteten Graph (V, E) mit zwei ausgezeichneten Knoten s und t, zusammen mit einer *nichtnegativen Gewichtsfunktion*

$$w\colon E \to \mathbb{R}_{\geq 0}\ .$$

Statt vom 'Gewicht' spricht man bei Transportnetzen gern von der *Kapazität* einer Kante.

Definition 140 (Flüsse)**.** *Ein* Fluss *in einem Transportnetz* (V, E, s, t, w) *ist eine nichtnegative Kantenbewertung*

$$f\colon E \to \mathbb{R}_{\geq 0}$$

mit der Eigenschaft, dass für alle Knoten $v \in V \setminus \{s,t\}$ die folgende Gleichung gilt:

$$\sum_{(u,v)\in E} f(u,v) = \sum_{(v,u)\in E} f(v,u) \qquad \text{(‘Was in } v \text{ hineinfließt, fließt auch wieder hinaus’)}$$

Ein Fluss heißt *zulässig*, wenn $f(u,v) \leq w(u,v)$ für alle $(u,v) \in E$ gilt. Wenn f ein Fluss ist, und U eine Teilmenge von $V \setminus \{s,t\}$, dann sieht man durch Aufsummieren über die Elemente in U, dass gelten muss

$$\sum_{(u,v)\in E, u\notin U, v\in U} f(u,v) = \sum_{(v,u)\in E, v\in U, u\notin U} f(v,u) \qquad \text{(‘Flusserhaltung’).}$$

Partitioniert man die Knotenmenge V in beliebiger Weise in zwei Mengen M, N, von denen die eine s und die andere t enthält, also

$$V = M \cup N, \quad M \cap N = \emptyset, \quad s \in M, t \in N$$

dann definiert man die *Stärke* $||f||$ des Flusses f wie folgt:

$$||f|| := \sum_{(v,u)\in E, v\in M, u\in N} f(v,u) - \sum_{(u,v)\in E, v\in M, u\in N} f(u,v). \tag{8}$$

Min-Cut Max-Flow

Ein *Schnitt* in einem Netzwerk ist eine Menge $S \subseteq E$ mit der Eigenschaft, dass es im gerichteten Graphen $(V, E \setminus S)$ keinen gerichteten Pfad von s nach t gibt. Anders formuliert ist ein Schnitt eine Kantenmenge, die aus jedem gerichteten Weg von s nach t mindestens eine Kante enthält. Die *Kapazität* eines Schnittes S ist definiert als

$$w(S) := \sum_{e\in S} w(e)\ .$$

Lemma 141. *Ist S ein Schnitt des Transportnetzwerks (V, E, s, t, w) und f ein zulässiger Fluss, dann gilt $||f|| \leq w(S)$.*

Satz 142 (Ford[70] und Fulkerson[71]; Max Flow = Min Cut)**.** *Es sei (V, E, s, t, w) ein Transportnetz. Dann gilt*

$$\max_{f \text{ zulässiger Fluss}} ||f|| = \min_{S \text{ Schnitt}} w(S).$$

In Worten: die Stärke des größten zulässigen Flusses ist genau so groß wie die Kapazität des kleinsten Schnitts.

[70] Lester Randolph Ford junior; geboren am 23. September 1927 in Houston.

[71] Delbert Ray Fulkerson; geboren am 14. August 1924 in Tamms, Illinois. Gestorben am 10. Januar 1976 in Ithaca, New York.

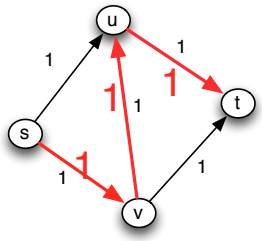


Abbildung 37: Ein Beispiel zur Frage: Warum durchlaufen wir Kanten bei der Suche nach einer Flussverbesserung manchmal rückwärts?

