Hochschule Aalen – Technik und Wirtschaft

Grundlagen der Mathematik

Skriptum des Sommersemesters 2020 mit Videolinks

Florian Wörz

M. Sc. in Mathematik

Basierend auf dem Skript von Prof. Dr. Thomas Thierauf

Hochschule Aalen – Technik und Wirtschaft

Disclaimer
Dies ist ein Skript zur Vorlesung „Grundlagen der Mathematik“ in den Bachelor-Studiengängen *Allgemeine Informatik*, *IT-Sicherheit*, *Medieninformatik*, *Software Engineering*, und *Data Science*, welche an der Hochschule Aalen von mir im Sommersemester 2020 als Lehrbeauftragter gehalten wurde. Ich bin Herrn Prof. Dr. Thomas Thierauf zutiefst zu Dank verpflichtet, da die ursprüngliche Version des Skriptes auf ihn zurückgeht.

Die Videomaterialen zur Vorlesung sind abrufbar unter `https://bit.ly/GdM-Playlist`.

Die vorliegende Version des Skriptes ist vorläufig. Bei Fragen, Wünschen und Verbesserungsvorschlägen können Sie gerne eine E-Mail an mich schreiben.

Version
14. Juli 2020, 14:58 Uhr

Mathematics, rightly viewed, possesses not only truth, but supreme beauty.

— Bertrand Russell

Vorwort

Diese Vorlesung bildet den Grundstock Ihrer Informatikausbildung. Die Inhalte werden Sie durch Ihr gesamtes Studium begleiten und immer wieder relevant sein. Die Vorlesung ist daher vermutlich eine der wichtigsten in Ihrem Studium.

Wir geben Ihnen zunächst einige allgemeine Tipps, wie Sie die Teilnahme an der Vorlesung erfolgreich gestalten können: Sehr wahrscheinlich ist die Vorlesung ganz anders als Sie erwartet haben, und möglicherweise empfinden Sie die Art der Stoffvermittlung und das Abstraktionsniveau als eine Zumutung. Fassen Sie dies als eine **Herausforderung** auf, und nehmen Sie diese Herausforderung an. Stellen Sie Fragen, wenn Sie etwas nicht verstehen und tauschen Sie sich (online*) mit Ihren Kommilitonen aus. Versuchen Sie den Inhalt der Vorlesung wirklich zu verstehen (und lassen Sie sich nicht entmutigen, wenn Ihnen das nicht immer sofort gelingt). Geben Sie sich nicht damit zufrieden, irgendwie durchzukommen. Versuchen Sie, mit Spaß und Interesse bei der Sache zu sein. Weitere Tipps finden Sie unter `https://image.informatik.htw-aalen.de/~thierauf/MatheGrundlagen/Handouts/mathe-anleitung.pdf`.

Inhalt der Vorlesung

Es werden grundlegende mathematische Begriffe und Techniken eingeführt. Dabei werden auch zahlreiche Anwendungen in der Informatik beleuchtet. Schwerpunkte liegen auf den Methoden aus der diskreten Mathematik, die zur Entwicklung und Analyse von effizienten Algorithmen gebraucht werden. Induktion, Kombinatorik, Zahlentheorie, Graphentheorie, Algebra sind besonders wichtige Gebiete für die Informatik und Grundlage weiterer Anwendungen, wie zum Beispiel Kryptographie und Codierungstheorie. Bei der Darstellung des Stoffes soll neben der mathematisch exakten Vorgehensweise besonders das intuitive Verständnis gefördert werden. Dies wird durch zahlreiche Beispiele und Übungsaufgaben unterstützt.

Voraussichtliches Inhaltsverzeichnis / Gliederung der Vorlesung:

1. **Logik**
   Junktoren, Formeln, Äquivalenz von Formeln, Normalformen, Minimierung.
2. **Mengenlehre**
   Definition, Notationen, Antinomien, Operationen, Potenzmenge, Kreuzprodukt.

* Diese Vorlesung wurde online während der COVID-19-Pandemie gehalten.

3. **Relationen und Graphen**
   Verkettung, Umkehrrelation, Äquivalenzrelationen, Ordnungsrelationen.
4. **Funktionen**
   Bijektionen, Permutationen, Mächtigkeiten.
5. **Vollständige Induktion**
   Summen, Rekursion, Eulersche Polyeder-Formel, planare Graphen, Graphenfärbung, Schleifen-Invarianten.
6. **Kombinatorik**
   Anzahlprobleme, Binomialkoeffizienten, Pascalsches Dreieck, Binomialtheorem.
7. **Zahlentheorie** (nur bei viel Zeit am Ende)
   Teilbarkeit, Euklidischer Algorithmus, Primzahlen, Kongruenzen, Chinesischer Restsatz, Eulersche φ-Funktion, Exkurs in die Kryptologie.

Modulbeschreibung

Modulart	Pflichtmodul
Credits	5 CP (ECTS)
SWS	4
Workload	Präsenz 60 h
	Selbststudium 90 h

Empfehlenswerte Literatur

- *Crashkurs Mathematik für Informatiker*, Stasys Jukna, Teubner, 2008.
- *Diskrete Strukturen 1*, Angelika Steger, Springer 2001.
- *Mathematik für Informatiker*, Dirk Hachenberger, Pearson Studium, 2005.
- *Diskrete Mathematik*, Laszlo Lovasz, Joszsef Pelikan und Katalin Vesztergombi, Springer, 2003.
- *Diskrete Mathematik für Einsteiger*, Alfred Beutelspacher, Vieweg 2002.
- *Concrete Mathematics*, Ronald Graham, Donald Knuth und Oren Patashnik, Addison-Wesley, 1989.
- *Introduction to Algorithms*, Thomas Corman, Charles Leiserson und Ronald Rivest und Cliff Stein The MIT Press, 2001.
- *Diskrete Mathematik*, Martin Aigner, Vieweg 2004.
- *Graphentheorie*, Reinhard Diestel, Springer 2000.
- *Mathe-Toolbox: Mathematische Notationen, Grundbegriffe und Beweismethoden*, Uwe Schöning und Hans A. Kestler, Lehmanns, 2015.

Modulziele

Fachkompetenz („Wissen und Verstehen“ und „Fertigkeiten“) Anhand von Beispielen in der Vorlesung sowie dem selbständigen Lösen von Übungsaufgaben können die Studierenden Sachverhalte durch logische Formeln beschreiben und dann vereinfachen. Sie können den prinzipiellen Aufbau der Mathematik aus der Mengenlehre erklären. Die Studierenden können die Beweismethode der vollständigen Induktion in Bereichen wie der Graphentheorie, der Programmverifikation und rekursiver Programmierung anwenden. Mit Mitteln der Kombinatorik sind die Studierenden in der Lage, die Laufzeiten von Algorithmen zu analysieren.

Überfachliche Kompetenz („Sozialkompetenz“ und „Selbstständigkeit“) Die Studierenden können sich in Kleingruppen organisieren, gemeinsam Übungsaufgaben bearbeiten und das erlernte Wissen vertiefen. In den angebotenen Tutorien können die Studierenden offene Fragen klären und verschiedene Lösungswege diskutieren.

Besondere Methodenkompetenz Die Studierenden verstehen Formeln als Handlungsvorschriften und können die daraus resultierenden Berechnungen durchführen. Sie sind in der Lage, Fragestellungen bedarfsgerecht zu erfassen und geeignete Verfahren zur Bearbeitung auszuwählen und zielgerichtet einzusetzen, um einen Transfer zu ähnlich gelagerten Fragestellungen herzustellen.

Organisation und Ablauf der Vorlesungen und Tutorien

Canvas und Moodle-Kurse

Die Organisation der Vorlesung erfolgt über

`https://aalen.instructure.com/courses/1533`.

Mittwochs erhalten Sie immer das neue Skript mit den zugehörigen Videolektionen, die im Skript verlinkt sind. Bearbeiten Sie das Material langsam, sorgfältig und gewissenhaft jede Woche.

Freiwillige Übungsblätter

In unregelmäßigen Abständen stellen wir Ihnen Aufgabenblätter zur Wiederholung des Stoffes zur Verfügung. Die Bearbeitung dieser Blätter ist freiwillig – aber dringend angeraten. Sie unterscheiden sich von den Tutorienblättern, die zum Erhalt der Prüfungszulassung bearbeitet werden müssen (siehe weiter unten).

Lesen Sie vor der Bearbeitung der Blätter den Artikel „Wie bearbeitet man ein Übungsblatt?“ von Prof. Dr. Manfred Lehn der Johannes Gutenberg-Universität Mainz sorgfältig durch. Er gibt einige wichtige Hinweise, wie an Hochschulen Übungsblätter zu bearbeiten sind (dies unterscheidet sich stark von den gewohnten Hausaufgaben an Schulen!).

Wir besprechen die Blätter in der Regel im Rahmen der Vorlesungen der folgenden Vorlesungswoche. Es ist nicht sinnvoll, sich bei der Besprechung nur „berieseln“ zu lassen; man sollte sich zuvor (intensiv) mit den Aufgaben auseinander gesetzt haben.

Tutorien und Tutorienblätter

Die Tutorien finden begleitend zur Vorlesung statt. Die Tutorien werden gemäß der geplanten Zeiten des Vorlesungsplans bis auf weiteres als Zoom/BigBlueButton-Meeting stattfinden. Die Tutorien finden (voraussichtlich) in 3 Gruppen statt. Sie müssen sich in eine Gruppe eintragen (siehe Gruppeneinteilung im unten verlinkten Canvas-Kurs):

- montags von 08:00 – 09:30 Uhr,
- dienstags von 08:00 – 09:30 Uhr,
- dienstags von 09:45 – 11:15 Uhr.

Zur Organisation hierzu verwenden wir den Kurs

`https://aalen.instructure.com/courses/221/`.

Den Einladelink zum Kurs finden Sie in den Ankündigungen im Vorlesungs-Canvas-Kurs. Herr Bernd Oder vom Grundlagenzentrum ist der Ansprechpartner für die Verwaltung der Tutorien. Sie können sich ab dem 22. 04. 2020 um 18:00 Uhr in eine der drei Tutoriumsgruppen eintragen. Um eine ausgewogene Betreuungsrelation zu gewährleisten, ist die Teilnehmendenzahl pro Gruppe auf ~25 Studierende begrenzt. Bis Mittwoch, den 29. 04. 2020 können Sie noch selber die Gruppe wechseln. Danach ist ein Wechsel aus organisatorischen Gründen nicht mehr möglich.

Ablauf. Die Tutorien starten ab dem 27. 04. 2020 mit einer Einführung in den organisatorischen und technischen Ablauf, bei dem insbesondere auch ihre Fragen geklärt werden.

Jeweils donnerstags wird ein Aufgabenblatt zum Vorlesungsstoff im Canvas-Tutorienkurs bereitgestellt. Sie haben dann bis zum Freitag der darauffolgende Woche um 12:00 Uhr Zeit, die Aufgaben zu bearbeiten und Ihre schriflichen Lösungen unter Diskussionen (https://aalen.instructure.com/courses/221/discussion_topics) hochzuladen bzw. auch Fragen dazu im Laufe der Woche zu posten.[†] Dafür wird es jede Woche einen neuen Thread geben.

Hilfe bei Verständnisproblemen. Wir wollen Sie natürlich beim Bewältigen des Stoffes nicht alleine lassen! Bitte versuchen Sie, die Fragen Ihrer Mitstudierenden in Canvas zu beantworten und sich gegenseitig zu helfen. Probleme, die nicht gelöst werden konnten, werden gemeinsam im Tutorium besprochen. Der Austausch zwischen Ihnen ist sehr gewünscht. Helfen Sie Ihren Mitstudierenden, so hat das nicht nur für die Fragestellenden Vorteile, sondern auch für Sie – Sie festigen Ihr Wissen und das ist Voraussetzung für eine erfolgreiche Klausur.

Voraussetzungen zur Teilnahme an der Klausur. Die Teilnahme an der GdM-Klausur ist nur möglich, wenn Sie einen „Übungsschein“ erwerben.

1. Sie können wöchentlich schriftliche Lösungen zum aktuellen Übungsblatt fristgerecht in der dazugehörigen Diskussion einreichen sowie am dazugehörigen Tutoriumstermin teilnehmen. Stimmen eingereichte Lösungen einer Aufgabe bei mehreren Studierenden 1:1 überein, so wird diese Aufgabe bei *keinem* dieser Studierenden gewertet.
2. Gemäß Modulhandbuch ist zusätzlich ein Zwischentest in der Mitte der Vorlesungszeit zu schreiben, der in die Wertung für den Übungsschein eingeht. Die Hochschulleitung klärt momentan ob und wie Prüfungen im online-Format durchgeführt werden können. Da aufgrund der aktuellen Corona-Situation gegenwärtig noch keine sichere Planung möglich ist, werden wir Sie in den nächsten Wochen informieren, ob ein Zwischentest durchgeführt wird. Wir bitten um Verständnis.

Die Vergabe des Übungsscheins erfolgt nach folgenden Kriterien: Ist a der Anteil der Punkte, die Sie im Zwischentest erreicht haben und b der Anteil der gemäß Punkt 1 gewerteten Aufgaben, dann erhalten Sie einen Übungsschein, falls

$$\frac{1}{4}(3a + b) \geq 60\,\%.$$

[†] Das erste Aufgabenblatt erscheint am Donnerstag, den 23. 04. 2020. Achtung: Wegen des Maifeiertags haben Sie abweichend nur bis zum Donnerstag, den 30. 04. 2020 um 12:00 Uhr Zeit, Ihre schriftlichen Lösungen einzureichen.

Falls kein Zwischentest durchgeführt werden kann, behält sich der Studiengang vor, den Übungsschein auch ohne Kurztest zu vergeben, falls

$$b \geq 60\,\%$$

ist. Wir werden Sie rechtzeitig über die finale Entscheidung informieren.

Die asynchrone Begrüßung zum Kurs findet am Mittwoch, den 22. April 2020 um 14:00 Uhr über eine YouTube Premiere statt. Das Video ist abrufbar unter

`https://youtu.be/GbyCCfbxP4Y`.

Inhaltsverzeichnis

1 Logik

Logik ist die *Sprache der Mathematik*. Sie dient einerseits dazu, mathematische Objekte sowie deren Eigenschaften und Beziehungen klar zu beschreiben, und ermöglicht andererseits eine verbindliche und missverständnisfreie Argumentation über diese Eigenschaften und Beziehungen, die – bei extrem penibler Durchführung – sogar ein Computer algorithmisch nachprüfen kann.

Das rote ▶-Symbol in der Randspalte kennzeichnet die zugehörigen Vorlesungsvideos zum behandelten Thema (das Symbol ist anklickbar!). Skript und Videos ergänzen sich dabei gegenseitig: Nicht alles findet sich im Video, und nicht alles im Video Gesagte ist im Skript abgetippt. Die Playlist ist verfügbar unter `https://bit.ly/GdM-Playlist`.

Grundelemente der Logik sind *Aussagen*. Dies sind Sätze, die wahr oder falsch sind. Beispiele sind

$$A \quad = \quad \text{“3 ist Primzahl”}$$
$$B \quad = \quad \text{“4 ist Primzahl”}$$

▶ Grundlagen Mathematik | 01.01: Einführung Logik – Was sind Aussagen in der Logik? Was sind keine?

Von der mathematischen Bedeutung der Aussagen wissen wir, dass A wahr ist und B falsch. Dagegen sind zum Beispiel Fragesätze (“Wieviel Uhr ist es?”) oder Aufforderungen (“Sagen Sie mir bitte die Uhrzeit”) keine Aussagen.

Weitere Beispiele und Gegenbeispiele wurden im Vorlesungsvideo gegeben. Denken Sie daran: Video und Skript ergänzen sich in beide Richtungen (ohne, dass dies notwendigerweise immer erwähnt wird)!

1.1 Formeln

Aussagen können zu neuen, komplexeren Sätzen verbunden werden, wie

A und B,
A oder B,
entweder A oder B,
wenn A dann B,
nicht A.

Solche zusammengesetzten Aussagen heißen *(logische) Formeln*, oder, zu Ehren von George Boole, auch *Boolesche Formeln*. Die Verknüpfungen *und*, *oder*, . . . heißen *Junktoren* oder *Konnektoren*. In der Logik wollen wir solchen Formeln ebenfalls einen Wahrheitwert, *wahr* oder *falsch*, zuordnen. Dieser hängt natürlich von den Wahrheitswerten der beteiligten Aussagen ab. Wir definieren den Wahrheitswert einer Formel mit einer *Wahrheitstabelle*, in die wir für jede mögliche Kombination der Wahrheitswerte der einzelnen Aussagen den Wahrheitswert der Formel angeben. Zur Abkürzung ersetzen wir dabei den Wahrheitswert wahr durch 1 und falsch durch 0.

Grundlagen Mathematik | 01.02: Wahrheitstabelle Negation, Konjunktion, Disjunktion, XOR, Implikation

Negation

Die Verneinung von Aussage A ist *nicht* A. Die Abkürzung dafür ist $\neg A$ oder $\overline{A}$. Wir definieren $\overline{A}$ als wahr, wenn A falsch ist, und als falsch, wenn A wahr ist. Damit erhalten wir folgende Wahrheitstabelle für die Negation:

A	$\neg A$
0	1
1	0

In der linken Spalte sind die beiden möglichen Wahrheitswerte der Aussage A. Man sagt dazu auch die beiden *(Wahrheits-) Belegungen* der *Aussage-Variablen* A. In der rechten Spalte sind die entsprechenden Wahrheitwerte von $\neg A$ angegeben.

Konjunktion

Die *und*-Verknüpfung A *und* B wird auch als *Konjunktion* bezeichnet und mit $A \wedge B$ abgekürzt. Wir definieren $A \wedge B$ als wahr, wenn beide Aussagen, A und B, wahr sind. Bei zwei Aussagen gibt es 4 Kombinationen der Wahrheitwerte. Deshalb hat die Wahrheitstabelle nun 4 Einträge:

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

Disjunktion und exklusive Disjunktion

Die *oder*-Verknüpfung A *oder* B wird auch als *Disjunktion* bezeichnet und mit $A \vee B$ abgekürzt. Wir definieren $A \vee B$ als wahr, wenn mindestens eine der Aussagen, A oder B, wahr ist.

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

Sind also beide Aussagen A, B wahr, dann ist $A \vee B$ als wahr definiert ist. Soll dieser Fall ausgeschlossen werden, sagen wir *entweder* A *oder* B. Diese Verknüpfung wird als *ausschließliches oder* bzw. *exklusives oder* bezeichnet und mit $A \oplus B$ abgekürzt. $A \oplus B$ ist also wahr, wenn genau eine der beiden Aussagen, A oder B, wahr ist.

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Implikation und Äquivalenz

Die *wenn-dann*-Verknüpfung *wenn A dann* B wird auch als *Implikation* bezeichnet und mit $A \to B$ abgekürzt. Die Aussage vor dem Pfeil, A, nennt man auch die *Voraussetzung*, und die Aussage nach dem Pfeil, B, die *Folgerung*. Wir definieren $A \to B$ in folgenden Fällen als wahr: Ist die Voraussetzung A wahr, dann muss auch die Folgerung B wahr sein. Ist die Voraussetzung A dagegen falsch, dann ist $A \to B$ wahr, unabhängig von B. Diese letzte Festlegung ist sinnvoll, da die Implikation ja lediglich sagen soll, dass B wahr ist, wenn A wahr ist. Es soll aber nichts über B ausgesagt werden, wenn die Voraussetzung A gar nicht zutrifft. Als Wahrheitstabelle erhalten wir:

A	B	$A \to B$
0	0	1
0	1	1
1	0	0
1	1	1

Die Implikation $A \to B$ ist also nur dann falsch, wenn A wahr ist und B falsch.

Nehmen wir als Beispiel A = "es regnet" und B = "wir bleiben zu Hause". Dann ist $A \to B$ = "wenn es regnet dann bleiben wir zu Hause". Betrachten wir den Wahrheitswert:

- Wenn es tatsächlich regnet und die Voraussetzung damit wahr ist, dann ist $A \to B$ wahr wenn wir zu Hause bleiben, und falsch wenn wir ausgehen.
- Wenn es nicht regnet und die Voraussetzung damit falsch ist, dann ist $A \to B$ wahr, egal ob wir zu Hause bleiben oder ausgehen. Darüber, was wir machen wenn es nicht regnet, haben wir ja nichts ausgesagt.