Zum Beweis dieses Satzes stellen wir eine Methode vor, die ebenfalls nach Ford und Fulkerson benannt ist, und die zu einem gegebenen Netzwerk einen zulässigen Fluss f und einen Schnitt S der Kapazität $w(S) = ||f||$ konstruiert. Nach dem Hilfssatz muss dieser Fluss maximal stark und der Schnitt von minimaler Kapazität sein. Der Satz ist damit also bewiesen.

Flussverbesserung

Unser Algorithmus leistet bei einem Durchlauf folgendes: ausgehend von einem zulässigen Fluss im gegebenen Netzwerk T konstruiert der Algorithmus einen stärkeren zulässigen Fluss, oder einen Schnitt, dessen Kapazität gleich der Stärke des gegebenen Flusses ist. Zu Beginn starten wir mit dem zulässigen Fluss, der konstant Null ist. Sind die auftretenden Gewichte und Flussstärken ganzzahlig, dann ist auch die Verbesserung ganzzahlig. Damit kann man zeigen, dass der Algorithmus irgendwann (nach endlich vielen Durchläufen) den Fluss nicht mehr verbessern kann und deshalb einen minimalen Schnitt gefunden haben muss.

Eine wichtige Idee hierbei ist, dass wir auch untersuchen müssen, ob es Kanten gibt, bei denen der Fluss 'in die falsche Richtung' fließt. In solchen Fällen ist es für manche Flussverbesserungen notwendig, den Rückfluss zu verringern. Dazu betrachten wir das Beispiel in Abbildung 37. Dort finden wir ein kleines Netzwerk, mit einem Fluss der Stärke 1 in rot eingezeichnet. Es gibt keinen Pfad, der entlang von Kanten über nicht voll ausgelastete Kanten von s nach t führt. Aber trotzdem ist der Fluss nicht maximal, denn wenn man den Fluss über die Kante (v, u) um 1 verringert, kann man ihn auf den Kanten (s, u) und (v, t) um 1 vergrößern.

Eine *Doppelkante* in einem gerichteten Graphen (V, E) sind zwei Kanten in E der Gestalt (u, v) und (v, u). Im folgenden wollen wir mit Transportnetzwerken arbeiten, in denen

der zugrundeliegende gerichtete Graph ohne Doppelkante ist[72]; das ist keine wichtige Einschränkung, da man eine Doppelkante (u, v) und (v, u) wir folgt eliminieren kann:

- Entfernen von (u, v),
- Einfügen von einem neuen Knoten w und den Kanten (u, w), (w, v) mit der gleichen Kapazität wir vormals (u, v).

Aus einem Fluss in einem dergestalt modifizierten Netzwerk kann man einen Fluss gleicher Stärke im ursprünglichen Netzwerk gewinnen, und andersherum, so dass das Flussproblem für die beiden Netzwerke im Prinzip das gleiche ist.

Definition 143. *Sei $T := (V, E, s, t, w)$ ein Transportnetzwerk ohne Doppelkanten, und sei f ein zulässiger Fluss in T. Dann ist das* Residualnetzwerk (von T bezüglich f) *das Transportnetzwerk $T_f := (V, E_1 \cup E_2, s, t, w')$ mit*

$$\begin{aligned}
E_1 &:= \{(u, v) \in E \mid f(u, v) < w(u, v)\} && \textit{(Nicht voll ausgelastete Kanten);}\\
w'(u, v) &:= w(u, v) - f(u, v) && \textit{falls } (u, v) \in E_1;\\
E_2 &:= \{(u, v) \mid (v, u) \in E, f(v, u) > 0\} && \textit{(Rückkanten);}\\
w'(u, v) &:= f(v, u) && \textit{falls } (u, v) \in E_2.
\end{aligned}$$

Bei der Methode von Ford-Fulkerson berechnen wir eine Flussverbesserung, indem wir einen gerichteten Pfad P von s nach t in T_f berechnen; ein solcher Pfad heißt *augmentierender Pfad*. Dass wir den gleichen Ausdruck bereits im Zusammenhang von Paarungen verwendet haben, ist Absicht; der Zusammenhang wird in Abschnitt 12.7 besprochen. Die *Stärke* eines augmentierenden Pfades P ist das Minimum der Kapazität der Kanten von P in T_f (um diesen Wert kann der Fluss verbessert werden, wie wir gleich sehen werden). Falls ein augmentierender Pfade existiert, so kann er zum Beispiel mit Tiefensuche leicht berechnet werden, und falls keiner existiert, so kann man das zum Beispiel mit Tiefensuche ebenfalls herausfinden.

1. Fall. Es gibt einen augmentierenden Pfad P der Stärke δ. In diesem Fall erhalten wir den um δ besseren Fluss f' wie folgt

- $f'(u, v) := f(u, v) + \delta$ falls $(u, v) \in E_1$ Vorwärtskante in P ist,
- $f'(u, v) := f(u, v) - \delta$ falls $(v, u) \in E_2$ Rückkante in P ist,
- und $f'(u, v) = f(u, v)$ sonst.

2. Fall. Es gibt keinen augmentierenden Pfad. Wenn es keinen augmentierenden Pfad in T_f gibt, so betrachten wir die Menge S aller Kanten $(u, v) \in E$, so dass es einen

[72] Eine Alternative wäre, mit gerichteten Multikanten zu arbeiten.

Pfad in T_f von s nach u gibt, aber nicht zu v. Diese Menge ist ein Schnitt von T, dessen Kapazität genau die Stärke von f ist.

Damit haben wir den Max-Flow Min-Cut Satz von Ford und Fulkerson (Satz 142) bewiesen. □

Allerdings ist dieses Verfahren zur Berechnung eines stärksten Flusses nicht effizient. Das sieht man bereits am Beispiel aus Abbildung 36: denn wenn wir Pech haben, erhalten wir in jeder Runde mit dem Algorithmus von Ford-Fulkerson eine Veränderung des Flusses der mittleren Kante, mit $\delta(t) = 1$. In dem Fall benötigen wir zur Berechnung des größtmöglichen Flusses 200 Durchläufe des Algorithmus. Und wenn die Kapazitäten irrational sein dürfen, so kann es sogar passieren, dass das Verfahren überhaupt nicht terminiert! Oder dass es gegen einen Fluss konvergiert, dessen Stärke gar nicht maximal ist.

12.6 Der Algorithmus von Edmonds-Karp

Die Idee, um aus der Methode von Ford-Fulkerson einen *effizienten* Algorithmus zu gewinnen, besteht darin, den Fluss stets entlang von *kürzesten* augmentierenden Pfaden zu verbessern; einen solchen können wir mit Breitensuche (Abschnitt 12.4) berechnen. Die Länge eines Pfades ist hier die Anzahl der Kanten auf dem Pfad – wir beachten also die Gewichte der Kanten hierbei *nicht*. Der Algorithmus wurde (in einer schnelleren Variante) zuerst von Dinitz[73] publiziert, blieb aber im Westen unbekannt, bis ihn Edmonds[74] und Karp[75] 1972 wiederentdeckten, in der Variante, die wir hier vorstellen.

Für $u, v \in V$ schreiben wir $\mathrm{dist}_f(u, v)$ (kurz für *Distanz*) für die Länge (wir zählen die Anzahl der Kanten) des kürzesten Pfades von u nach v in T_f. Die so definierte Distanz erfüllt die *Dreiecksungleichung*: für alle $u, v, w \in V$ gilt:

$$\mathrm{dist}_f(u, w) \leq \mathrm{dist}_f(u, v) + \mathrm{dist}_f(v, w).$$

Lemma 144. *Sei (V, E, s, t, w) ein Transportnetzwerk, f ein zulässiger Fluss, und f' ein Fluss nach einer Flussverbesserung im Algorithmus von Edmonds-Karp. Dann gilt für alle $v \in V \setminus \{s, t\}$ dass $\mathrm{dist}_{f'}(s, v) \geq \mathrm{dist}_f(s, v)$.*