Am nächsten Tag erzählen wir jemand ob wir ausgegangen sind oder nicht. Was kann derjenige über das Wetter schließen wenn $A \to B$ wahr war?

- Sind wir ausgegangen, dann hat es folglich nicht geregnet.
- Sind wir dagegen zu Hause geblieben, dann kann man nicht sagen, ob es geregnet hat oder nicht, beide Fälle sind möglich.

Will man nun zusätzlich ausdrücken, dass wir nur zu Hause bleiben wenn es regnet und ansonsten ausgehen, dann benützt man die *genau-dann-wenn*-Verknüpfung *genau wenn A dann B*. Sie wird auch als *Äquivalenz* bezeichnet und mit $A \leftrightarrow B$ abgekürzt. Die Wahrheitstabelle ist wie folgt definiert:

A	B	$A \leftrightarrow B$
0	0	1
0	1	0
1	0	0
1	1	1

In unserem Beispiel haben wir $A \leftrightarrow B$ = "genau dann wenn es regnet bleiben wir zu Hause". Im Unterschied zur Implikation können wir nun nicht zu Hause bleiben wenn es nicht regnet.

Bei der Äquivalenz sind die beiden Aussagen A und B in gewisser Weise gleichwertig: Wenn $A \leftrightarrow B$ wahr ist sind entweder beide Aussagen wahr oder beide falsch.

Längere Formeln

Grundlagen Mathematik | 01.03: Ausfüllen Wahrheitstabelle zusammengesetzte Boolesche Formel

Damit können wir auch größere Formeln nach dem Baukasten-Prinzip konstruieren: sind F und G bereits Formeln, dann sind auch $\overline{F}$, $F \wedge G$, $F \vee G$, $F \oplus G$, $F \rightarrow G$ und $F \leftrightarrow G$ Formeln, wobei wir eventuell noch Klammern setzen müssen. Ein Beispiel: Starten wir mit den Variablen A, B und C, dann können wir $F_1 = \overline{A}$ bilden. Dann verknüpfen wir F_1 mit B zu $F_2 = \overline{A} \wedge B$. Sei $F_3 = A \rightarrow C$. Wir verknüpfen F_2 und F_3 zu $F_4 = F_2 \oplus F_3 = (\overline{A} \wedge B) \oplus (A \rightarrow C)$ usw.

Wir verwenden die Konvention, dass $\neg$ stärker als alle anderen Konnektoren bindet. Das hilft uns, Klammern zu sparen.

Bei solchen größeren Formeln sieht man nicht mehr direkt ob sie wahr oder falsch sind wenn man die Wahrheitswerte der beteiligten Aussagen kennt. Mithilfe der Wahrheittabellen können wir den Wahrheitswert jetzt aber ausrechnen. Betrachten wir z.,B. obige Formel F_4. Es sind drei Aussagen beteiligt. Dafür gibt es 8 Kombinationen der Wahrheitwerte. Gemäß dem Aufbau von F_4 bestimmen wir der Reihe nach die Wahrheitswerte von F_1, F_2, F_3, F_4.

A	B	C	$F_1 = \overline{A}$	$F_2 = \overline{A} \wedge B$	$F_3 = A \rightarrow C$	$F_4 = F_2 \oplus F_3$
0	0	0	1	0	1	1
0	0	1	1	0	1	1
0	1	0	1	1	1	0
0	1	1	1	1	1	0
1	0	0	0	0	0	0
1	0	1	0	0	1	1
1	1	0	0	0	0	0
1	1	1	0	0	1	1

Nun können wir den Wahrheitswert von F_4 ablesen. Sind z. B. alle drei Aussagen A, B, C falsch, dann ist F_4 wahr.

Erfüllbare Formeln und Tautologien

Grundlagen Mathematik | 01.04: Erfüllbare Boolesche Formel, Unerfüllbare logische Formel, Tautologie

Die folgenden Begriffe klassifizieren Boolesche Formeln nach Eigenschaften ihrer Wahrheitstabelle.

Ein Formel F heißt *erfüllbar*, wenn es mindestens eine Belegung ihrer Variablen gibt, so dass F wahr wird. D. h. erfüllbare Formeln haben mindestens eine 1 in ihrer Wahrheitstabelle. Z. B. ist die Formel $F = A \wedge \overline{B}$ erfüllbar, da für $A = 1$ und $B = 0$ die Formel F wahr ist.

Man fasst die erfüllbaren Booleschen Formeln oft zu einer Menge zusammen, SAT genannt. Wir lernen Mengen in Kapitel 2 kennen. Ist eine Formel F erfüllbar, kann man dann auch $F \in \text{SAT}$ schreiben.

Ist eine Formel nicht erfüllbar, so nennt man sie auch *unerfüllbar*. Ein Beispiel für eine unerfüllbare Formel ist $A \wedge \overline{A}$.

Hierfür wird auch $F \in \text{UNSAT}$ geschrieben.

Eine Formel, die bei jeder Belegung ihrer Variablen wahr ist, nennt man *gültig* oder eine *Tautologie*. D. h. dass in allen Zeilen der Wahrheitstabelle eine 1 steht. Ein Beispiel für eine Tautologie ist $A \vee \overline{A}$.

Die Menge der Tautologien erhält den Bezeichner TAUT.

Tautologien und unerfüllbare Formeln sind Negationen voneinander: F ist Tautologie genau dann wenn $\overline{F}$ unerfüllbar ist.

Beweise und Beweisprinzipien. Wir haben den Begriff der Tautologie eingeführt, da Tautologien beim Beweisen eine zentrale Rolle zukommt. Folgende Tautologien sind dabei so zentral, dass sie einen eigenen Namen erhalten:

Die folgenden Ausführungen eignen sich am besten für ein ruhiges, konzentriertes Selbststudium ohne Videos. Kommen Sie ruhig im Semesterverlauf immer wieder zum Nachschlagen zu den Beweisprinzipien zurück. Um Beweise kennen zu lernen, gehen Sie diese langsam, Schritt für Schritt, Zeile für Zeile durch. Jedes Wort ist wichtig! Überspringen Sie nichts! Am Ende des Abschnitts hole ich Sie wieder mit Videos ab. Lesen Sie immer Skript *und* schauen Sie die Videos. Beides ergänzt sich gegenseitig. In einer erstmaligen Lektüre dieses Skripts können Sie jedoch die nachfolgenden Ausführungen bis Abschnitt 1.2 überspringen und später darauf zurückkommen.

- Direkter Beweis (*modus ponens*):

$$\big(A \wedge (A \to B)\big) \to B.$$

- Widerspruchsbeweis (*reductio ad absurdum*):

$$\big((\neg A \to B) \wedge \neg B\big) \to A.$$

- *Fallunterscheidung*:

$$\big((A \to B) \wedge (\neg A \to B)\big) \to B.$$

- *Doppelte Negation*:

$$\neg\neg A \leftrightarrow A.$$

Siehe auch Gleichung (1.2).

- *Kontraposition*:

$$(A \to B) \leftrightarrow (\neg B \to \neg A).$$

Die Kontraposition werden wir in Abschnitt 1.2 formal herleiten.

- *Äquivalenz durch Ringschluss*:

$$\big((A \to B) \wedge (B \to A)\big) \leftrightarrow (A \leftrightarrow B).$$

Der Ringschluss lässt sich leicht auf mehrere Variablen anwenden (so wird er auch am häufigsten verwendet): Um etwa die Äquivalenz von drei Aussagen A, B, C zu zeigen, also um zu zeigen, dass $A \leftrightarrow B \leftrightarrow C$ eine Tautologie ist, genügt es z. B. zu zeigen, dass $(A \to B) \wedge (B \to C) \wedge (C \to A)$ gültig ist.

- Prinzip des ausgeschlossenen Dritten (*tertium non datur*):

$$A \vee \neg A.$$

Warum sind wir nun aber so an Tautologien interessiert? In Beweisen können diese immer benutzt werden – sie sind ja schon aus „logischen Gründen“ wahr, also ohne, dass wir spezielle Voraussetzungen oder Eigenschaften von speziellen Objekten benutzen müssen. So erlauben wir z. B. den modus ponens als Deduktionsregel, d. h., wenn A gilt und $A \to B$ gilt, dann dürfen wir B schlussfolgern.

Im Video über Prädikatenlogik werden wir später noch das Generalisierungs-Konzept besprechen, das uns hilft Aussagen der Form „für alle x gilt die Aussage $\varphi(x)$“ zu zeigen.

Nachdem wir dies besprochen haben, können wir den Begriff des Beweises formalisieren.*

Definition 1.1.1. Ein *Beweis* ist eine endliche Abfolge von (prädikaten-)logischen Formeln. Jede dieser Formeln ist dabei entweder eine gegebene Voraussetzung oder eine Annahme, ein gegebenes Axiom, eine logische Tautologie, oder eine Formel, die sich durch modus ponens (oder Generalisierung) aus den zuvor aufgelisteten Formeln ableiten lässt. Die letzte Formel der Liste ist dann die unter den Voraussetzungen bewiesene Formel.

Aktuell kennen wir nur logische Formeln. Wir werden prädikatenlogische Formeln erst später kennen lernen.

* Dies ist eine etwas grobe Beschreibung des *Hilbert-Kalküls*, siehe z. B. https://de.wikipedia.org/wiki/Hilbert-Kalk%C3%BCl. Bei späteren Beweisen, werden wir aber deutlich weniger formal vorgehen (da unsere Beweise nicht computerlesbar sein müssen), d. h. Sachverhalte werden oft textuell oder schematisch beschrieben, ohne genaue aussagenlogische Formeln anzugeben. Dabei achten wir allerdings auf Schlüssigkeit und Lückenlosigkeit in unserer Argumentationskette! Unsere Beweise müssen sich jederzeit solange expandieren lassen, bis man bei einem ganz formalen Beweis im Sinne der Definition 1.1.1 ankommt!

Beispiel 1.1.2 (Direkter Beweis)**.** Wir wollen zeigen, dass für alle natürlichen Zahlen n gilt: Wenn n gerade ist, dann ist n^2 gerade.

Implizit benutzen wir hier sogar bereits Generalisierung.

Sei also n eine beliebige gerade natürliche Zahl. Da n gerade ist, muss es eine natürliche Zahl k geben, sodass wir $n = 2k$ schreiben können. Dann ist aber $n^2 = (2k)^2 = 4k^2 = 2 \cdot 2k^2$. Da sich n^2 in dieser Form schreiben lässt, ist n^2 eine gerade natürliche Zahl. Weil wir n zu Beginn beliebig gewählt haben, gilt unser Beweis für alle natürlichen Zahlen n.

Beispiel 1.1.3 (Beweis über Kontraposition)**.** Wir wollen zeigen, dass für alle natürlichen Zahlen n gilt: Wenn n^2 gerade ist, dann ist n gerade.

Dies zeigen wir, indem wir stattdessen die Kontraposition zeigen. Dazu sei eine beliebige natürliche Zahl n fixiert. Weiter sei $A =$ " n^2 ist gerade" und $B =$ " die fixierte Zahl n ist gerade". Anstatt nun $A \implies B$ zu zeigen, zeigen wir $\neg B \implies \neg A$. Wir zeigen also: Wenn n ungerade ist, dann ist n^2 auch ungerade. Dies ist nun wieder ein direkter Beweis. Man erhält ihn leicht, indem man Beispiel 1.1.2 modifiziert. Dies sei dem Leser zur Übung überlassen!

Beispiel 1.1.4 (Widerspruchsbeweis)**.** Wir wollen zeigen, dass $\sqrt{2}$ irrational ist. Die Beweisführung erfolgt nach der Methode des Widerspruchsbeweises, das heißt, es wird gezeigt, dass die Annahme, die Wurzel aus 2 sei eine rationale Zahl, zu einem Widerspruch führt (reductio ad absurdum).

Pedantischer Weise sei angemerkt, dass wir weder definiert haben, was eine Wurzel einer Zahl a ist (diejenige nichtnegative Zahl x für die $x^2 = a$ gilt), noch, was irrational (nicht als Bruch darstellbar) bedeutet. An dieser Stelle ziehen wir uns gelegentlich noch auf Schulstoff zurück, um bereits jetzt anschauliche Beispiele bieten zu können.

Nehmen wir also an, dass die Quadratwurzel aus 2 rational ist und sich somit als Bruch $\frac{p}{q}$ darstellen lässt (mit rationalen Zahlen p, q und $q \neq 0$). Wir können ferner ohne Beschränkung der Allgemeinheit annehmen, dass p und q teilerfremde ganze Zahlen sind, der Bruch $\frac{p}{q}$ also in gekürzter Form vorliegt:

$$\sqrt{2} = \frac{p}{q}.$$

Also erhalten wir durch Quadrieren

$$2 = \left(\frac{p}{q}\right)^2,$$

oder umgeformt

$$p^2 = 2q^2.$$

Da $2q^2$ eine gerade Zahl ist, ist auch p^2 gerade. Mit Beispiel 1.1.3 folgt, dass auch die Zahl p gerade ist. Also lässt sie sich darstellen durch

$$p = 2a,$$

wobei a eine ganze Zahl ist. Damit erhält man mit obiger Gleichung

$$2q^2 = p^2 = (2a)^2 = 4a^2.$$

Nach Division durch 2 folgt

$$q^2 = 2a^2.$$

Mit der gleichen Argumentation wie zuvor folgt, dass q^2 und damit auch q eine gerade Zahl ist. Nun haben wir aber herausgefunden, dass p und q durch 2 teilbar sind, d. h. der Bruch $\frac{p}{q}$ lag gar nicht in gekürzter Form dar. Das ist ein Widerspruch!

Dieser Widerspruch zeigt, dass die Annahme, die Wurzel aus 2 sei eine rationale Zahl, falsch ist und daher das Gegenteil gelten muss. Damit ist die Behauptung, dass $\sqrt{2}$ irrational ist, bewiesen.

Zur Belohnung, dass Sie diesen Beweis durchgearbeitet haben, können Sie sich nun unter `https://youtu.be/tPfnEByx9r0` den Song zum Beweis vom Mathe-Youtuber DorFuchs anschauen – ein viraler Hit zu meinen Studiumszeiten.

1.2 Äquivalenz von Formeln

Grundlagen Mathematik | 01.05: Äquivalenz von Booleschen Formeln, Wahrheitstabellen Vergleich, gdw

Manche Formeln sind zwar (syntaktisch) verschieden, d. h. wenn man sie Zeichen für Zeichen von links nach rechts vergleicht stimmen sie nicht überein, aber sie haben immer den gleichen Wahrheitswert, egal welche Wahrheitswerte die beteiligten Aussagen auch haben. Zum Beispiel ist intuitiv klar, dass es keinen Unterschied macht, ob man $A \vee B$ oder $B \vee A$ sagt. Auch formal kann man das leicht einsehen indem man die Wahrheitstabellen der beiden Formeln vergleicht. Die von $A \vee B$ ist oben angegeben. Die von $B \vee A$ erhält man, indem B die Rolle von A in $A \vee B$ übernimmt und A die Rolle von B. Dann ergibt sich diesselbe Wahrheitstabelle. Man nennt solche Formeln *äquivalent* und drückt dies durch die Schreibweise $A \vee B \equiv B \vee A$ aus.

Definition 1.2.1. Zwei Formeln F und G heißen *äquivalent*, schreibe $F \equiv G$, wenn sie bei jeder Belegung der Aussage-Variablen den gleichen Wahrheitswert haben.

Sie haben sicher schon bemerkt: Begriffe, die wir neu einführen oder definieren sind kursiv gedruckt.

Die Äquivalenz von Formeln können wir auch mit der *genau-dann-wenn* Verknüpfung ausdrücken. $F \equiv G$ heißt ja nichts anderes als dass F genau dann wahr ist, wenn G wahr ist. Also: $F \equiv G$ genau dann, wenn $F \leftrightarrow G$ Tautologie ist.

Das *genau-dann-wenn*, so wie wir es gerade benützt haben, macht eine Aussage *über* Formeln. Wir stellen uns gewissermaßen eine Stufe höher (auf eine *Metaebene*) und benützen die Logik um über Formeln zu sprechen. Wir verwenden dabei zwar die logischen Verknüpfungen *und, oder, nicht, entweder oder, wenn-dann, genau-dann-wenn*, und zwar genau im oben definierten Sinn, aber eben nicht die dafür innerhalb von Formeln verwendeten Zeichen $\wedge, \vee, \neg, \oplus, \rightarrow, \leftrightarrow$. Stattdessen belassen wir es meistens bei der verbalen Formulierung. Anstatt $F \equiv G$ kann man also auch sagen: *F genau dann, wenn G* (das "ist wahr" lässt man dabei weg). Da "genau dann wenn" viel verwendet wird und eben doch ein bisschen lang ist (immerhin 13 Buchstaben!), hat man dafür Abkürzungen. Gelegentlich sieht man *gdw.*, die am häufigsten verwendete Abkürzung ist aber wohl $\Longleftrightarrow$. Die hat den Vorteil, dass sie an den Junktor $\leftrightarrow$ erinnert. Anstatt $F \equiv G$ kann man also auch $F \Longleftrightarrow G$ schreiben. Entsprechend wird die Formulierung "wenn dann" durch $\Longrightarrow$ abgekürzt.

Grundlagen Mathematik | 01.06: Kommutativgesetze Boolesche Formeln erklärt, Implikation kommutativ?

Kommutativgesetze

Im obigen Beispiel wurde gezeigt, dass man die Reihenfolge der Variablen bei einer oder-Verknüpfung vertauschen kann. Man sagt, dass die oder-Verknüpfung *kommutativ* ist. Dasselbe gilt auch für die und-Verknüpfung, das exklusive Oder und die genau-dann-wenn-Verknüpfung:

$$\begin{aligned} A \vee B &\equiv B \vee A, \\ A \wedge B &\equiv B \wedge A, \\ A \oplus B &\equiv B \oplus A, \\ A \leftrightarrow B &\equiv B \leftrightarrow A. \end{aligned}$$

Die Implikation ist dagegen nicht kommutativ. Vergleichen wir die beiden Wahrheitstabellen miteinander:

A	B	$A \to B$	$B \to A$
0	0	1	1
0	1	1	0
1	0	0	1
1	1	1	1

Da die beiden letzten Spalten nicht übereinstimmen gilt

$$A \to B \not\equiv B \to A.$$

Grundlagen Mathematik | 01.07: Assoziativgesetz für Boolesche Formel mit Wahrheitstabelle beweisen

Assoziativgesetze

Sollen drei Aussagevariablen mit oder verknüpft werden, so können wir (zunächst) nicht einfach $A \vee B \vee C$ schreiben, da so nicht klar ist, welche der beiden oder-Verknüpfungen zuerst angewendet werden soll. Deshalb setzen wir Klammern. Hier gibt es zwei mögliche Klammerungen,

$$(A \vee B) \vee C \qquad \text{oder} \qquad A \vee (B \vee C).$$

Wenn wir allerdings die Wahrheitstabelle der beiden Formeln ansehen, stellen wir fest, dass sie äquivalent sind.

A	B	C	$A \vee B$	$(A \vee B) \vee C$	$B \vee C$	$A \vee (B \vee C)$
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	1	1	1	1
0	1	1	1	1	1	1
1	0	0	1	1	0	1
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

D. h. bei der oder-Verknüpfung spielt es keine Rolle, wie die Klammern gesetzt werden, sie ist *assoziativ*. Deshalb lässt man in der Regel die Klammern einfach weg und schreibt eben doch $A \vee B \vee C$. Analog überzeugt man sich, dass die und-Verknüpfung, das exklusive Oder und

die genau-dann-wenn-Verknüpfung assoziativ sind:

$$\begin{aligned}(A \vee B) \vee C &\equiv A \vee (B \vee C),\\ (A \wedge B) \wedge C &\equiv A \wedge (B \wedge C),\\ (A \oplus B) \oplus C &\equiv A \oplus (B \oplus C),\\ (A \leftrightarrow B) \leftrightarrow C &\equiv A \leftrightarrow (B \leftrightarrow C).\end{aligned}$$

Die Implikation ist dagegen nicht assoziativ:

A	B	C	$A \to B$	$(A \to B) \to C$	$B \to C$	$A \to (B \to C)$
0	0	0	1	0	1	1
0	0	1	1	1	1	1
0	1	0	1	0	0	1
0	1	1	1	1	1	1
1	0	0	0	1	1	1
1	0	1	0	1	1	1
1	1	0	1	0	0	0
1	1	1	1	1	1	1

Grundlagen Mathematik | 01.08: Ist die Implikation für Boolesche Formeln assoziativ? Beweis erklärt

Distributivgesetze

Grundlagen Mathematik | 01.09: Distributivgesetze Logik und Boolesche Formeln – Und, Oder, XOR

Kommen verschiedene logische Verknüpfungen in einer Formel vor, dann kann man auf Klammern nicht mehr verzichten. Die Formel $A \wedge (B \vee C)$ kann man so umschreiben, dass man A in die Klammer bringt und mit B und C und-verknüpft. Also

$$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C),$$

wie man leicht durch eine Wahrheitstabelle überprüfen kann. Dies ist analog wie in der Arithmetik $a \cdot (b + c) = a \cdot b + b \cdot c$ ist.