Beweis. Angenommen, es gäbe einen Knoten $v \in V \setminus \{s, t\}$ mit $\mathrm{dist}_{f'}(s, v) < \mathrm{dist}_f(s, v)$. Wir wählen v aus allen Knoten mit dieser Eigenschaft so, dass $\mathrm{dist}_{f'}(s, v)$ minimal ist. Betrachte einen kürzesten Pfad P von s zu v in $T_{f'}$, und sei (u, v) die letzte Kante auf P. Da P kürzest möglich gilt

$$\mathrm{dist}_{f'}(s, v) = \mathrm{dist}_{f'}(s, u) + 1. \tag{9}$$

[73] Yefim A. Dinitz.
[74] Jack R. Edmonds; geboren am 5. April 1934.
[75] Richard Manning Karp; geboren am 3. Januar 1935 in Boston.

Fall 1. (u, v) ist Kante in T_f. Dann gilt

$$\begin{aligned}
\mathrm{dist}_f(s, v) &\leq \mathrm{dist}_f(s, u) + 1 && \text{(Dreiecksungleichung)} \\
&\leq \mathrm{dist}_{f'}(s, u) + 1 && \text{(nach Wahl von } v\text{)} \\
&= \mathrm{dist}_{f'}(s, v) && \text{(nach (9))}
\end{aligned}$$

im Widerspruch zur Annahme.

Fall 2. (u, v) ist keine Kante in T_f. Da aber (u, v) eine Kante von $T_{f'}$ ist, muss es zwischen f und f' eine Veränderung des Flusses von v nach u gegeben haben. Eine Flussveränderung auf (v, u) findet im Algorithmus von Dinitz immer auf einem kürzesten Weg von s nach u statt, also gilt

$$\begin{aligned}
\mathrm{dist}_f(s, v) &\leq \mathrm{dist}_f(s, u) - 1 && \text{(Dreiecksungleichung)} \\
&\leq \mathrm{dist}_{f'}(s, u) - 1 && \text{(nach Wahl von } v\text{)} \\
&= \mathrm{dist}_{f'}(s, v) - 2 && \text{(nach (9))}
\end{aligned}$$

ebenfalls ein Widerspruch. □

Proposition 145. *Der Algorithmus von Edmonds-Karp führt auf einem Transportnetzwerk (V, E, s, t, w) höchstens $|V| \cdot |E|$ viele Flussverbesserungen durch.*

Beweis. Es sei f ein Fluss und P ein augmentierender Pfad der Stärke δ. Eine Kante (u, v) in einem Residualnetzwerk T_f heißt *kritisch auf P* falls entweder (u, v) eine Vorwärtskante in T_f ist und $w(u, v) = f(u, v) + \delta$ gilt, oder (u, v) eine Rückkante in T_f ist, und $f(v, u) = \delta$ gilt. Das bedeutet, dass nach der Flussverbesserung die Kante (u, v) verschwindet. Auf jedem augmentierenden Pfad ist mindestens eine Kante kritisch.

Wir betrachten nun $(u, v) \in E$, und fragen uns, wie oft (u, v) während der Ausführung des Algorithmus kritisch sein kann. Wenn (u, v) zum ersten Mal kritisch wird, haben wir $\mathrm{dist}_f(s, v) = \mathrm{dist}_f(s, u) + 1$, da wir bei Edmonds-Karp einen kürzesten augmentierenden Pfad wählen. Die Kante (u, v) verschwindet dann im Residualnetzwerk, und kann nur dann wieder auftauchen, falls die Kante (v, u) als Rückkante in einem augmentierenden Pfad ausgewählt wird. Es sei f' der Fluss, den wir mit diesem augmentierenden Pfad berechnen. Dann haben wir

$$\begin{aligned}
\mathrm{dist}_{f'}(s, v) &= \mathrm{dist}_{f'}(s, u) + 1 && \text{(da } (u, v) \text{ Vorwärtskante in } T_{f'}\text{)} \\
&\geq \mathrm{dist}_f(s, u) + 1 && \text{(Lemma 144)} \\
&= \mathrm{dist}_f(s, v) + 2 && \text{(da } (v, u) \text{ auf kürzestem Pfad in } T_f\text{)}
\end{aligned}$$