In der Arithmetik darf man die Operationen nicht vertauschen, $a + (b \cdot c)$ ist im Allgemeinen nicht das gleiche wie $(a+b)\cdot(a+c)$. Die Operationen $\wedge$ und $\vee$ darf man dagegen schon vertauschen, es gilt auch

$$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C).$$

Wir werden später noch sehen, dass das $\wedge$ einer Multiplikation entspricht, aber bei $\vee$ und der Addition gibt es gewisse Unterschiede, und die zeigen sich zum Beispiel hier beim Distributivgesetz.

Dagegen entspricht $\oplus$ einer Addition, wie wir noch sehen werden. Wenn wir also $\wedge$ und $\oplus$ betrachten, erhalten wir analog wie in der Arithmetik nur ein Distributivgesetz

$$A \wedge (B \oplus C) \equiv (A \wedge B) \oplus (A \wedge C),$$

die Formeln $A \oplus (B \wedge C)$ und $(A \oplus B) \wedge (A \oplus C)$ sind dagegen nicht äquivalent wie man anhand ihrer Wahrheitstabelle überprüfen kann.

Grundlagen Mathematik | 01.10: Negation, deMorgansche Regeln, Doppelnegation

Negation und deMorgansche Gesetze

Negiert man eine Formel F, betrachtet also $\overline{F}$, dann kann man mit Hilfe der deMorganschen Gesetze die Negation nach innen zu den einzelnen Aussagen bringen:

$$\begin{aligned} \overline{A \vee B} &\equiv \overline{A} \wedge \overline{B}, \\ \overline{A \wedge B} &\equiv \overline{A} \vee \overline{B}. \end{aligned}$$

Umgangsprachlich setzt man keine Klammern. Bei "nicht (ich bleibe zu Hause oder gehe ins Kino)" müsste man die Klammern aber mitsprechen. Um das zu umgehen zieht man die Negation nach innen und sagt "ich bleibe nicht zu Hause und gehe nicht ins Kino", oder gleichwertig "ich bleibe weder zu Hause, noch gehe ich ins Kino".

Anders verhält sich das exklusive oder. Negiert man beide Aussagen in $A \oplus B$, so ändert sich der Wahrheitwert nicht,

$$\overline{A} \oplus \overline{B} \equiv A \oplus B.$$

Negiert man dagegen die ganze Formel $A \oplus B$, so wird daraus eine Äquivalenz,

$$\overline{A \oplus B} \equiv A \leftrightarrow B. \tag{1.1}$$

Manchen Formeln geben wir eine Nummer, damit wir diese später referenzieren können.

Negiert man eine Aussage oder Formel doppelt, so erhält man wieder dieselbe Aussage.

$$\overline{\overline{A}} \equiv A. \tag{1.2}$$

Grundlagen Mathematik | 01.11: Absorptionsregel in der Logik hergeleitet

Absorbtionsgesetze

Wir wollen die Formel $A \vee (A \wedge B)$ vereinfachen. Versuchen wir es mit dem Distributivgesetz:

$$\begin{aligned} A \vee (A \wedge B) &\equiv (A \vee A) \wedge (A \vee B) \\ &\equiv A \wedge (A \vee B). \end{aligned}$$

Wenn wir auf die so erhaltene Formel $A \wedge (A \vee B)$ nochmal das Distributivgesetz anwenden, so landen wir wieder bei der Ausgangsformel. Mit unseren seitherigen Rechenregeln lässt sich diese Formel nicht vereinfachen.

Schauen wir uns die Formel direkt an. Damit $A \wedge B$ wahr ist, müssen A und B wahr sein. Bei der oder-Verknüpfung mit A reicht es aber schon, wenn A alleine wahr ist. Es gilt also

$$A \vee (A \wedge B) \equiv A \wedge (A \vee B) \equiv A.$$

Grundlagen Mathematik | 01.12: Rechnen mit Konstanten in der Logik, Wdh. unerfüllbar und Tautologie

Rechnen mit Konstanten

Die kürzesten Formeln die wir seither gesehen haben bestehen nur aus einere Aussage, z. B. A. Es erweist sich als praktisch auch noch die Konstanten 0 und 1 als Formeln zuzulassen. Dabei steht 0 für eine

unerfüllbare Formel, und 1 für eine Tautologie. Wir können also schreiben $A \wedge \overline{A} \equiv 0$ und $A \vee \overline{A} \equiv 1$.

In einer Wahrheitstabelle wird mit 0 und 1 gerechnet indem die ganze Spalte mit 0 bzw. 1 ausgefüllt wird. Damit ergeben sich folgende Rechenregeln:

$$\begin{aligned} A \wedge 0 &\equiv 0, & A \wedge 1 &\equiv A, \\ A \vee 0 &\equiv A, & A \vee 1 &\equiv 1, \\ A \oplus 0 &\equiv A, & A \oplus 1 &\equiv \overline{A}. \end{aligned}$$

Weitere Gesetze

Grundlagen Mathematik | 01.13: Weitere Rechenregeln für Boolesche Formeln, Anwendung Rechenregeln

Oft ist es nützlich eine gegebene Formel umzuschreiben. So kann man beispielsweise eine Äquivalenz $A \leftrightarrow B$ mit Hilfe von Implikationen ausdrücken,

$$A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A). \tag{1.3}$$

Eine Implikation kann man wiederum durch eine Disjunktion ersetzen,

$$A \rightarrow B \equiv \overline{A} \vee B. \tag{1.4}$$

Nehmen wir nochmal unser früheres Beispiel $A =$ "es regnet" und $B =$ "wir bleiben zu Hause". Dann ist $A \rightarrow B =$ "wenn es regnet dann bleiben wir zu Hause" gleichwertig zu $\overline{A} \vee B =$ "es regnet nicht oder wir bleiben zu Hause".

Bis hierher haben wir Äquivalenzen immer durch ihre Wahrheitstabelle überprüft. Wir können aber auch die bereits als richtig bewiesenen Äquivalenzen benützen, um damit gegebene Formeln äquivalent umzuformen. Nehmen wir als Beispiel obige Formel $\overline{A} \vee B$. Wir negieren B doppelt, wenden das Kommutativgesetz an und dann Formel 1.4 für die Implikation:

$$\overline{A} \vee B \equiv \overline{A} \vee \overline{\overline{B}} \equiv \overline{\overline{B}} \vee \overline{A} \equiv \overline{B} \rightarrow \overline{A}.$$

Damit haben wir die Äquivalenz

$$A \rightarrow B \equiv \overline{B} \rightarrow \overline{A}. \tag{1.5}$$

Die Formel $\overline{B} \rightarrow \overline{A}$ wird als *Kontraposition von* $A \rightarrow B$ bezeichnet. In unserem obigen Beispiel könnten wir also auch sagen: "Wenn wir nicht zu Hause bleiben dann regnet es nicht".

Ersetzen wir in (1.3) die Implikationen durch Disjunktionen gemäß (1.4), so erhalten wir

$$A \leftrightarrow B \equiv (\overline{A} \vee B) \wedge (A \vee \overline{B}). \tag{1.6}$$

Wir können die rechte Seite weiter umformen indem wir das Distributivgesetz (zweifach) anwenden:

$$\begin{aligned} (\overline{A} \vee B) \wedge (A \vee \overline{B}) &\equiv \underbrace{(\overline{A} \wedge A)}_{\equiv 0} \vee (\overline{A} \wedge \overline{B}) \vee (B \wedge A) \vee \underbrace{(B \wedge \overline{B})}_{\equiv 0} \\ &\equiv (\overline{A} \wedge \overline{B}) \vee (B \wedge A) \\ &\equiv (A \wedge B) \vee (\overline{A} \wedge \overline{B}). \end{aligned}$$

Dabei wurde gemäß den Rechenregeln mit Konstanten die 0 weggelassen, da $F \vee 0 \equiv F$.

Damit haben wir eine weitere Darstellung der Äquivalenz,

$$A \leftrightarrow B \equiv (A \wedge B) \vee (\overline{A} \wedge \overline{B}). \tag{1.7}$$

Nach (1.1) ist $A \oplus B$ die Negation von $A \leftrightarrow B$. Mit den deMorganschen Gesetzen erhalten wir damit aus (1.6) und (1.7)

$$\begin{aligned} A \oplus B &\equiv (A \wedge \overline{B}) \vee (\overline{A} \wedge B) && (1.8)\\ &\equiv (A \vee B) \wedge (\overline{A} \vee \overline{B}). && (1.9) \end{aligned}$$

ToDos für Sie diese Woche

- In Canvas finden Sie unter den Modulen (siehe https://aalen.instructure.com/courses/1533/modules) bereits das erste freiwillige Übungsblatt. Bearbeiten Sie dies nach einem sorgfältigen Studium der Inhalte der ersten Vorlesungswoche. Wir werden das Blatt im Rahmen der Vorlesungswoche 2 besprechen. Sie bekommen dadurch bereits einen Einblick, was wir von Ihnen im Rahmen des Tutoriums erwarten.
- Tragen Sie sich außerdem in den Tutorienkurs ein: https://aalen.instructure.com/courses/221/.
- Stimmen Sie **bis Mittwoch, den 29. 04. 2020** für ein Tutorium ab!
- Am Donnerstag, den 22. April 2020 kommt das erste Tutorienübungsblatt online. Denken Sie auch an dessen Bearbeitung.

↓ **Ende der 1. Vorlesungswoche**

Besprechung Blatt 01

Schauen Sie sich die Besprechung des Freiwilligen Übungsblattes 01 an: https://youtu.be/lEj_DI_wZIg. Bei Fragen, nutzen Sie bitte gerne unsere Hilfsangebote (allen voran die Tutorien). Keine Frage ist dumm! Oft haben viele andere Studierende den gleichen Gedanken und trauen sich nur nicht nachzufragen. Diese sind Ihnen dann dankbar.

Grundlagen Mathematik | 01.14: Konjunktive und disjunktive Normalform aus einer Wahrheitstabelle

1.3 Normalformen

Im vorherigen Abschnitt sind wir von Formeln ausgegangen und haben deren Wahrheitstabellen betrachtet. Jetzt wollen wir den umgekehrten Schritt tun: wir starten mit einer Wahrheitstabelle und wollen eine dazu passende Formel angeben. Wir geben zwei Methoden an, die zu unterschiedlichen Darstellungen führen.

Disjunktive Normalform (DNF)

Betrachten wir ein Beispiel. Gegeben sei die Formel F mit der folgenden Wahrheitstabelle:

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Die Formel F ist also genau einmal wahr, und zwar wenn A und B falsch sind und C wahr ist. D. h. eine Formel für F ist

$$F = \overline{A} \wedge \overline{B} \wedge C.$$

Eine Aussagevariable oder ihre Negation bezeichnet man als *Literal*. Die Formel F ist also eine Konjunktion von Literalen.

Ist die gesuchte Formel an mehreren Stellen wahr, so können wir zunächst für jede Zeile mit Wert 1 eine entsprechende Konjunktion wie oben für F angeben. Wenn wir dann diese Konjunktionen mit *oder* verknüpfen, so erhalten wir genau die gewünschte Formel. Als Beispiel betrachten wir

A	B	C	G
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	1

Damit erhalten wir

$$G = (\overline{A} \wedge \overline{B} \wedge C) \vee (\overline{A} \wedge B \wedge C) \vee (A \wedge B \wedge C).$$

Eine Konjunktion von Literalen bezeichnet man als *und-Klausel* (auch *Minterm*). Die Formel G ist also eine Disjunktion von und-Klauseln, und damit eine Disjunktion von Konjunktionen von Literalen. Diese Darstellung von G bezeichnet man als *disjunktive Normalform (DNF) von G*.

Unser Vorgehen klappt natürlich analog bei jeder Anzahl von Einsen in der Tabelle und bei jeder Anzahl von Variablen. Damit haben wir das eingangs gestellte Problem also schon gelöst. Folglich lässt sich jede Formel F in disjunktive Normalform bringen: man erstellt zunächst die Wahrheitstabelle von F und daraus dann die disjunktive Normalform. Damit haben wir nachfolgendes Theorem bewiesen.

Theorem 1.3.1. *Zu jeder Formel gibt es eine äquivalente Formel in disjunktiver Normalform.*

Konjunktive Normalform (KNF)

Die disjunktive Normalform haben wir bekommen, indem wir angefangen haben die Einsen in der Wahrheitstabelle durch Konjunktionen auszudrücken. Man kann sich aber auch die Nullen zunütze machen. Betrachten wir ein Beispiel, in dem genau eine Zeile den Wahrheitswert falsch hat:

A	B	C	H
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

H ist also die Negation von F:

$$H \equiv \overline{F} \equiv \overline{\overline{A} \wedge \overline{B} \wedge C} \equiv A \vee B \vee \overline{C}.$$

Diese letzte für Formel H ist also eine Disjunktion von Literalen, eine *oder-Klausel* oder kurz *Klausel* (auch *Maxterm*). Man hätte diese Formel auch direkt angeben können: eine Disjunktion von Literalen ist bei genau einer Belegung falsch, nämlich dann, wenn alle Literale falsch sind. Dies ist hier genau dann der Fall, wenn A und B falsch sind und C wahr ist. Daraus ergibt sich ebenfalls die Darstellung $H \equiv A \vee B \vee \overline{C}$.

Ist die gesuchte Formel an mehreren Stellen falsch, so können wir zunächst für jede Zeile mit Wert 0 eine entsprechende Disjunktion wie für H angeben, und diese dann mit *und* verknüpfen. Für

A	B	C	K
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

erhalten wir also

$$K = (A \vee B \vee \overline{C}) \wedge (A \vee \overline{B} \vee \overline{C}) \wedge (\overline{A} \vee \overline{B} \vee \overline{C}).$$

Die Formel K ist also eine Konjunktion von oder-Klauseln, und damit eine Konjunktion von Disjunktionen von Literalen. Diese Darstellung von K bezeichnet man als *konjunktive Normalform (KNF) von K*.

Da unser Vorgehen wieder analog bei jeder Anzahl von Nullen in der Tabelle und bei jeder Anzahl von Variablen funktioniert. lässt sich folglich jede Formel auch in konjunktive Normalform bringen. Das heißt, wir haben auch das folgende Theorem bereits bewiesen.

Theorem 1.3.2. *Zu jeder Formel gibt es eine äquivalente Formel in konjunktiver Normalform.*

Der Weg über die Wahrheitstabelle ist nicht die einzige Möglichkeit um eine Formel F in DNF oder KNF zu bringen. Man kann F auch mit den Rechenregeln aus dem vorherigen Abschnitt passend umformen. Wir haben auch schon einige Beispiele gesehen. So ist nach Äquivalenz (1.4) $\overline{A} \vee B$ sowohl DNF wie auch KNF von $A \to B$. In Äquivalenz (1.6) haben wir die DNF von $A \leftrightarrow B$, und in (1.7) die KNF. In (1.8) und (1.9) haben wir KNF und DNF von $A \oplus B$. Man kann auch allgemein zeigen, dass man jede Formel allein durch Anwendung unserer Rechenregeln in DNF und in KNF bringen kann.

Vollständige Basen

Zu diesem Thema ist kein Video verfügbar. Der Abschnitt ist sehr leicht verständlich (falls nicht: stellen Sie Fragen). Hier können Sie einüben, sich ein kleines Thema selbstständig anzueignen – Sie werden das im Verlauf Ihres Studiums immer wieder tun müssen. Die Schreibweise mit geschweiften Klammern wird in Kapitel 2 klarer.

DNF und KNF benützen lediglich die drei Junktoren $\wedge, \vee, \neg$. Diese reichen also aus, um jede Formel oder Wahrheitstabelle äquivalent auszudrücken. Man sagt, $\{\wedge, \vee, \neg\}$ bildet eine *vollständige Basis*.

Nachdem wir anfangs eine ganze Reihe von Junktoren definiert haben und man natürlich leicht noch weitere definieren kann, ist es vielleicht doch einigermaßen überraschend, dass man mit lediglich 3 Junktoren jeden logischen Sachverhalt beschreiben kann. Und es reichen sogar zwei: Bereits $\{\wedge, \neg\}$ bildet eine vollständige Basis. Das liegt daran, dass man die „fehlende Operation" $\vee$ durch $\wedge$ und $\neg$ ausdrücken kann:

$$A \vee B \equiv \overline{\overline{A \vee B}} \equiv \overline{\overline{A} \wedge \overline{B}}.$$

D. h. in jeder Formel, die $\wedge, \vee, \neg$ benützt, lässt sich $\vee$ ersetzen und es bleibt nur noch $\wedge, \neg$ übrig. Analog kann man zeigen, dass auch $\vee, \neg$ eine vollständige Basis bilden.

Versuchen Sie dies zu zeigen!

Eine offensichtliche Frage ist nun, ob auch $\{\wedge, \vee\}$ eine vollständige Basis bildet. Dies ist allerdings nicht der Fall. Das kann man wie folgt einsehen. Wir interpretieren die beiden Wahrheitswerte 0 und 1 wie Zahlen und ordnen sie wie üblich durch $0 < 1$.

Der Beweis zur Beantwortung dieser Frage ist für interessierte Leser empfohlen. Er ist nicht klausurrelevant.

Mehr zu sogenannten Ordnungsrelationen erfahren wir in Kapitel 3.

Der Junktor $\vee$ verknüpft zwei Aussagen A und B miteinander. Halten wir die eine Aussage fest, sagen wir A, und für die zweite Aussage setzen wir zuerst den kleineren Wert 0 und danach den größeren Wert 1 ein. Dann gilt auch, dass der Wahrheitswert von $A \vee 0$ kleiner oder gleich dem Wahrheitswert von $A \vee 1$ ist, egal welchen Wahrheitswert A hat. Dasselbe gilt für den Vergleich von $0 \vee B$ mit $1 \vee B$. Man sagt, dass $\vee$ *monoton* ist.

Einen analoge Betrachtung zeigt, dass auch $\wedge$ monoton ist. Außerdem sieht man leicht ein, dass eine Formel, die aus lauter monotonen Junktoren aufgebaut ist, selbst auch wieder monoton ist. D. h. mittels $\wedge, \vee$ bekommt man ausschließlich monotone Formeln.

Im Gegensatz dazu ist die Negation $\neg$ nicht monoton. Wenn man in $\neg A$ den Wert von A von 0 auf 1 erhöht, dann verkleinert sich der Wahrheitswert von 1 auf 0. Folglich ist es nicht möglich $\neg$ durch $\wedge, \vee$ auszudrücken, und somit bildet $\{\wedge, \vee\}$ keine vollständige Basis.

Grundlagen Mathematik | 01.15: Sind Normalformen von Booleschen Formeln (KNF/DNF) vereinfachbar?

1.4 Minimierung der Normalformen

Wenn wir die Normalform so wie oben aus der Wahrheitstabelle erzeugen, dann kommt in jeder Klausel jede Variable (direkt oder negiert) genau einmal vor. In diesem Fall nennt man die Normalform auch *kanonische disjunktive* bzw. *kanonische konjunktive Normalform*. Diese sind eindeutig. Bei n Variablen hat die Wahrheitstabelle 2^n Zeilen. Besteht die kanonische DNF einer Formel aus m und-Klauseln, dann hat die kanonische KNF $2^n - m$ oder-Klauseln. Manchmal kann man Normalformen verkürzen. Dann sind sie natürlich nicht mehr kanonisch. Wir lernen im Folgenden zwei Verfahren dazu kennen.