Daher kann (u, v) höchstens $|V|$ mal kritisch werden. Da es höchstens $|E|$ Kanten gibt, werden maximal $|E||V|$ Flussverbesserungen durchgeführt. □

Jede Iteration der Methode von Ford-Fulkerson kann mit Hilfe von Breitensuche so implementiert werden, dass die Laufzeit linear ist in der Anzahl der Kanten. Die gesamte Laufzeit des Algorithmus von Edmonds-Karp ist dann also beschränkt durch $c \cdot |V| \cdot |E|^2$ für eine geeignete Konstante c.

Ein großartiges, ganz aktuelles Resultat aus dem Jahr 2022 [3], vorgestellt auf der Konferenz *Foundations of Computer Science (FOCS, 2023)* zeigt, dass maximale Flüsse in fast linearer Zeit in der Größe der Eingabe berechnet werden können.

12.7 Nochmal Paarungen

Viele Argumente im Kapitel zu Flüssen fühlen sich ähnlich an wie Argumente, die wir beim Studium von Paarungen kennengelernt haben. Das ist kein Zufall. Es gibt hier viele Querbezüge.

Übersetzung in ein Flussproblem. Man verwandelt den gegebenen bipartiten Graphen G in ein Transportnetz. Zunächst werden die Kanten *orientiert*: wenn A, B eine Bipartition von G ist, dann werden Kanten $\{u, v\}$ mit $u \in A$ und $v \in B$ zu gerichteten Kanten (u, v). Diese Kanten erhalten die Kapazität $|V| + 1$. (Für die Reduktion auf das Flussproblem könnten wir hier auch die Kapazität 1 wählen. Für die Übersetzung zwischen Schnitten im Transportnetzwerk und Knotenüberdeckungen im Graphen, die wir weiter unten angeben, ist es allerdings praktisch, das Kantengewicht auf $|V| + 1$ zu setzen.) Dann fügt man eine Quelle s und eine Senke t hinzu, und verbindet s über eine gerichtete Kante mit Kapazität 1 mit allen Knoten in A, und alle Knoten in B über eine gerichtete Kante der Kapazität 1 mit t. Wir bezeichnen das resultierende Transportnetzwerk mit (V, E, s, t, w).

Von Flüssen zu Paarungen und Andersrum. Wir berechnen nun mit dem Verfahren aus Abschnitt 12.5 einen größtmöglichen Fluss im oben beschriebenen Transportnetzwerk. Weil der Algorithmus von Ford-Fulkerson bei ganzzahligen Kapazitäten auch ganzzahlige Flüsse liefert, so fließt zu jedem Knoten aus A entweder ein Fluss der Stärke eins oder null. Entsprechend kann auch zu jedem Knoten aus B nur ein Fluss der Stärke eins oder null fließen, und die Kanten (u, v) zwischen A und B mit $f(u, v) = 1$ teilen sich keine Knoten. Also ist die entsprechende Menge an Kanten $\{u, v\}$ in G eine Paarung in G. Die Anzahl der Kanten in dieser Paarung ist gleich der Stärke des Flusses. Umgekehrt erhält man aus jeder Paarung einen Fluss der entsprechenden Stärke. Wir können also insbesondere Algorithmen zur Berechnung von stärksten Flüssen zur Berechnung von größtmöglichen Paarungen verwenden.

Auch der Satz von König (Satz 81) kann man mit dem Min-Cut Max-Flow Satz bewiesen werden. Der Satz von König besagte, dass in einem bipartiten Graphen G die Größe einer größtmöglichen Paarung gleich der Größe einer kleinstmöglichen Kantenüberdeckung von G ist.

Von Schnitten zu Knotenüberdeckungen und Andersrum. Alle Kanten der Gestalt (s, u) mit $u \in A$ bilden einen Schnitt S in (V, E, s, t, w); also gilt $|S| \leq |V|$. Daraus

folgt, dass jeder Schnitt minimaler Kapazität keine Kante von A nach B verwenden darf, denn jede solche Kante hat die Kapazität $|V| + 1$. Ein minimaler Schnitt besteht also nur aus Kanten, die in s starten oder in t enden. Setze für eine Menge C von solchen Kanten

$$X := \{v \in V \mid (s, v) \in C\}$$
$$Y := \{u \in V \mid (u, t) \in C\} .$$

Dann ist C genau dann ein Schnitt in (V, E, s, t, w), wenn es keine Kante von X nach Y gibt, also wenn $X \cup Y$ eine Knotenüberdeckung von G ist. Die Größe dieser Überdeckung ist gleich der Kapazität des Schnittes. Also ist die Größe der kleinsten Überdeckung von G gleich der Größe eines minimalen Schnittes in (V, E, s, t, w).

Wegen des Max-Cut Min-Flow Satzes ist die Größe des kleinsten Schnittes gleich der Größe des größtmöglichen Flusses, und die wiederum, wie wir oben gesehen haben, gleich der größtmöglichen Paarung in G. Wir haben also den Satz von Kőnig (ein weiteres Mal) bewiesen.

Übungen.

56. Zeigen Sie den Satz von Menger für gerichtete Graphen (Übung 55) mit Hilfe des Max-Cut Min-Flow Satzes.

'La pensée n'est qu'un éclair au milieu de la nuit. Mais c'est cet éclair qui est tout'
Henri Poincaré

Literatur

[1] M. Agrawal, N. Kayal, and N. Saxena. Primes is in P. *Annals of Mathematics*, 160(2):781–793, 2004.

[2] F. Bergeron, G. Labelle, and P. Leroux. *Théorie des espèces et combinatoire des structures arborescentes*. LaCIM, Montréal, 1994. English version: Combinatorial Species and Tree-like Structures, Cambridge University Press (1998).

[3] L. Chen, R. Kyng, Y. P. Liu, R. Peng, M. P. Gutenberg, and S. Sachdeva. Almost-linear-time algorithms for maximum flow and minimum-cost flow. *Commun. ACM*, 66(12):85–92, Nov. 2023.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT Press, 2009. Third edition.

[5] R. Diestel. *Graphentheorie*. Springer-Verlag, Heidelberg, 2010. Vierte Auflage.

[6] J. Matoušek and J. Nešetřil. *Invitation to Discrete Mathematics*. Oxford University Press, 1998.

[7] S. Perifel. *Complexité algorithmique*. Ellipses, 2014.

[8] R. P. Stanley. *Enumerative Combinatorics, Volume 1*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2011. Second Edition.

Index

k-Färbbarkeit, 71
k-regulärer Graph, 84
Äquivalenz, 24
Äquivalenzklasse, 87
Überdeckung, 84
äquivalent, 26
öffentliche Schlüssel, 63
2-SAT, 131

Abbildung, 14
abelsch, 53
Addition, 35
Addition modulo n, 42
Adjazenz, 69
Adjazenzmatrix, 109
alternierender Pfad, 83
Antikette, 95
antisymmetrisch, 86
Aquivalenzrelation, 86
Argument einer Funktion, 21
assoziativ, 6, 22
augmentierender Pfad, 83, 137
Ausdruck, 25
Ausmultiplizieren, 36
Aussagenlogik, 25
aussagenlogische Formel, 25
Auswahlaxiom, 17
axiomatische Mengenlehre, 11

Baum, 73, 125
Baumkanten, 127
Baumlemma von Kőnig, 102
Belegung, 25
Bellsche Zahlen, 89
bijektiv, 14
Bild, 14
Bild von A unter f, 14
binäre Exponentiation, 45
Binomialkoeffizienten, 9
bipartiter Graph, 72
Bipartition, 72
Blatt (eines Baumes), 73
Block, 75
Blockgraph, 75
Boolsche Funktion, 21
Boolsche Relation, 28
Brücke, 75
Breitensuche, 133

chinesischer Restsatz, 47

De Morgansche Regeln, 23
Definitionsbereich, 14
Diedergruppe, 54
Differenz, 6
Diffie-Hellman-Merkle Verfahren, 59
Disjunkt, 27
disjunkt, 7
disjunkte Vereinigung, 71
Disjunktion, 22
disjunktive Normalform, 27
diskretes Logarithmusproblem, 60
distributiv, 6, 22
dual, 121
Dualität, 23