Das Quine-McCluskey Verfahren

Nehmen wir als Beispiel obige Formel G in KNF. Die beiden letzten Klauseln von G kann man vereinfachen:

$$\begin{aligned}(A \vee \overline{B} \vee \overline{C}) \wedge (\overline{A} \vee \overline{B} \vee \overline{C}) &\equiv (A \wedge \overline{A}) \vee (\overline{B} \vee \overline{C}) \\ &\equiv \overline{B} \vee \overline{C}.\end{aligned}$$

Folglich erhalten wir für G die kürzere KNF

$$G \equiv (A \vee B \vee \overline{C}) \wedge (\overline{B} \vee \overline{C}).$$

Etwas allgemeiner können wir also Formeln in KNF oder DNF nach folgenden Regeln verkürzen: für Literale L_1, L_2 und eine Aussagevariable A gilt

$$(L_1 \vee L_2 \vee A) \wedge (L_1 \vee L_2 \vee \overline{A}) \equiv L_1 \vee L_2, \tag{1.10}$$

$$(L_1 \wedge L_2 \wedge A) \vee (L_1 \wedge L_2 \wedge \overline{A}) \equiv L_1 \wedge L_2. \tag{1.11}$$

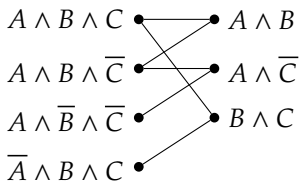
Grundlagen Mathematik | 01.16: Einfaches Beispiel zum Quine-McCluskey Verfahren zum Minimieren

Eine interessante Aufgabe ist es, die minimale DNF oder KNF für eine Formel zu bestimmen. Das Verfahren von Quine und McCluskey wendet auf eine gegebene Formel F obige Regeln so lange an, bis keine weitere Verkürzung mehr möglich ist. Die Regeln werden dabei auch auf die abgeleiteten Klauseln angewandt. Sei zum Beispiel

Die Zeichen „:=“ werden gelesen als *definiert als/durch*.

$$F := (A \wedge B \wedge C) \vee (A \wedge B \wedge \overline{C}) \vee (A \wedge \overline{B} \wedge \overline{C}) \vee (\overline{A} \wedge B \wedge C).$$

Wir können nun Regel (1.11) mehrfach anwenden, wie in folgendem Diagramm gezeigt wird.



Auf der linken Seite stehen die Klauseln von F, auf der rechten Seite die verkürzten Klauseln. Auf diese verkürzten Klauseln lässt sich Regel (1.11) nicht weiter anwenden. Die Klauseln, die sich nicht weiter

verkürzen lassen nennt man *Primimplikanten* von F. Die Disjunktion aller Primimplikanten ist äquivalent zu F:

$$F \equiv (A \wedge B) \vee (A \wedge \overline{C}) \vee (B \wedge C).$$

Wie wir an dem Beispiel sehen, kann eine Klausel in mehrere solcher Ableitungsschritte involviert sein. Wir stellen dies durch folgende Tabelle dar.

	1	2	3	4
$A \wedge B$	×	×		
$A \wedge \overline{C}$		×	×	
$B \wedge C$	×			×

Die Zahlen 1, 2, 3, 4 in den Spalten stehen stellvertretend für die Klauseln von F, von links nach rechts durchnummeriert. Für jede der erhaltenen verkürzten Klauseln haben wir eine Zeile. In der Tabelle markieren wir die Eingangsklauseln aus denen die kürzeren Klauseln erzeugt wurden.

Nun sieht man, dass man gar nicht alle abgeleiteten Klauseln braucht: bereits die beiden unteren Klauseln $(A \wedge \overline{C})$ und $(B \wedge C)$ decken alle vier Klauseln von F ab. Die Formel F lässt sich also noch weiter verkürzen, es gilt

$$F \equiv (A \wedge \overline{C}) \vee (B \wedge C).$$

Dies ist nun die minimale DNF von F.

Nur durch Anwendung von Regel 1.11 lässt sich also nicht unbedingt die minimale DNF erzeugen. Es bleibt im Anschluss noch die Aufgabe, aus den abgeleiteten Klauseln eine minimale Auswahl zu treffen. Das Verfahren für die KNF läuft völlig analog mit Regel (1.10).

Grundlagen Mathematik | 01.17: Weiteres Beispiel zum Verfahren nach Quine und McCluskey

Karnaugh-Veitch Diagramme

Eine graphische Darstellung des Quine/McCluskey Verfahrens erhält man mittels *KV-Diagrammen*, benannt nach Karnaugh und Veitch. Es eignet sich aber nur für Formeln mit wenigen Aussagevariablen. Bis zu vier Variablen ist es aber sehr übersichtlich.

Das Verfahren wird ausschließlich in der Videolektion vorgestellt.

Grundlagen Mathematik | 01.18: Karnaugh-Veitch Diagramme, KV-Diagramme

1.5 Die Länge der Normalformen

Dieser Abschnitt ist nicht klausurrelevant. Er bietet aber eine interessante Lektüre. Wir sehen den ersten anspruchsvolleren Beweis und lernen die Paritätsfunktion kennen, die uns sicher wieder begegnen wird.

Wir wollen noch auf einige Aspekte bzgl. der Länge der Normalformen hinweisen. Es gibt Formeln, bei denen die kanonische Normalform bereits minimal ist. Ein Beispiel dafür ist

$$P_n := A_1 \oplus \cdots \oplus A_n.$$

Die Formel P_n hat also n Variable die mit exklusivem oder verknüpft sind. Klammern sind keine gesetzt, da $\oplus$ assoziativ ist. Bei $\oplus$ spricht man manchmal auch von der *Paritätsverknüpfung*, da die Formel P_n genau dann wahr ist, wenn eine ungerade Anzahl der Variablen $A_1, \ldots, A_n$ wahr ist. Diese Eigenschaft werden wir in einem späteren Kapitel voraussichtlich

noch beweisen. Für den Moment genügt uns folgende Beobachtung: wenn man die Belegung einer Variablen von P_n ändert, dann ändert sich auch der Wahrheitswert von P_n. Nehmen wir z. B. die Variable A_n, dann gilt

$$P_n = P_{n-1} \oplus A_n.$$

Nach Definition von $\oplus$ (siehe Wahrheitstabelle auf Seite 2) ändert sich der Wahrheitswert von P_n wenn wir die Belegung von A_n ändern, egal welchen Wahrheitswert P_{n-1} hat.

Theorem 1.5.1. *Die kanonischen Normalformen für P_n sind beide minimal.*

Für das Prinzip eines Widerspruchsbeweises sei auch an Beispiel 1.1.4 erinnert.

Wir führen einen *indirekten Beweis* oder *Widerspruchsbeweis* dessen Vorgehensweise wir hier zunächst noch einmal allgemein erläutern wollen. Bei einem Widerspruchsbeweis nimmt man an, dass die zu beweisende Aussage falsch ist. Dann zeigt man, dass diese Annahme zu einem Widerspruch führt und die Annahme deswegen falsch sein muss. Jetzt habe wir eine doppelte Negation vorliegen: wir haben gezeigt, dass die negierte Aussage falsch ist. Folglich war die Aussage richtig.

Beweis von Theorem 1.5.1. Wir beweisen die Behauptung für die DNF. Wir machen also die Annahme, dass es eine DNF für P_n gibt, die nicht kanonisch ist. Sei $P_n = K_1 \vee \cdots \vee K_m$ eine solche nicht kanonische DNF, wobei $K_1, \ldots, K_m$ die und-Klauseln sind. Dann kommt in (mindestens) einer der Klauseln, sagen wir K_1, nicht alle n Variablen vor. Sagen wir, A_n kommt in K_1 nicht vor. Dann können wir die restlichen $n-1$ Variablen so belegen, dass K_1 wahr ist. Wenn K_1 wahr ist, ist auch P_n wahr, da die Klauseln mit oder verknüpft sind. Dies gilt, egal ob A_n mit wahr oder falsch belegt wird. Dies ist aber ein Widerspruch zu unserer oben gemachten Beobachtung, dass sich der Wahrheitswert von P_n ändert wenn wir die Belegung von A_n ändern.

Folglich war unsere Annahme falsch und und somit die Aussage des Satzes richtig. Den Beweis für die KNF überlassen wir dem Leser. □

Das Ende eines Beweises kennzeichnen wir mit einem offenen Tombstone Symbol. Das Symbol geht auf den Mathematiker Paul Halmos zurück und steht für *quod erat demonstrandum*, was zu beweisen war.

Es gibt gleich viele Belegungen die eine gerade bzw. eine ungerade Anzahl von Variablen mit wahr belegen. Da P_n die Paritätsformel ist, haben also DNF und KNF jeweils 2^{n-1} Klauseln.

Als nächstes sehen wir ein Beispiel mit sehr kurzen Normalformen. Dazu betrachten wir folgende Wahrheitstabelle einer Formel G:

A	B	C	G
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Die obere Hälfte der Wahrheitstabelle ist also falsch, die untere Hälfte wahr. D. h. dass die kanonischen Normalformen aus jeweils 4 Klauseln bestehen. Die Formel G hat aber eine sehr einfach DNF und KNF die nur

aus einer Klausel besteht: $G \equiv A$. Verallgemeinert man das Beispiel auf n Aussagevariablen, so haben die kanonischen Normalformen, genau wie bei der Parität oben, jeweils 2^{n-1} Klauseln. Hier lassen sich beide aber bis auf eine Klausel verkürzen.

Das letzte Beispiel ist aber eher die Ausnahme. Meistens ist es so wie bei der Parität, dass man die kanonischen Normalformen nicht verkürzen kann, oder zumindest nur sehr wenig. Das kann man sich einfach plausibel machen: von den kürzeren Formeln gibt es ja viel weniger als von längeren, eben weil sie kürzer sind. Entsprechend kann es nur für wenige längere Formeln eine äquivalente kürzere geben. D. h. insbesondere, dass wenn eine der beiden Normalformen kurz ist, dass dann die andere in der Regel lang ist, so wie es auch bei den kanonischen Normalformen der Fall ist. Ein Beispiel liefert folgende Formel

$$F := (A_1 \oplus A_2) \wedge (A_3 \oplus A_4) \wedge \cdots \wedge (A_{n-1} \oplus A_n),$$

mit n Variablen $A_1, \ldots, A_n$, wobei wir annehmen, dass n gerade ist. In F werden also die Variablen in $n/2$ Zweier-Gruppen eingeteilt. Die erste Gruppe ist A_1, A_2, die zweite Gruppe A_3, A_4, usw. Die Formel F ist wahr, genau dann wenn in jeder Gruppe genau eine der beiden Variablen wahr ist. Für jede der $n/2$ Gruppen gibt es 4 Belegungen und bei 2 davon wird die entsprechende Klammer in F war. Folglich gibt es insgesamt $2^{n/2}$ Belegungen, bei denen F wahr wird.

Solche Argumente werden wir später im Kapitel über Kombinatorik vertiefend studieren.

- Mit einem ganz ähnlichen Argument wie in Theorem 1.5.1 kann man zeigen, dass die kanonische DNF für F minimal ist. D. h. die minimale DNF von F hat $2^{n/2}$ Klauseln.
- Betrachten wir die KNF von F. Dazu ersetzen wir einfach die Klammern durch ihre KNF gemäß Äquivalenz (1.9) auf Seite 12. D. h. jede der $n/2$ Klammern wird durch 2 Klauseln ersetzt. Die entstehende KNF hat folglich $2 \cdot \frac{n}{2} = n$ Klauseln.

Die minimale DNF von F ist also exponentiell größer als die minimale KNF von F. Entsprechend hat jeder Algorithmus, der die KNF von F in eine DNF umwandelt, exponentielle Laufzeit.

Betrachtet man statt F die Formel G,

$$G := (A_1 \oplus A_2) \vee (A_3 \oplus A_4) \vee \cdots \vee (A_{n-1} \oplus A_n),$$

so erhält man ein analoges Beispiel mit kleiner DNF und großer KNF.

1.6 Resolution

Grundlagen Mathematik | 01.19: Resolution zum Beweis der Unerfüllbarkeit Boolescher Formeln

Die Resolution ist ein Verfahren der formalen Logik, um eine logische Formel auf Gültigkeit zu testen. Genauer ist es ein Widerlegungsverfahren: Statt direkt die Allgemeingültigkeit einer Formel zu zeigen, leitet es einen logischen Widerspruch aus deren Verneinung ab. Diese Herleitung geschieht mittels eines Algorithmus auf rein formalem Weg und kann deshalb von einem Computerprogramm durchgeführt werden. Die Resolution ist eine der bekanntesten Techniken des Maschinengestützten Beweisens.

Die Größe und der Platz von Resolutionsbeweisen ist ein sehr aktuelles Forschungsgebiet. Aktuelle Forschungsarbeiten und Vorträge meinerseits zum von der Deutschen Forschungsgemeinschaft (DFG) geförderten Forschungsprojekt *Komplexitätsmaße für die Lösung von aussagenlogischen Formeln* finden Sie unter `https://www.uni-ulm.de/in/theo/m/woerz/`.

Im nächsten Kapitel lernen wir die Vereinigung von Mengen erst formal kennen. Im Video ist aber erklärt, was dies anschaulich bedeutet.

Seien C und D Klauseln einer aussagenlogischen Formel, die in konjunktiver Normalform vorliegt. Gibt es ein Literal L, welches in C positiv und in D negativ vorkommt, ist die Vereinigung beider Klauseln ohne das positive und negative Literal L der Resolvent R, in Zeichen:

$$R := \big(C_1 \setminus \{L\}\big) \cup \big(C_2 \setminus \{\overline{L}\}\big).$$

Das heißt insbesondere: Gibt es kein komplementäres Literal, so gibt es auch keinen Resolventen. Es darf immer nur genau ein Literal resolviert werden. Je nach Ausgangsklauseln ist die Bildung verschiedener Resolventen möglich.

Ein Beispiel zum Verfahren wurde in der Videolektion gegeben.

Anders notiert: Aus

$$(L \vee A_2 \vee \cdots \vee A_n) \wedge (B_1 \vee B_2 \vee \cdots \vee B_m \vee \neg L)$$

wird auf den Resolventen

$$A_2 \vee \cdots \vee A_n \vee B_1 \vee \cdots \vee B_m$$

geschlossen.

Der Resolvent ist *nicht* äquivalent zu den Ausgangsklauseln. Die Bedeutung des Resolventen liegt vielmehr darin, dass die Ausgangsklauseln nur dann beide gleichzeitig erfüllbar sind, wenn auch der Resolvent erfüllbar ist (eine sogenannte *notwendige Bedingung*). Gelingt es, die *leere Klausel* zu resolvieren, die stets unerfüllbar ist, ist somit die Unerfüllbarkeit der gesamten Formel gezeigt.

Die leere Klausel ist die Klausel mit keinen Literalen.

1.7 Prädikatenlogik

Grundlagen Mathematik | 01.20: Prädikatenlogik

In der nebenstehenden Videolektion geben wir eine Einführung in die Prädikatenlogik. Wir lernen insbesondere *Generalisierung* kennen und erfahren, wie man *Quantoren* negiert. Prädikatenlogik wird Ihnen vor allem in der Analysis begegnen. Ein sicherer Umgang mit ihr sollte daher eingeübt werden.

1.8 Zusätzliche Übungsaufgaben zum Kapitel

Die hier vorgestellten Übungsaufgaben sind oft leichte Veränderungen von bewiesenen Sachverhalten aus der Vorlesung. Diese werden (anders als die Übungsblätter und Tutorien-Übungsblätter) nicht besprochen und müssen nicht abgegeben oder bearbeitet werden.

Eine Primzahl ist eine natürliche Zahl, die größer als 1 und ausschließlich durch sich selbst und durch 1 teilbar ist. Insbesondere ist 1 per Definition *keine* Primzahl. Diese Festlegung hat gute Gründe, die durch den Fundamentalsatz der Arithmetik klarer werden (den wir aber an dieser Stelle nicht verstehen zu brauchen).

1. Beweisen Sie durch Widerspruch, dass es unendlich viele Primzahlen gibt (etwas knifflig). Überlegen Sie sich dazu, wie Sie aus einer gegebenen endlichen Liste von Primzahlen eine weitere Primzahl, die nicht in der Liste auftaucht, konstruieren können. Lösungshinweise finden Sie unter `https://de.wikipedia.org/wiki/Satz_von_Euklid`.

2. Beweisen Sie Theorem 1.5.1 für die KNF von P_n.

3. Zeigen Sie, dass dass die kanonische DNF minimal ist für

$$F := (A_1 \oplus A_2) \wedge (A_3 \oplus A_4) \wedge \cdots \wedge (A_{n-1} \oplus A_n).$$

4. Es sei

$$G := (A_1 \oplus A_2) \vee (A_3 \oplus A_4) \vee \cdots \vee (A_{n-1} \oplus A_n).$$

Zeigen Sie:

(a) die kanonische KNF von G hat $2^{n/2}$ Klauseln und ist minimal.
(b) G hat eine DNF mit n Klauseln.

↓ Ende der 2. Vorlesungswoche

2 Mengen

Gregor Cantor definierte 1895 eine *Menge* als eine Zusammenfassung wohldefinierter, unterscheidbarer Objekte. Eine Menge wird als neues Objekt angesehen, die *Menge ihrer Objekte*. Ein Objekt x aus der Menge M wird als *Element von* M bezeichnet. Wir schreiben dafür kurz: $x \in M$. Gehört ein Objekt y nicht zur Menge M, so schreiben wir: $y \notin M$.

Etwas formaler könnte man definieren:

$$y \notin M :\iff \neg(y \in M).$$

Hierbei bedeutet das Zeichen $:\iff$ *per definitionem genau dann wenn*.

Die Anzahl der Elemente einer Menge M wird auch als deren *Mächtigkeit* bezeichnet und mit $|M|$ abgekürzt.

Notation von Mengen. Es gibt verschiedene Möglichkeiten, Mengen aufzuschreiben. Eine davon ist die Aufzählung der Elemente innerhalb von *Mengenklammern* $\{\dots\}$.

Grundlagen Mathematik | 02.01: Einführung Mengenlehre, Notationen, Mengengleichheit

- Zum Beispiel ist $D = \{3\}$ die Menge, die nur aus dem Element 3 besteht. Also ist $|D| = 1$.
- Die Menge der Teiler von 10 ist $T_{10} = \{1, 2, 5, 10\}$. Es gilt also $|T_{10}| = 4$.

Eine natürliche Zahl a ist genau dann ein Teiler einer natürlichen Zahl n, wenn es eine natürliche Zahl b gibt, für die $a \cdot b = n$ gilt. Man schreibt dafür formal: $a \mid n$.

Die *leere Menge* ist die Menge, die keine Elemente enthält. Sie wird mit $\emptyset$ bezeichnet. Es ist also $\emptyset = \{\}$ und $|\emptyset| = 0$. Die Menge der *natürlichen Zahlen* wird mit $\mathbb{N}$ bezeichnet,

$$\mathbb{N} := \{0, 1, 2, \dots\}.$$

Obacht: Hier gibt es unterschiedliche Konventionen. Darauf wird im Video eingegangen.

Da diese Menge unendlich viele Elemente enthält kann man nicht mehr alle aufschreiben. Stattdessen sollen die 3 Punkte andeuten, wie es weiter geht. Aber auch bei Mengen mit endlich vielen Elementen kommt die Schreibweise mit Punkten zum Einsatz: für $n \in \mathbb{N}^+ := \{1, 2, 3, \dots\}$ definieren wir die Menge

$$[n] := \{1, \dots, n\}.$$

Die Menge $[n]$ besteht also aus allen natürlichen Zahlen von 1 bis n. Ein Würfel zeigt die Zahlen der Menge $[6] = \{1, 2, 3, 4, 5, 6\}$ an. Für Lottospieler ist speziell die Menge $[49] = \{1, 2, \dots, 49\}$ interessant. Mit $\mathbb{Z}$ wird die Menge der *ganzen Zahlen* bezeichnet,

$$\mathbb{Z} := \{\dots, -2, -1, 0, 1, 2, \dots\}.$$

Dies ist die Ergänzung der natürlichen Zahlen durch entprechende negative Zahlen.

Eine Menge ist alleine dadurch definiert, welche Elemente sie enthält. Zwei Mengen A und B heißen *gleich*, falls sie die gleichen Elemente enthalten. Wir schreiben dann $A = B$. Insbesondere spielt die Reihenfolge in der man die Elemente einer Menge notiert keine Rolle. Beispielsweise kann man die Menge T_{10} auch wie folgt schreiben: $T_{10} = \{10, 5, 2, 1\}$.

Anders formuliert: $A = B$ gilt genau dann, wenn $\forall x : (x \in A \iff x \in B)$ gilt.

Eine andere Art Mengen zu notieren ist die Form:

$$M = \{\, x \mid x \text{ hat die Eigenschaft } E \,\}.$$

Formaler, mit Hilfe von Prädikaten, haben wir dies im Video besprochen.

Es gehören dann alle Elemente x zur Menge M, die die Eigenschaft E haben. Zum Beispiel

$$\begin{aligned} T_{10} &= \{\, t \mid t \in \mathbb{N} \text{ und } t \text{ teilt } 10 \,\}, \\ [n] &= \{\, k \mid k \in \mathbb{N} \text{ und } 1 \leq k \leq n \,\}. \end{aligned}$$

Arbeitet man über einer Grundmenge, so wie hier über den natürlichen Zahlen, so schreibt man auch (übersichtlicher) $T_{10} = \{\, t \in \mathbb{N} \mid t \text{ teilt } 10 \,\}$. Ein Beispiel für eine unendliche Menge sind die *rationalen Zahlen*,

$$\mathbb{Q} := \left\{ \frac{a}{b} \,\middle|\, a, b \in \mathbb{Z} \text{ und } b \neq 0 \right\}.$$

Grundlagen Mathematik | 02.02: Russelsche Antinomie, Barbier von Sevilla, Barbier-Paradoxon

Antinomien. Der hier vorgestellte Ansatz für die Mengenlehre nach Cantor wird als *naive Mengenlehre* bezeichnet, da er zu Widersprüchen führt. Ende des 19-ten Jahrhunderts entdeckte Bertrand Russell diese sogenannten *Antinomien*. Wir betrachten ein Beispiel.

Ein Teil der Männer in einem Dorf rasiert sich selbst. Die anderen Männer rasiert der Barbier.

Wir bilden folgende „Menge“

$$M := \{\text{alle Männer, die sich nicht selbst rasieren}\}.$$

Nach unserer naiven Definition ist M tatsächlich eine Menge. Aufgrund des Paradoxos, das sich aus der Definition entwickelt, lässt man solche Mengen nicht zu. Es handelt sich formal um eine Klasse. Eine Erklärung sprengt allerdings den Vorlesungsrahmen.

Offenbar werden hier wohldefinierte, unterscheidbare Objekte zusammengefasst. Es handelt sich somit um eine zulässige Mengenbildung gemäß obiger Definition. Die Frage ist nun, ob der Barbier b zu M gehört.

Es gilt

$$\begin{aligned} b \in M &\iff b \text{ rasiert sich nicht selbst} \\ &\iff b \text{ wird vom Barbier rasiert} \\ &\iff b \text{ rasiert sich selbst} \\ &\iff b \notin M, \end{aligned}$$

was absurd ist!

Abbildung 2.1: Die Plaza de España in Sevilla. Es kursieren viele Varianten des Paradoxons, zum Beispiel: *Der Barbier von Sevilla rasiert alle Männer von Sevilla, nur nicht die, die sich selbst rasieren*. Diese Ausschmückung liefert aber genau genommen nicht Russells sinnlose Definition, sondern impliziert nur, dass der Barbier kein Mann von Sevilla ist (vielleicht ein weiblicher Barbier oder ein dort arbeitender Barbier aus einer Nachbarstadt).

Beispiele wie dieses lösten damals eine tiefe Krise in der Mathematik aus. Es bedurfte eines erheblichen Aufwands die Mengenlehre so aufzubauen, dass diese Widersprüche nicht mehr auftreten können. In der heutigen Mengenlehre ist eine Mengenbildung wie in obiger Antinomie nicht möglich. Für unsere Zwecke reicht die naive Mengendefinition aber völlig aus. Unsere Mengen sind immer Teilmengen von vorher festgelegten Grundmengen, wie beispielsweise den ganzen Zahlen. Dann können obige Widersprüche nicht auftreten.

2.1 Teilmengen

Grundlagen Mathematik | 02.03: Teilmengen und Mengengleichheit beweisen – Beispiel

Für zwei Mengen A und B heißt A *Teilmenge von* B (oder B *Obermenge von* A), falls jedes Element von A auch Element von B ist. Wir schreiben dafür kurz: $A \subseteq B$. Zur Teilmengen-Beziehung $\subseteq$ sagt man auch *Inklusion*. Zum Beispiel gilt $T_{10} \subseteq [12] \subseteq \mathbb{N} \subseteq \mathbb{Z} \subseteq \mathbb{Q}$.

Wir können auch $B \supseteq A$ schreiben.

Ist A keine Teilmenge von B, so schreiben wir $A \not\subseteq B$. Beispielsweise ist $[5] \not\subseteq T_{10}$, da $3 \in [5]$ aber $3 \notin T_{10}$.

Für jede Menge A ist A Teilmenge von sich selbst, $A \subseteq A$. Ebenso ist die leere Menge Teilmenge von A, $\emptyset \subseteq A$, da jedes Element der leeren Menge, nämlich keines, auch Element von A ist. Deswegen nennt man A und $\emptyset$ auch die *trivialen Teilmengen von* A. Ist $A \subseteq B$ für eine Menge $B \neq A$, dann sagt man auch, dass A *echte Teilmenge von* B ist, und schreibt $A \subsetneq B$.

Auch hier gibt es unterschiedliche Konventionen, die im Video besprochen wurden.

Die Inklusion ist *transitiv*: für drei Mengen A, B und C gilt

$$A \subseteq B \text{ und } B \subseteq C \implies A \subseteq C.$$

Mengengleichheit beweisen. Eine weitere einfache Beobachtung ist, dass zwei Mengen A und B genau dann gleich sind, wenn jede in der anderen enthalten ist:

$$A \subseteq B \text{ und } B \subseteq A \iff A = B.$$

Diese Eigenschaft ist nützlicher als es auf den ersten Blick erscheinen mag. Oft hat man nämlich den Fall, dass Mengen A und B mit so unterschiedlichen Beschreibungen gegeben sind, dass man nicht sofort sieht, dass in Wirklichkeit $A = B$ gilt. Mit obiger Eigenschaft kann man diesen Nachweis nun in zwei Schritte zerlegen, was oft einfacher ist. Man weist dann nach, dass jede Menge in der anderen enthalten ist.

Als Beispiel betrachten wir die Menge A, die aus allen geraden ganzen Zahlen besteht. Das sind alle Zahlen $g \in \mathbb{Z}$, die sich in der Form $g = 2n$ schreiben lassen, für eine ganze Zahl n,

$$A := \{\, 2n \mid n \in \mathbb{Z} \,\}.$$

Die Menge B besteht aus allen Zahlen, die sich als Summe von zwei ungeraden ganzen Zahlen schreiben lassen,

$$B := \{\, u + v \mid u, v \in \mathbb{Z} \text{ sind ungerade} \,\}.$$

Man sieht also nicht sofort, dass A und B gleich sind (zumindest als Anfänger). Wir zeigen $A = B$, indem wir zunächst $A \subseteq B$, und dann $B \subseteq A$ nachweisen.

Beweis. 1. $A \subseteq B$: Für ein beliebiges Element $a \in A$ zeigen wir, dass a auch in B liegt. Sei $a = 2n$ für eine ganze Zahl n. Wir zerlegen a in $u = 2n - 1$ und $v = 1$. Dann sind u und v beide ungerade und

$$u + v = (2n - 1) + 1 = 2n = a.$$

Damit ist $a \in B$.

2. $B \subseteq A$: Sei nun b ein beliebiges Element aus B. Die Zahl b lässt sich also schreiben als $b = u + v$ für ungerade ganze Zahlen u und v. Da u und v ungerade sind, lassen sie sich wiederum schreiben als

$$\begin{aligned} u &= 2k+1, \\ v &= 2\ell+1, \end{aligned}$$

für ganze Zahlen k und ℓ. Damit erhalten wir

$$b = u + v = (2k+1) + (2\ell+1) = 2k + 2\ell + 2 = 2(k+\ell+1).$$

Da $k + \ell + 1$ eine ganze Zahl ist, ist folglich $b = 2(k + \ell + 1)$ gerade und damit $b \in A$. □

Grundlagen Mathematik | 02.04: Potenzmengen erklärt (die Menge aller Teilmengen)

Potenzmengen. Die Menge [3] hat insgesamt 8 Teilmengen:

$$\emptyset,\ \{1\},\ \{2\},\ \{3\}\ ,\{1,2\},\ \{1,3\},\ \{2,3\},\ \{1,2,3\}.$$

Diese Teilmengen sind wohldefinierte, unterscheidbare Objekte. Nach der Cantor'schen Mengendefinition können wir also die Teilmengen in einer neuen Menge zusammenfassen: Für eine Menge A definieren wir die *Potenzmenge von* A als die Menge $\mathcal{P}(A)$ aller Teilmengen von A,

$$\mathcal{P}(A) := \{\, T \mid T \subseteq A \,\}.$$

Setzen wir also Mengenklammern um obige Aufzählung der Teilmengen von [3], dann erhalten wir die neue Menge $\mathcal{P}([3])$, die Potenzmenge von [3].

Weitere Beispiele sind

$$\begin{aligned} \mathcal{P}([2]) &= \big\{\emptyset,\ \{1\},\ \{2\},\ \{1,2\}\big\}, \\ \mathcal{P}(\{1\}) &= \big\{\emptyset,\ \{1\}\big\}, \\ \mathcal{P}(\emptyset) &= \{\emptyset\}. \end{aligned}$$

Man beachte, dass $\{\emptyset\}$ nicht das gleiche ist wie $\emptyset$: die Menge $\{\emptyset\}$ enthält ein Element, nämlich $\emptyset$. Dagegen enthält $\emptyset$ kein Element.

Obige Beispiele zeigen die Potenzmenge von Mengen mit $k = 0, 1, 2, 3$ Elementen. Die Potenzmenge hat dann jeweils 2^k Elemente. Wir werden später sehen, dass (für endliche Mengen) dies auch allgemein die richtige Formel ist, es gilt

$$|\mathcal{P}(A)| = 2^{|A|}. \tag{2.1}$$

Aus diesem Grund wird in manchen Büchern die Potenzmenge auch mit 2^A statt $\mathcal{P}(A)$ bezeichnet. Gleichung (2.1) liest sich dann wie folgt: $|2^A| = 2^{|A|}$. Wir wollen diese Notation allerdings nicht verwenden.

Weiter gilt für $A \subseteq B$ die Inklusion $\mathcal{P}(A) \subseteq \mathcal{P}(B)$.

Beweis. Ist $T \in \mathcal{P}(A)$, also $T \subseteq A$, dann gilt auch $T \subseteq B$, da ja nach Voraussetzung $A \subseteq B$ gilt und die Inklusion transitiv ist. Folglich ist $T \in \mathcal{P}(B)$. □

2.2 Operationen auf Mengen

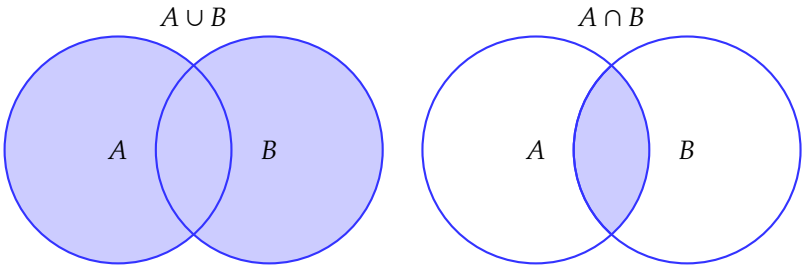
Aus zwei Mengen A und B kann man neue Mengen bilden. Die *Vereinigungsmenge von A und B* wird mit $A \cup B$ bezeichnet und fasst die Elemente beider Mengen in einer Menge zusammen:

$$A \cup B := \{\, x \mid x \in A \text{ oder } x \in B \,\}.$$

Die *Durchschnittsmenge von A und B* wird mit $A \cap B$ bezeichnet und enthält die Elemente in beiden Mengen vorkommen:

$$A \cap B := \{\, x \mid x \in A \text{ und } x \in B \,\}.$$

Die Mengen A und B heißen *disjunkt*, falls $A \cap B = \emptyset$.



Wir fassen die 6 Zahlen, die wir im Lotto getippt haben, in der Menge $T \subseteq [49]$ zusammen. Die Lottoziehung ergab die 6 Zahlen $Z \subseteq [49]$. Dann sind in $T \cup Z$ alle Zahlen aus Tipp und Ziehung und in $T \cap Z$ sind unsere „Richtigen“. Sind T und Z disjunkt, so lagen wir mit unserem Tipp voll daneben!

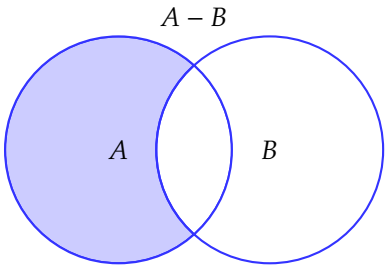
Mit $D = \{3\}$ und $T_{10} = \{1, 2, 5, 10\}$ wie auf Seite 23 sind weitere Beispiele:

$$\begin{aligned}
[6] \cup T_{10} &= \{1,2,3,4,5,6,10\}, & [6] \cap T_{10} &= \{1,2,5\},\\
D \cup T_{10} &= \{1,2,3,5,10\}, & D \cap T_{10} &= \emptyset,\\
\emptyset \cup T_{10} &= T_{10}, & \emptyset \cap T_{10} &= \emptyset,\\
[n] \cup \mathbb{N} &= \mathbb{N}, & [n] \cap \mathbb{N} &= [n].
\end{aligned}$$

Offensichtlich gilt $A \cap B \subseteq A \subseteq A \cup B$.

Die *Differenzmenge von A und B* wird mit $A - B$ oder auch $A \setminus B$ bezeichnet und enthält alle Elemente von A, die nicht in B sind,

$$A - B := \{\, x \mid x \in A \text{ und } x \notin B \,\}.$$



Grundlagen Mathematik | 02.05: Vereinigung, Schnitt, Komplement, symmetrische Differenz von Mengen

Wenn man anstelle zweier Mengen A und B ein *System* oder eine *emph* $\mathcal{S}$ von Mengen betrachtet, d. h. eine Menge von Mengen, dann lassen sich die nebenstehenden Definitionen verallgemeinern: Die *Vereinigung* der Mengen M eines Systems $\mathcal{S}$ ist die Menge

$$\bigcup_{M \in \mathcal{S}} M := \{x \mid \exists M \in \mathcal{S} : \ x \in M\}.$$

Entsprechend ist der *Durchschnitt* der Mengen $M \in \mathcal{S}$ die Menge

$$\bigcap_{M \in \mathcal{S}} M := \{x \mid \forall M \in \mathcal{S} : \ x \in M\}.$$

Sind die Mengen in $\mathcal{S}$ alle paarweise disjunkt, d. h. gilt $M \cap M'$ für alle $M \neq M'$, so nennen wir die Vereinigung aller Mengen aus $\mathcal{S}$ *disjunkte Vereinigung* und notieren dies mit

$$\bigcup_{M \in \mathcal{S}} M.$$

Beispiele sind $[6] - T_{10} = \{3, 4, 6\}$ und $T_{10} - [6] = \{10\}$. Im Lotto-Beispiel sind $T - Z$ alle Nieten, die wir unglücklicherweise anstatt den Zahlen in $Z - T$ angekreuzt haben. Hätten wir die Zahlen in $(T \cap Z) \cup (Z - T)$ angekreuzt, so würden wir jetzt auf Hawaii am Strand brüten, und nicht über diesem Manuskript!

Sei M eine feste Grundmenge, so dass alle betrachteten Mengen Teilmengen von M sind. Sei also $A \subseteq M$. Die Differenz von M und A wird auch als *Komplement von A (bzgl. M)* bezeichnet,

$$\overline{A} := M - A.$$

Ein Beispiel bzgl. der Grundmenge $\mathbb{N}$ ist $\overline{[6]} = \{0, 7, 8, 9, \dots\}$.

Für eine Menge $A \subseteq M$ ergibt die Vereinigung mit ihrem Komplement die Grundmenge und der Schnitt mit ihrem Komplement ist leer,

$$\begin{aligned} A \cup \overline{A} &= M, \\ A \cap \overline{A} &= \emptyset. \end{aligned}$$

Komplementieren wir A doppelt, so erhalten wir wieder A,

$$\overline{\overline{A}} = A.$$

Das Komplement der leeren Menge ist die Grundmenge, und das der Grundmenge ist die leere Menge,

$$\begin{aligned} \overline{\emptyset} &= M, \\ \overline{M} &= \emptyset. \end{aligned}$$

Seien nun $A, B \subseteq M$ und sei $A \subseteq B$. Damit ist also jedes Element aus A auch in B enthalten. Folglich ist jedes Element außerhalb von B auch außerhalb von A. Also gilt dann $\overline{B} \subseteq \overline{A}$,

$$A \subseteq B \quad \Longrightarrow \quad \overline{B} \subseteq \overline{A}.$$

Die Definition von $A - B$ können wir in Bezug auf die Grundmenge M wie folgt umschreiben:

$$\begin{aligned} A - B &= \{\, x \mid x \in A \text{ und } x \notin B \,\} \\ &= \{\, x \mid x \in A \text{ und } x \in \overline{B} \,\} \\ &= A \cap \overline{B}. \end{aligned}$$

Damit können wir die Differenz mit Hilfe von Schnitt und Komplement ausdrücken.

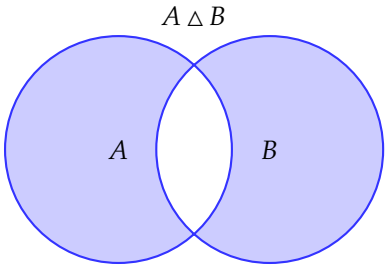
Bei der *symmetrische Differenz von A und B* nehmen wir neben $A - B$ auch die Elemente von $B - A$ auf.

$$A \triangle B := (A - B) \cup (B - A).$$

Die Menge $A \triangle B$ enthält also alle Elemente die in genau einer der Mengen A und B enthalten sind. Das können wir auch durch eine

entweder-oder Verknüpfung ausdrücken:

$$A \triangle B = \{\, x \mid \text{entweder } x \in A \text{ oder } x \in B \,\}.$$



Zum Beispiel ist $[6] \triangle T_{10} = \{3, 4, 6, 10\}$. Es gilt

$$\begin{aligned} A \triangle A &= \emptyset, \\ A \triangle \overline{A} &= M. \end{aligned}$$

Sind A und B disjunkt, dann ist $A \triangle B = A \cup B$. Ist andererseits $A \subseteq B$, dann ist $A \triangle B = B - A$.

2.3 Rechengesetze

Dieser Abschnitt ist für Ihr Selbststudium ohne Videobegleitung bestimmt.

Viele Gesetzmäßigkeiten bekommen wir direkt aus der Logik. Zum Beispiel ist der *Schnitt* zweier Mengen A und B ist mittels einer und-Verknüpfung definiert, und die ist bekanntlich kommutativ. Es gilt also:

$$\begin{aligned} A \cap B &= \{\, x \mid x \in A \text{ und } x \in B \,\} \\ &= \{\, x \mid x \in B \text{ und } x \in A \,\} \\ &= B \cap A. \end{aligned}$$

Folglich ist auch der Durchschnitt eine kommutative Operation.

Das Kommutativgesetz gilt analog auch für die Vereinigung, die mittels einer oder-Verknüpfung definiert ist, und der symmetrischen Differenz, die mittels einer entweder-oder-Verknüpfung definiert ist:

$$\begin{aligned} A \cup B &= B \cup A, \\ A \triangle B &= B \triangle A. \end{aligned}$$

Dagegen ist die einfache Differenz nicht kommutativ: im Allgemeinen ist $A - B \neq B - A$. Zum Beispiel ist $[2] - [1] = \{2\}$ und $[1] - [2] = \emptyset$.