Ebene, 115
ebener Graph, 115
ebener Multigraph, 115, 117
Ecke, 69
Einbahnfunktionen, 60
Einheit, 54
Einheitengruppe, 55
Einheitssphäre, 118
Einschränkung, 15
Erfüllbarkeit, 26
Erfüllbarkeitsproblem, 28
erhalten, 31

Erreichbarkeit, 125
Erreichbarkeitsrelation, 126
Ersetzungsschema, 12
erweiterter euklidischer Algorithmus, 41
Erzeugung, 61
euklidischer Algorithmus, 40
eulersche ϕ-Funktion, 56
eulersche Polyederformel, 116
eulerscher Graph, 81
Eulerzug, 81
    gerichtet, 133
exklusives Oder, 24
Exponentiation, 35
Extensionalität, 12

Faktormenge, 87
Faktorordnung, 94
Fakultät, 9
feiner, 88
Fixpunkt, 19
Flächen, 116
Flusserhaltung, 135
Fundierung, 12
Funktion, 14

Gebiete, 116
Gelenkpunkt, 75
geordnete Paare, 8
Gerüst, 108
gerichteter
    Kreis, 125
    Pfad, 125
gerichteter Graph, 125
geschlossener
    Kantenzug, 125
Gewichtsfunktion, 134
Gleichheitsrelation, 86
gröber, 88
größter gemeinsamer Teiler, 38
Grad (eines Knoten), 69
Gradmatrix, 110
Graph von f, 14
Graphminorenrelation, 122
Gruppe, 52
Gruppenisomorphie, 57

Hamiltonkreis, 82, 124
Hasse-Diagramm, 92
Heiratssatz, 83
Hilbert's Programm, 12
Hintereinanderausführung, 15
Homöomorphismus, 118
Homomorphieregel, 42
Horn-Formel, 30
Horn-SAT, 30

idempotent, 6, 22
Identität, 15
Identitätsfunktion, 15
Implikation, 24
Implikationsgraph, 131
Index (einer Untergruppe), 62
induzierter Subgraph, 70
Infix-Notation, 22
injektiv, 14
inverses Element, 52
irreflexiv, 86
Isomorphie von Graphen, 70

Königsberger Brückenproblem, 81
kürzester Pfad, 133
Kante, 69
Kantengraph, 80
Kantengraph (eines Polyeders), 121
Kantenkontraktion, 110
Kantenzug, 70, 125
Kantenzusammenhang, 80
Kapazität (einer Kante), 134
Kapazität (eines Schnittes), 135
Kardinalität, 5
Kern (einer Abbildung), 90
Kette, 95
Klausel, 27

Knoten, 69
Knotenpostordnung, 127
Knotenpräordnung, 126
kombinatorischer Isomorphismus, 119
kommutativ, 6, 22
Komplement (einer Menge), 7
Komplement (eines Graphen), 70
Komposition, 15
Komposition (von Relationen), 95
Kongruenz modulo n, 43
Kongruenzklassen, 43
Konjunkt, 27
Konjunktion, 22
Kontinuum, 17
Kontraposition, 26
Kreis (in einem Graph), 70
Kryptographie, 62

Las Vegas Algorithmus, 47
leere Menge, 5
Lemma von Bézout, 40
Lemma von Berge, 83
Lemma von Dickson, 98
Lemma von Euler-Fermat, 63
Lemma von Fermat, 62
lexikographische Ordnung, 100
lineare Erweiterung, 93
lineare Ordnung, 93
Literal, 27