Zur Erinnerung: Um zu zeigen, dass eine Aussage *nicht* gilt, genügt es *ein* konkretes Gegenbeispiel anzugeben. Die Angabe eines Beispiels ist allerdings nichts wert, will man die Allgemeingültigkeit einer Aussage zeigen.

Mit derselben Methode lassen sich Assoziativgesetze zeigen:

$$\begin{aligned} (A \cup B) \cup C &= A \cup (B \cup C), \\ (A \cap B) \cap C &= A \cap (B \cap C), \\ (A \triangle B) \triangle C &= A \triangle (B \triangle C). \end{aligned}$$

und genauso Distributivgesetze:

$$\begin{aligned} A \cup (B \cap C) &= (A \cup B) \cap (A \cup C), \\ A \cap (B \cup C) &= (A \cap B) \cup (A \cap C), \\ A \cap (B \triangle C) &= (A \cap B) \triangle (A \cap C). \end{aligned}$$

Wir zeigen das Argument nocheinmal am Beispiel des dritten Distributivgesetzes:

$$\begin{aligned} A \cap (B \triangle C) &= \{\, x \mid x \in A \text{ und (entweder } x \in B \text{ oder } x \in C) \,\} \\ &= \{\, x \mid \text{ entweder } (x \in A \text{ und } x \in B) \text{ oder } (x \in A \text{ und } x \in C) \,\} \\ &= (A \cap B) \triangle (A \cap C). \end{aligned}$$

Ein entsprechended duales Gesetz gilt allerdings wieder nicht. Zum Beispiel ist für $A = [1]$, $B = [2]$, $C = \emptyset$:

$$\begin{aligned} A \triangle (B \cap C) &= [1] \triangle ([2] \cap \emptyset) = [1], \\ (A \triangle B) \cap (A \triangle C) &= ([1] \triangle [2]) \cap ([1] \triangle \emptyset) = \emptyset. \end{aligned}$$

Auch die Gesetze von de Morgan übertragen sich aus der Logik:

$$\begin{aligned} \overline{A \cup B} &= \overline{A} \cap \overline{B}, \\ \overline{A \cap B} &= \overline{A} \cup \overline{B}. \end{aligned}$$

Ebenso die Verschmelzungsgesetze:

$$\begin{aligned} A \cup (A \cap B) &= A, \\ A \cap (A \cup B) &= A. \end{aligned}$$

Mit diesen Rechenregeln können wir Ausdrücke umformen und erhalt damit weitere Gleichungen. Folgende Rechnung liefert eine weitere Darstellung der symmetrischen Differenz:

$$\begin{aligned} A \triangle B &= (A - B) \cup (B - A) \\ &= (A \cap \overline{B}) \cup (B \cap \overline{A}) \\ &= (A \cup B) \cap (A \cup \overline{A}) \cap (\overline{B} \cup B) \cap (\overline{B} \cup \overline{A}) \quad \text{Distributivgesetz} \\ &= (A \cup B) \cap (\overline{A} \cup \overline{B}) \\ &= (A \cup B) \cap \overline{(A \cap B)} \quad \text{de Morgan} \\ &= (A \cup B) - (A \cap B). \end{aligned}$$

Man kann sich Mengenrechenregeln im Venn-Diagramm graphisch veranschaulichen. Das schauen wir uns im Video an.

Grundlagen Mathematik | 02.06: Mengenrechenregeln am Venn-Diagramm

Freiwilliges Übungsblatt 02

- In Canvas finden Sie unter den Modulen (siehe `https://aalen.instructure.com/courses/1533/modules`) das zweite freiwillige Übungsblatt zur Bearbeitung.

↓ Ende der 3. Vorlesungswoche

Besprechung Blatt 02

Das Freiwillige Blatt 02 wird in folgenden Videos besprochen:

- Aufgabe 1, One-time Pad:

  https://youtu.be/tRemMfmouwY.

- Aufgabe 2, Komplement in Mengensystemen:

  https://youtu.be/xm90PkcQ72o.

- Aufgabe 3, Potenzmengen-Schnitt und -Vereinigung:

  https://youtu.be/m8l6VrMRyC4.

- Aufgabe 4, Analyse des Nim-Spiels:

  https://youtu.be/RULqzcI65sg.

Ergänzend sei angemerkt, dass sich der Begriff *freiwillig* lediglich auf den zu erhaltenden Übungsschein bezieht. Diesbezüglich ist die Bearbeitung der freiwilligen Übungsblätter also freiwillig. Der Stoff, der in diesen Blättern besprochen wird, wird dennoch als klausurrelevant erachtet, zumal die Besprechung des Blattes (in Präsenz) während der Vorlesungszeit stattfindet.

2.4 Kreuzprodukte

Grundlagen Mathematik | 02.07: Kreuzprodukte, geordnete Paare und Tupel

Will man in einem Koordinatensystem mit einer x-Achse und einer y-Achse einen Punkt P spezifizieren, so gibt man (in Bezug auf eine Einheit) die *Koordinaten von* P an. Beispielsweise 3 Einheiten auf der x-Achse und 5 Einheiten auf der y-Achse. Schreibt man die Koordinaten nun einfach als Menge $\{3, 5\}$, so hat man das Problem, dass die Zuordnung zu den Achsen verloren geht, da $\{3, 5\} = \{5, 3\}$. Eine Menge mit 2 Elementen ist gewissermaßen ein *ungeordnetes Paar*. Was wir hier brauchen ist aber ein *geordnetes Paar* mit einer *ersten Komponente*, die wir der x-Achse zuordnen, und einer *zweiten Komponente*, die wir der y-Achse zuordnen.

Ein geordnetes Paar wird auch 2-*Tupel* genannt.

Geordnete Paare erfordern kein neues Konzept, analog zum Mengenbegriff. Man kann geordnete Paare mit einem Trick mittels Mengen definieren.

Definition 2.4.1. Für zwei Elemente a und b ist das *geordnetes Paar* (a, b) definiert durch

$$(a, b) = \{a, \{a, b\}\}.$$

Dabei heißt a die *ersten Komponente von* (a, b), und b die *zweite Komponente von* (a, b).

Mit dieser Definition wird in der Tat erste und zweite Komponente unterschieden.

Lemma 2.4.2. *Zwei geordnete Paare* (a, b) *und* (a', b') *sind genau dann*

Die Unterschiede Axiome, Lemma, Korollar und Satz sind ganz gut im *Simple Club* Video https://youtu.be/xFAVH1bj0fY erklärt. Zur Ergänzung: Ein *Theorem* ist ein besonders wichtiger Satz.

gleich, wenn ihre jeweiligen Komponenten gleich sind, das heißt

$$(a,b) = (a',b') \iff a = a' \text{ und } b = b'.$$

Beweis. Wurde in der Videolektion bewiesen. □

Definition 2.4.3. Seien A und B zwei Mengen. Die Menge der geordneten Paare, die als erste Komponente ein Element aus A und als zweite Komponente ein Element aus B haben, nennt man das *Kreuzprodukt* oder das *kartesische Produkt von A und B*.

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

Ist $A = B$, so schreibt man statt $A \times A$ kürzer A^2.

Für $A = \{0,1\}$ und $B = \{0,1,2\}$ erhalten wir:

$$\begin{aligned}
A \times \emptyset &= \emptyset \times A = \emptyset,\\
A^2 &= \{(0,0),\ (0,1),\ (1,0),\ (1,1)\},\\
A \times B &= \{(0,0),\ (0,1),\ (0,2),\ (1,0),\ (1,1),\ (1,2)\},\\
B^2 &= \{(0,0),\ (0,1),\ (0,2),\ (1,0),\ (1,1),\ (1,2),\ (2,0),\ (2,1),\ (2,2)\}.
\end{aligned}$$

Trägt man die Punkte wie oben beschrieben in ein Koordinatensystem, erhält man bei A^2 die Eckpunkte eines Quadrats mit Seitenlänge 1. Bei $A \times B$ ergibt sich ein Rechteck mit Seitenlängen 1 und 2 und B^2 enthält alle Gitterpunkte eines Quadrats mit Seitenlänge 2.

Sind die Mengen A und B endlich, so wie hier, dann gilt offensichtlich für die Anzahl der Elemente:

$$|A \times B| = |A| \cdot |B|.$$

Insbesondere gilt also $|A^2| = |A|^2$.

Die Mengen müssen aber nicht endlich sein. Analog zu den gerade betrachteten Beispielen besteht $\mathbb{N}^2$ aus allen (unendlich vielen) Gitterpunkten im so genannten *ersten Quadranten* des Koordinatensystems. $\mathbb{N} \times \{0\}$ sind alle Gitterpunkte auf der nicht-negativen x-Achse und $\{0\} \times \mathbb{N}$ sind alle Gitterpunkte auf der nicht-negativen y-Achse.

Das Kreuzprodukt kann man leicht auf mehrere Mengen verallgemeinern. Für 3 Elemente a, b und c definieren wir das *Tripel* (a,b,c) indem wir es in zwei Paarbildungen zerlegen:

$$(a,b,c) := \big((a,b),c\big).$$

Man beachte, dass dabei (a,b) bereits eine Menge ist. Ausgeschrieben erhalten wir

$$\big((a,b),c\big) \;=\; \big\{(a,b),\ \{(a,b),c\}\big\} \;=\; \Big\{\big\{a,\{a,b\}\big\},\ \Big\{\big\{a,\{a,b\}\big\},c\Big\}\Big\}.$$

Nach der gleichen Methode definiert man beliebige *n-Tupel* von Elementen $a_1, \ldots, a_n$:

$$(a_1, a_2, a_3, \ldots, a_n) := \Big(\cdots \big((a_1, a_2), a_3\big) \cdots , a_n \Big).$$

Das Kreuzprodukt von n Mengen $A_1, \ldots, A_n$ ist dann definiert als

$$A_1 \times \cdots \times A_n := \{ (a_1, \ldots, a_n) \mid a_i \in A_i \text{ für } i = 1, \ldots, n \}.$$

Sind alle Mengen gleich, so schreibt man wieder kürzer A^n statt

$$\underbrace{A \times \cdots \times A}_{n-\text{mal}}.$$

Sind die Mengen endlich, so gilt analog wie oben für die Anzahl der Elemente

$$|A_1 \times \cdots \times A_n| = |A_1| \cdot \ldots \cdot |A_n|,$$

und insbesondere also $|A^n| = |A|^n$.

Für $A = \{0, 1\}$ erhalten wir

$$A^3 = \left\{ \begin{array}{l} (0,0,0),\ (0,0,1),\ (0,1,0),\ (0,1,1), \\ (1,0,0),\ (1,0,1),\ (1,1,0),\ (1,1,1) \end{array} \right\}.$$

Wie man sieht gilt $|A^3| = 2^3 = 8$. Allgemein erhält man bei A^n alle 0-1-Tupel der Länge n. Davon gibt es $|A^n| = |A|^n = 2^n$ viele.

3 Relationen

Grundlagen Mathematik | 03.01: Einführung Relationen, Eigenschaften und Äquivalenzrelationen

Das Kreuzprodukt kombiniert jedes Element einer Menge mit jedem Element einer anderen Menge. Oft interessiert einen aber nur ein Teil aller Kombinationen. Dies führt zum Begriff der Relation.

Definition 3.0.1. Seien A und B zwei Mengen und R eine Teilmenge des Kreuzprodukts, $R \subseteq A \times B$, dann heißt R *Relation zwischen A und B*. Ist $A = B$, also $R \subseteq A^2$, so sagt man kürzer: R ist *Relation auf A*.

Ist beispielsweise B eine Menge von Büchern und S eine Menge von Sachgebieten, so bildet man zunächst alle Kombinationen $B \times S$ von Büchern mit Sachgebieten. Eine Bibliothek ist für ihren Katalog dann nur an einem Teil aller Kombinationen interessiert, sie bildet die Relation

$$R = \{ (b, s) \mid \text{Buch } b \in B \text{ hat Sachgebiet } s \in S \}.$$

In der Regel gibt es zu interessanten Sachgebieten mehrere Bücher. Aber auch umgekehrt hat ein Buch oftmals mehrere Sachgebiete. So kann ein Buch mit dem Sachgebiet *Mathematik* auch zum Sachgebiet *Informatik* oder zum Sachgebiet *spannende Unterhaltung* gehören, wobei Letzteres häufig umstritten ist.

Als weiteres Beispiel betrachten wir folgende Relation auf den natürlichen Zahlen, $\mathbb{N} = \{0, 1, 2, \dots\}$:

$$R = \{ (n, m) \mid \text{es gibt ein } k \in \mathbb{N}, \text{ so dass } n + k = m \}.$$

Zu R gehört das Paar $(1, 3)$, da für $k = 2$ gilt $1 + k = 3$. Das Paar $(3, 1)$ gehört nicht zu R, da egal welchen Wert man für k einsetzt, $3 + k \neq 1$ ist. Alle Paare der Form (n, n) sind in R, da für $k = 0$ gilt $n + k = n$. Mit $k = 1$ erhält man alle Paare der Form $(n, n + 1)$. Man sieht, dass alle Paare (n, m) zu R gehören, bei denen n kleiner oder gleich m ist. Die übliche Bezeichnungsweise dafür ist nicht R, sondern $\leq$. Es gilt also $(1, 3) \in \leq$ und $(3, 1) \notin \leq$. Aber auch diese Schreibweise sieht noch etwas ungewohnt aus: Meistens schreibt man das Relationssymbol zwischen die Elemente. D. h. statt $(a, b) \in R$ schreibt man aRb. Statt $(1, 3) \in \leq$ schreibt man also $1 \leq 3$, uns dies ist die aus der Schule gewohnte Darstellung. Man beachte aber, dass über den Wechsel der Schreibweise inhaltlich nichts verändert wurde. Für Anfänger ist es etwas ungewohnt: $\leq$ ist eine Teilmenge des Kreuzprodukts $\mathbb{N}^2$.

Allgemeiner kann man Relationen als Teilmenge des Kreuzprodukts mehrerer Mengen $A_1, \dots, A_n$ definieren, d.h. $R \subseteq A_1 \times \cdots \times A_n$ und spricht dann von einer *n-stelligen Relation zwischen $A_1, \dots, A_n$*. Deswegen bezeichnet man die oben definierte Relation manchmal auch als 2-*stellige* oder *binäre Relation*.

Eigenschaften von Relationen.

Definition 3.0.2. Eine Relation $R \subseteq A \times A$ heißt

- *reflexiv*, falls aRa für alle $a \in A$ gilt.
- *symmetrisch*, falls für alle $a, b \in A$ aus aRb stets bRa folgt.
- *antisymmetrisch*, falls für alle $a, b \in A$ aus aRb und bRa stets $a = b$ folgt.
- *asymmetrisch*, falls für alle $a, b \in A$ aus aRb stets $\neg(bRa)$ folgt.
- *transitiv*, falls für alle $a, b, c \in A$ aus aRb und bRc stets aRc folgt.
- *konnex*, wenn für alle $a, b \in A$ stets $aRb \vee bRa$ gilt.
- *irreflexiv*, wenn für alle $a \in A$ stets $\neg(aRa)$ gilt.

Viele Mathematiker nennen diese Eigenschaft auch *linear*, *total* oder *vollständig*. Sie fordern dann

$$\forall a, b \in A : a \neq b \implies aRb \vee bRa,$$

damit R konnex ist. Dabei ist $\forall a, b \in A$ eine Abkürzung für $\forall a \in A \, \forall b \in A$.

Der Unterschied zwischen antisymmetrischen und asymmetrischen Relationen wurde im Video erklärt.

Um die Begriffe einzuüben, versuchen Sie selbst Relationen (ruhig auch alltägliche, wie z. B. „ist verwandt mit“ auf der Menge der Menschen) zu definieren und untersuchen Sie, welche Eigenschaften sie erfüllen und welche nicht. Hier einige Anregungen:

- $R_1 := \{(x, y) \in \mathbb{N}^2 \mid x < y\}$.
- $R_2 := \{(x, y) \in \mathbb{N}^2 \mid x \leq y\}$.
- $R_3 := \{(x, y) \in \mathbb{N}^2 \mid x \text{ und } y \text{ sind ganze Zahlen}\}$.

Versuchen Sie weiter, die Begriffe voneinander abzugrenzen, z. B.:

- Ist jede nicht reflexive Relation irreflexiv? Warum; warum nicht?
- Ist jede symmetrische Relation antisymmetrisch? Wieso; wieso nicht?

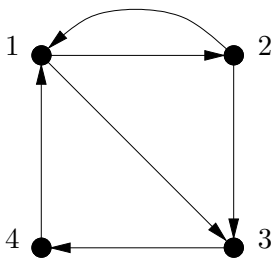
Äquivalenzrelationen. Ist eine Relation reflexiv, symmetrisch und transitiv, so ist sie eine Äquivalenzrelation. Dies ist die vermutlich wichtigste Art von Relationen in der Mathematik. In Abschnitt 3.5 werden wir mehr über diese Relationen erfahren.

Grundlagen Mathematik | 03.02: Graphen als spezielle Relationen

3.1 Relationen als Graphen

Ein häufig vorkommender Spezialfall sind Relationen auf einer endlichen Menge. In diesem Fall spricht man auch von *Graphen*. Sei beispielsweise $A = \{1, 2, 3, 4\}$ und $R = \{(1, 2), (1, 3), (2, 1), (2, 3), (3, 4), (4, 1)\}$. Dann bezeichnet man das Paar (A, R) als *Graph* G, schreibe $G = (A, R)$.

Einen Graph G kann man als Diagramm zeichnen:



Für jedes Element $a \in A$ zeichnen wir einen Punkt den wir mit a beschriften. Statt von Punkten spricht man bei Graphen von *Knoten* (engl. *vertices* oder *nodes*) und verwendet meistens die Bezeichnung V statt A. Die Punkte werden gemäß R durch Pfeile verbunden. Das Paar $(1,2)$ von R wird durch einen Pfeil von 1 nach 2 dargestellt. Analog verfährt man mit den anderen Paaren von R. Statt von Pfeilen spricht man bei Graphen von *Kanten* (engl. *edges*) und verwendet meistens die Bezeichnung E statt R. D. h. in der Regel bezeichnet man einen Graph G durch $G = (V, E)$.

Definition 3.1.1 (Graph). Ein (endlicher) *gerichteter Graph* G ist ein geordnetes Paar (V, E), wobei V eine nicht-leere (endliche) Menge von Knoten und $E \subseteq V \times V = \{(u, v) \mid u, v \in V\}$ eine dazu disjunkte Menge von gerichteten Kanten ist.

Eine Kante (u, u) für ein $u \in V$ heißt *Schlinge*.

Ist $E \subseteq \{(u, v) \mid u, v \in V \text{ und } u \neq v\}$, so spricht man von (endlichen) *einfachen gerichteten Graphen*.

Wir werden später auch noch über (endliche, einfache) ungerichtete Graphen sprechen. Bei diesen ist die Menge E eine Teilmenge von

$$\binom{V}{2} := \{\{u, v\} \mid u, v \in V \text{ und } u \neq v\},$$

d. h. von den 2-elementigen Teilmengen von V. Analog dazu sind die Kantenmengen bei (endlichen) ungerichteten Graphen mit Schlingen eine Teilmenge von

$$V \cup \binom{V}{2}.$$

Da wir in dieser Vorlesung nur endliche Graphen betrachten werden, werden wir dies zukünftig nicht weiter betonen.

Eine weitere Darstellungsform von Relationen auf einer endlichen Menge ist als *Matrix*, der sogenannten *Adjazenzmatrix* des Graphen. Sei $G = (V, E)$ mit $|V| = n$ Knoten. Wir nehmen der Einfachheit halber an, dass $V = \{1, 2, \ldots, n\}$ ist. Bestand V ursprünglich aus irgendwelchen anderen Elementen, so kann man diese durchnummerieren und dann nur mit diesen Nummern arbeiten. Das reicht für unsere Zwecke vollkommen aus. Die Matrix A ist eine quadratische $n \times n$ Matrix, d.h. sie hat n Zeilen (waagrecht) und n Spalten (senkrecht). Für den Graph in obiger Abbildung erhalten wir folgende Adjazenzmatrix,

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

Der Eintrag in Zeile i und Spalte j wird mit $a_{i,j}$ bezeichnet. Es gilt also

Für die Reihenfolge der Indizes merke: Zeilen zuerst, Spalten später.

$$a_{i,j} = \begin{cases} 1, & \text{falls } (i, j) \in E, \\ 0, & \text{sonst.} \end{cases}.$$

Wir schreiben kurz $A = \left(a_{i,j}\right)_{1 \leq i,j \leq n}$. Die Darstellung als Adjazenzmatrix liefert eine sehr einfache Umsetzung als Datenstruktur für Graphen in Programmiersprachen als zweidimensionales boolesches Feld (oder array). Damit kann man mit Graphen sehr einfach algorithmisch umgehen.

3.2 Definitions- und Wertebereich

Grundlagen Mathematik | 03.03: Definitionsbereich und Wertebereich von Relationen, Beispiele

Kommen wir zurück zum allgemeinen Fall einer Relation R zwischen den Mengen A und B. Der *Definitionsbereich* $D(R)$ *von* R ist die Menge aller $a \in A$, die in Relation zu einem Element von B stehen,

$$D(R) := \{\, a \in A \mid \text{es gibt ein } b \in B, \text{ so dass } aRb \,\}.$$

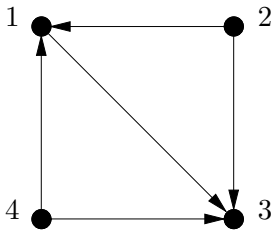
Analog dazu ist der *Wertebereich* $W(R)$ *von* R die Menge aller $b \in B$, die in Relation zu einem Element von A stehen,

$$W(R) := \{\, b \in B \mid \text{es gibt ein } a \in A, \text{so dass } aRb \,\}.$$

Zum Beispiel hat die Relation $<$ auf $\mathbb{N}$ den Definitionsbereich $D(<) = \mathbb{N}$, da jede natürliche Zahl n in $<$-Relation zu einer weiteren natürliche Zahl steht, beispielsweise zu $n+1$. Der Wertebereich ist $W(<) = \mathbb{N} - \{0\}$, da es für alle natürlichen Zahlen bis auf die 0 eine kleinere natürliche Zahl gibt, beispielsweise $n-1$.

Als weiteres Beispiel betrachten wir den Graph $G = (V, E)$ mit $V = \{1,2,3,4\}$ und

$$E = \big\{(1,3),\ (2,1),\ (2,3),\ (4,1),\ (4,3)\big\}.$$



Der Definitionsbereich besteht aus allen Knoten, die als 1. Komponente auftauchen, also $D(E) = \{1,2,4\}$. Das sind also alle Knoten, die Ausgangspunkt einer Kante sind. Entsprechend besteht der Wertebereich aus allen Knoten, die als 2. Komponente vorkommen, also $W(E) = \{1,3\}$. Dies sind alle Knoten, die Endpunkt einer Kante sind. Die Adjazenzmatrix von G ist

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}.$$

Auch hier kann man den Definitionsbereich einfach erkennen: Dies sind alle Knoten, die mindestens eine 1 in der Zeile stehen haben. In A sind dies alle Knoten bis auf Knoten 3, da Zeile 3 eine Nullzeile ist. Für den Wertebereich schaut man sich entsprechend die Spalten von A an. Knoten 2 und 4 haben eine Nullspalte und gehören deshalb nicht zum Wertebereich.

3.3 Die inverse Relation

Grundlagen Mathematik | 03.04: Inverse Relationen - Definition und Beispiel

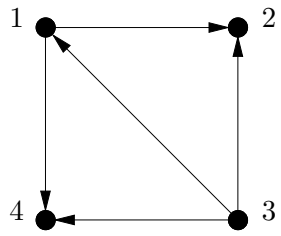
Die zur Relation $R \subseteq A \times B$ *inverse Relation* ist

$$R^{-1} := \{\, (b,a) \in B \times A \mid (a,b) \in R \,\}.$$

D. h. R^{-1} ist eine Relation auf $B \times A$ und enthält genau die Paare von R in vertauschter Reihenfolge. Man bezeichnet R^{-1} auch als die *Umkehrrelation von* R. Bildet man die Inverse der Umkehrrelation, also $(R^{-1})^{-1}$, so werden die Paare (a,b) von R zweimal in ihrer Reihenfolge vertauscht und man landet wieder beim Paar (a,b). Es gilt also $(R^{-1})^{-1} = R$.

Betrachten wir noch einmal obigen Graph $G = (V, E)$. Wir definieren den zu G inversen Graph $G^{-1} = (V, E^{-1})$, wobei E^{-1} die Umkehrrelation von E ist, d. h.

$$E^{-1} = \{(3,1),\ (1,2),\ (3,2),\ (1,4),\ (3,4)\}.$$



D. h. G^{-1} entsteht aus G einfach dadurch, dass man alle Kanten in ihrer Richtung umdreht.

Betrachten wir die Adjazenzmatrix von G^{-1}. Hat G eine Kante von Knoten i nach Knoten j, so hat die Adjazenzmatrix A an Position (i, j) eine 1, d. h. $a_{i,j} = 1$. In G^{-1} wird daraus eine Kante von j nach i, d.h. in der Adjazenzmatrix von G^{-1} steht diese 1 an Position (j, i). Umgekehrt kommt der Wert $a_{j,i}$ von A auf die Position (i, j) in der Adjazenzmatrix von G^{-1}. Es werden also je zwei Einträge von A vertauscht, man sagt auch *transponiert*, indem die jeweiligen Indizes umgedreht werden. Entsprechend bezeichnet man die Adjazenzmatrix von G^{-1} als die *Transponierte von* A und schreibt dafür $A^\top$. In unserem Beispiel erhalten wir

Naheliegend wäre es auch gewesen, von der inversen Matrix A^{-1} zu sprechen. Diese Bezeichnung ist aber bereits für etwas anderes vergeben.

$$A^\top = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Das Vertauschen der Einträge von A kann man auch geometrisch beschreiben: A wird an seiner *Haupdiagonalen* gespiegelt, das ist die Linie durch die Positionen $(1, 1),\ (2, 2),\ \ldots$ von A.

Die Umkehrrelation der Relation $<$ auf $\mathbb{N}$ ist

$$<^{-1} = \{\, (m, n) \in \mathbb{N}^2 \mid n < m \,\}.$$

Es rückt also die größere Zahl nach vorne, aus $1 < 2$ wird $2 <^{-1} 1$. Eine bekanntere Bezeichnung für $<^{-1}$ ist $>$. Statt $2 <^{-1} 1$ schreibt man $2 > 1$. Die Bezeichnung $>$ ist ein kürzerer Name für $<^{-1}$.

3.4 Verkettung von Relationen

Grundlagen Mathematik | 03.05: Verkettung / Komposition von Relationen

Hat man eine Relation R zwischen den Mengen A und B und eine weitere Relation S zwischen den Mengen B und C, so kann man daraus eine Relation zwischen A und C basteln, die *Verkettungsrelation* oder *Hintereinander-Ausührung* von R und S,

$$S \circ R := \{\, (a, c) \in A \times C \mid \exists b \in B :\ \ (a, b) \in R \text{ und } (b, c) \in S \,\}.$$

Weitere Beispiele sind in den empfehlenswerten Videos https://youtu.be/-aXOuleyZRk und https://youtu.be/zx8-e8ZfDgw der Hochschule für Angewandte Wissenschaften Hamburg zu finden. Mein Video habe ich daher kurz gehalten.

↓ **Ende der 4. Vorlesungswoche**

Man beachte die Reihenfolge in der Schreibweise $S \circ R$: die Relation R wird zuerst angewendet, steht aber rechts.

Ist $A = B$, also R eine Relation auf A, dann kann man R mit sich selbst verketten. Statt $R \circ R$ schreibt man kürzer R^2.

Für $k \geq 2$ kann man $R^k := R \circ R^{k-1}$ dann rekursiv definieren.

3.5 Äquivalenzrelationen

Grundlagen Mathematik | 03.06: Äquivalenzrelationen, Äquivalenzklassen, Quotientenmenge, Modulo

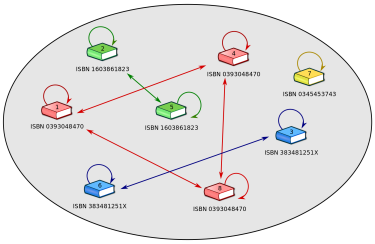


Abbildung 3.1: Eine Menge von Exemplaren von Bücher mit eingezeichneter Äquivalenzrelation „x und y haben dieselbe ISBN-Nummer“ als Pfeildiagramm. Quelle: Wikimedia Commons, Peter Kemp / LGPL.

Zur Wiederholung: Eine Relation $R \subseteq A^2$ heißt *reflexiv*, falls

$$\text{(R)} \quad \forall a \in A \quad aRa.$$

Eine Relation $R \subseteq A^2$ heißt *symmetrisch*, falls

$$\text{(S)} \quad \forall a, b \in A \quad aRb \implies bRa.$$

Eine Relation $R \subseteq A^2$ heißt *transitiv*, falls

$$\text{(T)} \quad \forall a, b, c \in A \quad aRb \text{ und } bRc \implies aRc.$$

Eine Relation $R \subseteq A^2$ heißt *Äquivalenzrelation*, falls sie reflexiv, symmetrisch und transitiv ist.

Bemerkung 3.5.1. *Ist $A = [n]$, so hat eine reflexive Relation auf A die Adjazenzmatrix*

$$\begin{pmatrix} 1 & & * \\ & \ddots & \\ * & & 1 \end{pmatrix},$$

wobei $*$ hier für beliebige Einträge steht. Eine symmetrische Relation hat eine symmetrische *Adjazenzmatrix, d. h. es gilt $A^\top = A$.*

Um die Theorie einfach zu halten, kann man auch definieren, dass ein Graph $G = (V, E)$ ungerichtet heißt, falls für jede Kante $(u, v) \in E$ auch $(v, u) \in E$ liegt, wobei $u \neq v$. D. h. man hat dann:

$$G \text{ ungerichtet} \iff E \text{ symmetrisch.}$$

Wir erhalten dann ein vereinfachtes Diagramm: Statt zwei Pfeilen im Graph, zeichnen wir einfach einen Strich zwischen die entsprechenden Knoten.

Sei $a \in A$. Die *Äquivalenzklasse von a* ist die Menge aller mit a in Relation stehenden Elemente von A,

$$[a]_R := \{\, b \in A \mid aRb \,\}.$$

Das Element a ist der *Repräsentant* der Klasse $[a]_R$. Eine wichtige Eigenschaft von Äquivalenzklassen ist, dass sie *unabhängig vom Repräsentanten* sind. D. h. ist $b \in [a]_R$, dann ist $[a]_R = [b]_R$.

In der Videolektion gehen wir etwas anders als im Skript vor. Schauen Sie sich unbedingt beides an. Die verschiedenen Zugänge helfen Ihnen hoffentlich, den für Sie passenden Zugang zum Thema zu finden.

Lemma 3.5.2 (Unabhängigkeit vom Repräsentanten). *Sei R Äquivalenzrelation über A und $a, b \in A$. Dann gilt*

$$aRb \implies [a]_R = [b]_R\,.$$

Beweis. Wir zeigen zunächst $[b]_R \subseteq [a]_R$: Sei $c \in [b]_R$ beliebig. Es gilt also bRc. Nach Voraussetzung gilt aRb. Da R transitiv ist, gilt somit auch aRc, also $c \in [a]_R$. Folglich ist $[b]_R \subseteq [a]_R$.

Da R symmetrisch ist, gilt auch bRa. Damit erhalten wir analog zu oben $[a]_R \subseteq [b]_R$, und somit $[a]_R = [b]_R$. □

Eine Folgerung aus Lemma 3.5.2 ist, dass Äquivalenzklassen entweder gleich oder disjunkt sind. Sie können sich nicht wie beliebige Mengen nur teilweise überlappen.

Lemma 3.5.3. *Sei R Äquivalenzrelation über A und $a, b \in A$. Dann gilt*

$$\textit{entweder } [a]_R = [b]_R \textit{ oder } [a]_R \cap [b]_R = \emptyset\,.$$

Beweis. Ist $[a]_R \cap [b]_R = \emptyset$, so liegt ein Fall vor, wie er behauptet wird. Nehmen wir also an, dass $[a]_R \cap [b]_R \neq \emptyset$. Es gibt also ein $c \in [a]_R \cap [b]_R$. Es gilt folglich sowohl aRc als auch bRc. Aus Lemma 3.5.2 folgt, dass dann $[a]_R = [c]_R$ und $[b]_R = [c]_R$ gilt, und damit auch $[a]_R = [b]_R$. □

Die Äquivalenzklassen kann man wieder zu einer neuen Menge zusammenfassen. Diese nennt man den *Quotientenmenge* oder *Faktormenge von A nach R*,

$$A/R := \{\, [a]_R \mid a \in A \,\}\,.$$

Nach Lemma 3.5.2 sind die Elemente von A/R wohldefiniert, da sie nicht vom Repräsentanten abhängen.

In der Videolektion haben wir hierzu ein Beispiel gesehen. Schauen Sie sich dieses unbedingt an.

Der Abschnitt über Äquivalenzrelationen ist leider nicht ganz einfach. Aber das Konzept ist sehr wichtig in der Mathematik! Stellen Sie also gerne Ihre Fragen. Zum Einüben und zur Verständniskontrolle empfehle ich den Artikel unter https://de.serlo.org/mathe/universitaet/grundlagen-mathematik/relationen/aequivalenzrelationen sowie das Quiz unter https://de.serlo.org/mathe/hochschule/grundlagen-mathematik/relationen/aufgaben-%C3%84quivalenzrelationen.

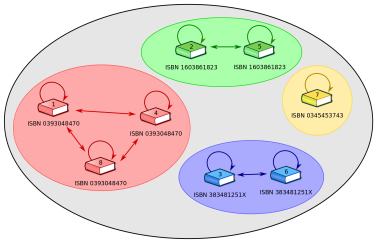


Abbildung 3.2: Menge von Buchexemplaren mit eingezeichneter Äquivalenzrelation „x und y haben dieselbe ISBN-Nummer“ als Pfeildiagramm und mit eingezeichneten Äquivalenzklassen. Quelle: Wikimedia Commons, Peter Kemp / LGPL.

Beispiel 3.5.4 (Partition einer endlichen Zahlenmenge)**.** Wir definieren zunächst sechs Mengen von natürlichen Zahlen von 1 bis 23:

$$\begin{aligned} A_1 &:= \{1, 7, 10, 13, 17\},\\ A_2 &:= \{2, 5, 8, 16\},\\ A_3 &:= \{3, 4, 6, 11, 18, 22\},\\ A_4 &:= \{9, 12, 14, 15, 23\},\\ A_5 &:= \{19\},\\ A_6 &:= \{20, 21\}. \end{aligned}$$

Sie haben die Eigenschaft, dass jede Zahl aus dem Bereich von 1 bis 23 in genau einer der sechs Mengen vorkommt, die damit eine *Zerlegung* oder *Partition* der Menge $A = \{1, \ldots, 23\}$ bilden (zu diesem Begriff sei auf die Videolektion verwiesen). Wie jede Partition von A sind sie die Äquivalenzklassen einer Äquivalenzrelation $\sim$ auf A, nämlich

$$a \sim b \;:\Longleftrightarrow\; \exists i \in \{1, \ldots, 6\}\colon\; a, b \in A_i\,.$$

Die Mengen wurden durch Würfeln ermittelt, also willkürlich aus den rund 44 Billiarden Partitionen — und damit ebenso vielen Äquivalenzrelationen – dieser 23-elementigen Menge ausgewählt.

- Äquivalenzklasse eines Elementes a ist diejenige Menge A_i, die a enthält.

Wie man solche Anzahlen bestimmt sind Fragestellungen aus der Kombinatorik, über die wir erst später sprechen wollen.

- Die Quotientenmenge ist

$$A/\sim = \{A_1, A_2, A_3, A_4, A_5, A_6\}.$$

Beispiel 3.5.5 (Rationale Zahlen)**.** Es sei $P := \{(z, n) \in \mathbb{Z}^2 \mid n \neq 0\}$ die Menge der Paare ganzer Zahlen, deren zweiter Eintrag von Null verschieden ist. Für zwei Paare $(z_1, n_1), (z_2, n_2) \in P$ soll folgende Äquivalenz gelten:

$$(z_1, n_1) \sim (z_2, n_2) \ :\Longleftrightarrow\ z_1 \cdot n_2 = z_2 \cdot n_1.$$

Dieses Beispiel werden wir auch auf dem Freiwilligen Übungsblatt 03 später näher beleuchten.

- Die Äquivalenzklasse des Paares (z, n) ist dann der „Bruch“

$$\tfrac{z}{n} := [(z, n)]_\sim.$$

- Mit der Quotientenmenge erhält man gerade die Menge der rationalen Zahlen

$$\mathbb{Q} := P/\sim = \left\{ \tfrac{z}{n} \;\middle|\; (z, n) \in P \right\}.$$

Beispiel 3.5.6 (Fortsetzung des Beispiels aus der Videolektion)**.** Die Äquivalenzrelation in der Videolektion lässt sich zu $\equiv_m$ für $m \in \mathbb{N}^+$ verallgemeinern: Für $a, b \in \mathbb{Z}$ gilt

$$a \equiv_m b :\Longleftrightarrow\ m|(b - a).$$

Wir schreiben in diesem Fall auch

$$a \equiv b \pmod{m}.$$

Damit ist

$$[r]_{\equiv_m} = \{x \in \mathbb{Z} \mid \exists k \in \mathbb{Z} :\ x = km + r\}.$$

Also ist

$$\mathbb{Z}/{\equiv_m} = \mathbb{Z}/m\mathbb{Z} = \{[0], \ldots, [m-1]\}.$$

Diese Menge ist der sog. *Restklassenring modulo m* der u. a. in der Kryptologie, Zahlentheorie, und Algebra eine äußerst wichtige Rolle spielt.

Grundlagen Mathematik | 3.07: Halbordnungen, kleinste und minimale Elemente und deren Unterschied

3.6 Ordnungsrelationen

Eine Relation $R \subseteq A^2$ heißt *anti-symmetrisch*, falls

$$\text{(AS)}\ \forall a, b \in A \quad aRb \wedge bRa \implies a = b\,.$$

Eine Relation $R \subseteq A^2$ heißt *Halbordnung*, falls sie reflexiv, anti-symmetrisch und transitiv ist.

Eine Relation $R \subseteq A^2$ heißt *Ordnung*, auch *totale Ordnung*, *Totalordnung*,

oder *lineare Ordnung*, falls R Halbordnung ist und zusätzlich gilt:

$$\forall a, b \in A \quad aRb \vee bRa\,.$$

In der Videolektion wurden zahlreiche Beispiele gezeigt, die wir später oft verwenden. Zudem lernen wir Hasse-Diagramme kennen. Diese führen wir im nächsten Abschnitt formal ein. Zum Verständnis hilft es sehr zuerst das Beispiel gesehen zu haben.

Wir haben auch kleinste/größte und minimale/maximale Elemente in Halbordnungen eingeführt und den Unterschied erklärt.

Definition 3.6.1. Ist A eine Menge und $\leq \subseteq A^2$ eine Halbordnung auf A, so nennt man das geordnete Paar $(A, \leq)$ *halbgeordnete Menge*.

Definition 3.6.2. Es sei $(A, \leq)$ eine halbgeordnete Menge. Dann heißt ein Element $a \in A$

- *kleinstes Element*, falls: $\forall b \in A$: $a \leq b$.
- *minimales Element*, falls: $\nexists b \in A$: $b \leq a$ und $b \neq a$.

Ein kleinstes Element (wenn es ein solches gibt; z. B. hat die Menge der ganzen Zahlen kein kleinstes Element) ist mit allen anderen Elementen aus A vergleichbar. Eine halbgeordnete Menge kann nur ein kleinstes (wegen der Anti-Symmetrie), wohl aber mehrere minimale Elemente haben. Hat sie ein kleinstes Element, dann ist es auch minimal.

In einer Totalordnung bedeutet „kleinstes Element“ und „minimales Element“ dasselbe, aber in allgemeinen Halbordnungen kann eine Menge mehrere minimale Elemente haben, von denen dann keines das kleinste ist.