Matching, 82
Max Flow = Min Cut, 135
Mengenangaben durch Aussonderung, 5
Minor, 122
Minorentheorem, 122
Monto Carlo Algorithmus, 48
Multigraph, 111, 115, 117
Multiplikation, 35
Multiplikation modulo n, 42

Nachfahrenrelation, 101
Nachfolger, 33, 125
Nebenklassen, 61
Negation, 21
Netzwerk, 134
neutrales Element, 52
NP, 29
Nullteiler, 54

offener Eulerzug, 81
Ohrendekomposition, 77
Ohrenzerlegung, 77
Operation, 14, 21
Ordnung (einer Gruppe), 61
Ordnung (eines Gruppenelementes), 61

Paarmenge, 12
Paarung, 82
partielle Ordnung, 92
Partition, 87
perfekte Paarung, 82
Permutation, 18
planarer Graph, 115
Polyeder, 121
Polygonzug, 115
Potenzmenge, 8
Prädikatenlogik, 25
Präfixordnung, 100
Prüfercode, 106
prim, 36
Primitivwurzel, 59
Primzahl, 36
Produktmenge, 8

Quasiordnung, 94
Quelle, 125
Querkanten, 127

Rückwärtskanten, 127
randomisierter Algorithmus, 47
reduziert, 31
reflexiv, 86
Rekursionsformel, 89
Relation, 8

Repräsentant, 94
Residualnetzwerk, 137
Rest, 39
Restklassen, 42
RP, 48
RSA, 63
Russellsche Antinomie, 11

Satz von Cayley, 105
Satz von Dilworth, 96
Satz von Ford Fulkerson, 136
Satz von Kőnig, 84
Satz von Kirchhoff, 109
Satz von Kuratowski, 123
Satz von Lagrange, 62
Satz von Menger, 77
Satz von Steinitz, 121
Satz von Szpilrajn, 93
Satz von Whitney, 120
Schlusszeit (Tiefensuche), 127
Schnitt, 6
Schubfachprinzip, 96, 97
Semantik, 25
semilineare Ordnung, 101
Senke, 125
Sloane's Oline-Enzyklopädie, 89
Spannbaum, 108
Stärke (eines Flusses), 135
Stapel, 133
starke Zusammenhangskomponente, 128
Startzeit (Tiefensuche), 127
Stelligkeit (einer Operation), 21
Stirlingsche Formel, 19
Stirlingsche Zahlen, 88
Subgraph, 70
Subtraktion modulo n, 42
surjektiv, 14
symmetrisch, 86
symmetrische Differenz, 83
symmetrische Gruppe, 53
Syntax, 25

Tautologie, 26
Teilbarkeitsrelation, 36
Teiler, 36
teilerfremd, 46
Teilfolgenordnung, 100
Teilmenge, 5
teilt, 36
Teilwortordnung, 100
Tiefensuche, 126
Tiefensuchwald, 127
topologische Sortierung, 130
topologischer Isomorphismus, 119
transitiv, 86
transitive Hülle, 88, 95
transitive reflexive Hülle, 95
transitiver Abschluss, 126
Transportnetz, 134
Transposition, 19
Triangulation, 117

Umkehrabbildung, 15
Unerfüllbarkeit, 26
ungerade Kreise, 72
ungerichteter Graph, 69
Untergruppe, 60

Variablensymbole, 25
Vereinigung, 6
Verfeinerung, 88
Verschlüsselungsverfahren, 63
vollständige Induktion, 34
vollständige Tiefensuche, 127
Vorgänger, 33, 125
Vorwärtskanten, 127

Wald, 73, 125
Warteschlange, 134
Wohlordnung, 33
Wohlordnungsprinzip, 34
Wohlquasiordnung, 97
Wurzel (eines Baumes), 101

Zermelo-Fraenkel-Mengenlehre, 12
Zielbereich, 14
ZPP, 47
Zusammenhang (in Graphen), 71
Zusammenhangskomponente von G, 71
zweifach zusammenhängend, 75, 124
Zyklenschreibweise, 18
zyklische Gruppe, 57
zyklische Permutation, 18
Zyklus, 19