Es kann sogar vorkommen, dass eine (unendliche) Menge A zwar ein einziges minimales Element hat, dieses aber nicht das kleinste Element der Menge ist (dann hat A kein kleinstes Element). Beispiel: Für

$$A := \big\{\{2\}\big\} \cup \bigcup_{0<a<1} \{x \in \mathbb{R} \mid 0 \leq x \leq a\}$$

versehen mit $\subseteq$ als Halbordnung, ist $\{2\}$ zwar das einzige minimale Element, aber nicht das kleinste, da $\{2\} \subseteq S$ nicht für alle S aus A gilt.

Analog gilt:

Definition 3.6.3. Es sei $(A, \leq)$ eine halbgeordnete Menge. Dann heißt ein Element $a \in A$

- *größtes Element*, falls: $\forall b \in A$: $b \leq a$.
- *größtes Element*, falls $\nexists b \in A$: $a \leq b$ und $b \neq a$.

3.7 Hüllenoperatoren und Graphentheorie

Definition 3.7.1. Ist R eine binäre Relation auf einer Menge A, so

Dieser Abschnitt ist für Ihr Selbststudium gedacht. Diese Fähigkeit einzuüben ist auch ein wichtiges Ziel dieser Vorlesung.

Entsprechend könnte man auch die reflexive oder die reflexiv symmetrische Hülle einer Relation definieren. Das Adjektiv in $\mathcal{T}$ muss dazu lediglich angepasst werden.

In der Informatik wird es später darum gehen, Algorithmen zu entwerfen, die solche Probleme lösen, z. B. die transitive Hülle eines Graphen zu bestimmen. Dies leistet der Floyd-Warshall-Algorithmus. Wir lernen diesen später in Abschnitt 4.6 kennen. Zunächst legen wir hier aber die Grundlagen, damit Sie die Problemstellungen (auch in späteren Vorlesungen über Algorithmen) überhaupt verstehen können.

nennt man die Relation

$$R^+ := \bigcap_{T \in \mathcal{T}} T \quad \text{für} \quad \mathcal{T} := \{T \subseteq A \times A \mid R \subseteq T \text{ und } T \text{ ist transitiv}\}$$

die *transitive Hülle* von R. D. h. R^+ ist die Schnittmenge aller transitiven Obermengen von R.

Ein Beispiel in Form eines Graphen haben wir bereits auf Tutoriumsblatt 04 in Aufgabe 8 (c) gesehen. Konsultieren Sie auch dieses Beispiel, um sich obige technische Definition klarer zu machen. Wir stellen Ihnen nun einige weitere Beispiele zur Verfügung.

Beispiel 3.7.2.
- Ist $R := \{(a, b), (b, c)\}$, dann enthält R^+ zusätzlich das Paar (a, c); denn ist T eine transitive Relation, die alle Elemente aus R enthält, also $R \subseteq T$, so muss $(a, c) \in T$ liegen, sonst wäre T schließlich nicht transitiv.
- Ist R die Nachfolgerrelation auf den natürlichen Zahlen, also $R := \{(n', n) \in \mathbb{N} \times \mathbb{N} \mid n' = n + 1\}$, so ist R^+ die Größer-Relation $>$.
- Ist A eine Menge von Personen und gibt E die Elternbeziehung an und $K := E^{-1}$ die Kindbeziehung, so ist E^+ die Vorfahrenbeziehung und K^+ die Nachfahrenbeziehung.

Man beachte, dass in obiger Definition die Schnittbildung nicht-leer ist, da ja immer die sog. *Allrelation* $A \times A$ an der Schnittbildung teilnimmt (denn $A \times A \in \mathcal{T}$).

Zur Erinnerung: Die Inklusion ist eine Halbordnung auf $A \times A$.

Wir haben R^+ so definiert, dass dies die (bzgl. Inklusion) kleinste transitive Relation ist, die R enthält. D. h. für jede transitive Relation T mit $R \subseteq T$ gilt $R^+ \subseteq T$.

Lemma 3.7.3. *Es sei R eine Relation auf einer Menge A und es sei R^+ die transitive Hülle von R. Dann gilt:*

(i) R^+ *ist transitiv.*
(ii) Ist R reflexiv, so ist R^+ reflexiv.
(iii) Ist R reflexiv und symmetrisch, so ist R^+ eine Äquivalenzrelation.
(iv) $R^+ = \bigcup_{k \in \mathbb{N}^+} R^k = R^1 \cup R^2 \cup R^3 \cup \ldots$.

Beweis. Sei $\mathcal{T}$ wie in Definition 3.7.1.

(i) Für $a, b, c \in A$ mit aR^+b und bR^+c folgt aus der Definition von R^+, dass $(a, b), (b, c) \in T$ für alle $T \in \mathcal{T}$. Da alle Relationen in $\mathcal{T}$ transitiv sind, ist $(a, c) \in T$ für alle $T \in \mathcal{T}$, und damit aR^+c, d. h. R^+ ist transitiv. Insbesondere gilt $R^+ \in \mathcal{T}$.

(ii) Dies folgt sofort aus $\{(a, a) \mid a \in A\} \subseteq R \subseteq R^+$.

(iii) Sei nun R reflexiv und symmetrisch. Aus den zwei vorangegangenen Punkten folgt, dass R^+ reflexiv und transitiv ist. Nun zeigen wir, dass R^+ auch symmetrisch, und damit eine Äquivalenzrelation ist. Sei dazu

$$R_{\text{sym}} := R^+ \cap \{(b, a) \mid (a, b) \in R^+\}.$$

Aus $aR_{\text{sym}}b$ folgt aR^+b und bR^+a und daher auch $bR_{\text{sym}}a$, d. h. R_{sym} ist symmetrisch.

Der Name R_{sym} war also sinnvoll gewählt.

Aus aRb folgt bRa und daher aR^+b und bR^+a. Daraus folgt $aR_{\text{sym}}b$, d. h. $R \subseteq R_{\text{sym}}$.
Für $a, b, c \in A$ mit $aR_{\text{sym}}b$ und $bR_{\text{sym}}c$ folgt aR^+b, bR^+c, bR^+a und cR^+b. Da R^+ transitiv ist, folgt aR^+c sowie cR^+a und $aR_{\text{sym}}c$, d. h. R_{sym} ist transitiv.
Insgesamt folgt $R_{\text{sym}} \in \mathcal{T}$ und daher $R_{\text{sym}} \subseteq R^+ \subseteq R_{\text{sym}}$, woraus $R_{\text{sym}} = R^+$ folgt, d. h. R^+ ist symmetrisch.

(iv) Unsere Hilfsmittel reichen zum aktuellen Zeitpunkt noch nicht aus, um diese Aussage zu zeigen. Wir bräuchten hierfür das Prinzip der vollständigen Induktion. Daher nehmen wir die Aussage des Satzes an dieser Stelle nur zur Kenntnis. □

Das Prinzip der transitiven Hülle kann wieder – wie gewohnt – auf Graphen übertragen werden: Die transitive Hülle eines gerichteten Graphen $G = (V, E)$ ist $G^+ = (V, E^+)$ mit $E^+ \supseteq E$ minimal (bzgl. der Inklusion), sodass E^+ transitiv ist. Ist beispielsweise $G = (V, E)$ mit $V = [4]$ und $E = \{(1,2), (2,3), (3,4)\}$ so ist $E^+ = E \cup \{(1,3), (1,4), (2,4)\}$.

Definition 3.7.4. Ein Graph G heißt *transitiv*, falls $G = G^+$ gilt.

Der obere Graph in Abbildung Abbildung 3.3 war also nicht transitiv. Der untere in Abbildung Abbildung 3.3 ist es.

Definition 3.7.5. Ein Tupel $p = (v_0, v_1, \ldots, v_k)$ ist ein *Weg* der Länge $|p| := k$ von v_0 nach v_k in einem Graphen G, falls $(v_{i-1}, v_i) \in E(G)$ für alle $i = 1, 2, \ldots, k$ gilt.

Da man in obiger Definition auch $k = 0$ setzen kann, gibt es immer den Weg (v_0) von v_0 nach v_0 der Länge 0 – auch ohne eine Schleife (v_0, v_0) in $E(G)$.

Manche Autoren fordern in der Definition eines Weges auch, dass kein Knoten zweifach besucht wird. Dies wollen wir nicht tun.

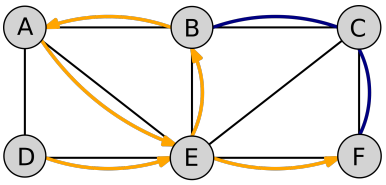


Abbildung 3.4: Zwei Wege in einem Graph. Quelle: Wikimedia Commons, Andreschulz / CC BY-SA.

Satz 3.7.6. *Ist G ein Graph, so gilt*

$$E^+ = E \cup \big\{(u,v) \in E \times E \mid \text{es gibt einen Weg } p \text{ von } u \text{ nach } v \text{ mit } |p| \geq 1\big\}.$$

Beweis. Ohne Beweis. □

Mit anderen Worten: Es gilt

$$xE^+y \iff \exists n \in \mathbb{N} : \exists v_1, \ldots, v_n \in E : xEv_1 \wedge v_1Rv_2 \wedge \cdots \wedge v_nRy.$$

Ähnlich kann man auch die reflexiv transitive Hülle definieren (wir werden dies nun ausschließlich für Graphen tun).

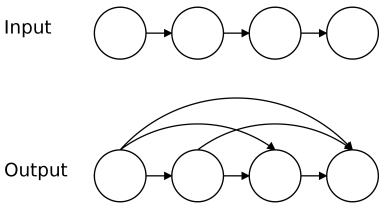


Abbildung 3.3: Die Berechnung einer transitiven Hülle. Genauer gesagt erhält der Warshall-Algorithmus als Input eine Adjazenzmatrix und liefert als Output die transitive Hülle dieser Matrix. Hier sind diese Matrizen als Graphen veranschaulicht. Quelle: Wikimedia Commons, Anish Bramhandkar / CC BY-SA.

Statt die Knoten- und Kantenmengen eines Graphen G mit den Symbolen V und E explizit einzuführen, kann man auch allgemeine Abbildungen V und E definieren, die einen Graphen auf dessen Knoten- und Kantenmenge abbilden. Die Mehrdeutigkeit $V(G) = V$ und $E(G) = E$ nehmen wir in Kauf.

Definition 3.7.7. Ist $G = (V, E)$ ein Graph, so ist seine *reflexiv transitive Hülle* $G^* = (V, E^*)$ definiert durch

$$E^* := E^+ \cup \mathrm{I}_V,$$

wobei

$$\mathrm{I}_V := \{(v, v) \mid v \in V\}$$

die *Identitätsrelation* auf V ist.

Anders ausgedrückt ist

$$E^* = \big\{(u, v) \mid \text{ es gibt einen Weg (evtl. der Länge 0) von } u \text{ nach } v\big\}.$$

Gibt es in einem Graph G einen Weg von einem Knoten u zu einem Knoten v, so wollen wir hierfür auch $u \rightsquigarrow_G v$ schreiben, bzw. $u \rightsquigarrow v$, wenn der Graph aus dem Kontext klar ist.

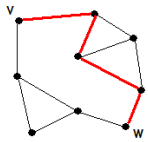


Abbildung 3.5: Ein zusammenhängender Graph mit einem verbindenden Weg.

Definition 3.7.8. Ein ungerichteter Graph G heißt *zusammenhängend*, falls

$$\forall u, v \in V(G) : \; u \rightsquigarrow v.$$

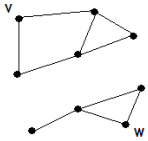


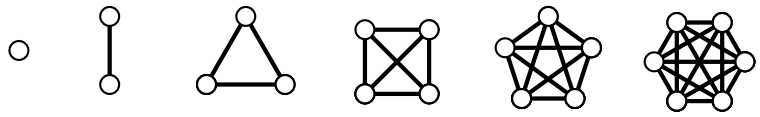
Abbildung 3.6: Ein unzusammenhängender Graph mit zwei Zusammenhangsksomponenten.

Definition 3.7.9. Eine *Zusammenhangskomponente* eines Graphen $G = (V, E)$ ist eine (bezüglich Inklusion) maximale Teilmenge $C \subseteq V$, sodass

$$\forall u, v \in C : \; u \rightsquigarrow v.$$

Definition 3.7.10 (Vollständige Graphen)**.** Für $n \in \mathbb{N}^+$ bezeichne K_n den Graphen mit der Knotenmenge $V = \{v_1, \ldots, v_n\}$ und Kantenmenge $E = \big\{\{v_i, v_j\} \,\big|\, 1 \le i < j \le n\big\}$; dies ist genau die Menge von Kanten zwischen zwei paarweise verschiedenen Knoten.

Hier sind $K_1, \ldots, K_6$ abgebildet:



Man beachte, dass alle K_n bereits transitiv sind. Entsprechend bezeichne K_n^* die reflexive Hülle von K_n.

Skizzieren Sie sich als Übung K_3^*.

Um die Notationen verständlich zu halten, identifizieren wir ab hier einen Graphen mit der Relation, die er definiert.

Satz 3.7.11. *Ist $G = (V, E)$, so gilt: G ist Äquivalenzrelation genau dann, wenn alle Zusammenhangskomponenten von G die Form K_n^* für ein $n \in \mathbb{N}^+$ haben, d. h. , wenn E geschrieben werden kann als*

$$G = K_{n_1}^* \cup K_{n_2}^* \cup \ldots \cup K_{n_t}^*$$

mit $n_1, \ldots, n_t \in \mathbb{N}^+$.

Beweis. Ohne Beweis. □

Definition 3.7.12. Es sei G ein gerichteter Graph. Das k-Tupel $c = (v_1, v_2, \ldots, v_k)$ heißt *(einfacher) Kreis* in G, falls

- $k \geq 1$ und
- $(v_i, v_{i+1}) \in E(G)$ für alle $i = 1, 2, \ldots, k-1$
- und $(v_k, v_1) \in E(G)$
- und $v_i \neq v_j$ für alle $1 \leq i < j \leq k$.

Die *Länge* von c notieren wir it $|c| := k$.

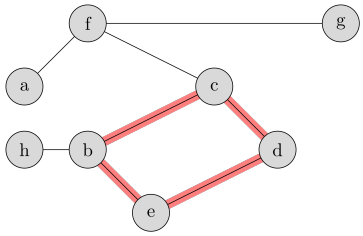


Abbildung 3.7: Ein ungerichteter, zyklischer Graph. Quelle: Wikimedia Commons, MartinThoma / CC BY-SA.

Der Spezialfall $k = 1$: Der Kreis besteht aus einem Knoten, $c = (v_1)$ und $(v_1, v_1) \in E(G)$. Verzichtet man auf die Forderung 4, so spricht man von einem *Zyklus*.

Definition 3.7.13. Ein Graph heißt *azyklisch*, falls G keine Zyklen enthält. Andernfalls heißt er *zyklisch*.

Definition 3.7.14. Ein Graph G heißt *Ablauf-Diagramm*, falls G^* eine Halbordnung ist.

Ist G ein minimales Ablaufdiagramm (bzgl. der Inklusion $\subseteq$), so heißt G *Hasse-Diagramm* der Halbordnung G^*.

Beispiele haben wir in der Videolektion des vorherigen Abschnitts gesehen.

Satz 3.7.15. *Ist G ein einfacher, gerichteter Graph, so gilt: G ist ein Ablaufdiagramm genau dann, wenn G azyklisch ist.*

Beweis. Man zeigt die Kontraposition, also

$$G \text{ kein Ablaufdiagramm} \iff G \text{ zyklisch.}$$

Wir zeigen die Äquivalenz getrennt in zwei Implikationen.
($\Longrightarrow$). Wenn $G = (V, E)$ kein Ablaufdiagramm ist, dann ist $G^* = (V, E^*)$ keine Halbordnung. Da G^* per Definition reflexiv und transitiv ist, muss (AS) verletzt sein. Folglich existieren zwei Knoten $u \neq v$ mit $(u, v) \in E^*$ und $(v, u) \in E^*$. Die Bemerkung nach Definition 3.7.7 liefert, dass es Wege $u \rightsquigarrow v$ und $v \rightsquigarrow u$ in G geben muss, Also haben wir einen Zyklus u–v–u.
($\Longleftarrow$). Sei z ein Zyklus in G, d. h. $z = (v_1, v_2, \ldots, v_k)$ mit $k \geq 2$ und $v_1 \neq v_2$ (da G ein einfacher Graph ist). Insbesondere ist $(v_1, v_2) \in E \subseteq E^*$ und $(v_2, v_1) \in E^*$, da $(v_2, v_3, \ldots, v_k, v_1)$ ein Weg in G ist. Also gilt $\neg$(AS) in G^*. Damit ist G^* keine Halbordnung. □

Das Gegenstück zur transitiven Hülle ist die *transitive Reduktion*. Eine transitive Reduktion einer Relation R ist eine bzgl. der Inklusion minimale Relation R', so dass $R^+ = (R')^+$, also eine minimale Relation mit derselben transitiven Hülle. Es gibt sowohl Relationen, für die keine transitive Reduktion existiert, als auch solche, für die mehrere unterschiedliche transitive Reduktionen existieren. Für gerichtete endliche azyklische Graphen jedoch existiert die transitive Reduktion und ist eindeutig. Man kann zeigen, dass $R' = R^+ \setminus (R^+)^2$ gilt.

Freiwilliges Übungsblatt 03

In den Modulen auf Canvas finden Sie das dritte freiwillige Übungsblatt, zu dem eine PDF-Musterlösung veröffentlicht werden wird.

↓ Ende der 5. Vorlesungswoche

Besprechung Blatt 03

Für das Freiwillige Blatt 03 erhalten Sie eine handschriftliche Musterlösung (unter den Modulen in Canvas). Gehen Sie diese sorgfältig durch und üben Sie sich am Kritisieren Ihrer eigenen Argumente. Haben Sie dies wirklich so detailliert geschrieben wie in der Musterlösung? Ist etwas an Ihrer Lösung nicht hieb- und stichfest? Durch diese Korrektur lernen Sie Fehler bei sich aufzuspüren.

4 Funktionen

Definition 4.0.1. Es seien X und Y nicht-leere Mengen.

1. Eine *Abbildung* von X nach Y ist eine Relation $f \subseteq X \times Y$, die die folgenden Eigenschaften hat:
   a) *Linkstotal*: Für jedes Element $x \in X$ existiert mindestens ein Element $y \in Y$, so dass $(x, y) \in f$ ist.
   b) *Rechtseindeutig/Funktional*: Zu jedem Element $x \in X$ gibt es höchstens ein Element $y \in Y$, so dass $(x, y) \in f$.

   Die Eigenschaften der Linkstotalität und Funktionalität lassen sich mit dem *vertikalen Linientest* überprüfen:

   Zu jedem $x \in X$ gibt es genau ein $y \in Y$, so dass $(x, y) \in f$.

   Gilt $(x, y) \in f$, so schreiben wir $y = f(x)$.
2. Die Menge X heißt *Definitionsbereich* von f. Die Menge Y heißt *Wertebereich* von f. Das *Bild* von $x \in X$ unter f ist $f(x) = y$ und x heißt *Urbild* von $y = f(x)$.
3. Eine Abbildung von X nach Y wird notiert als $f : X \to Y$ und die Zuordnung der Elemente mit $x \mapsto y$.

Beispiel 4.0.2. Definiere $X := \{\text{Apfel}, \text{Erdbeere}, \text{Kiwi}\}$ und $Y := \{0, 1\}$. Dann ist $f : X \to Y$ gegeben durch

$$f(\text{Apfel}) = 1, \quad f(\text{Erdbeere}) = 0, \quad f(\text{Kiwi}) = 0$$

eine Abbildung, während

$$\{(\text{Apfel}, 0), (\text{Apfel}, 1), (\text{Kiwi}, 0)\}$$

nur eine Relation zwischen X und Y ist.

Beispiel 4.0.3. Die Abbildung

$$\text{id}_X : X \to X \text{ mit } \text{id}_X(x) := x \text{ für alle } x \in X$$

heißt *Identität* auf X.

Die Bezeichnung *Funktion* verwendet man vor allem, wenn Y ein Zahlenbereich ist, z. B. $Y \subseteq \mathbb{R}$. Wir unterscheiden aber nicht zwischen den Begriffen.

Beispiel 4.0.4. Die Funktion $f : \mathbb{R} \to \mathbb{R}$ mit $x \mapsto 2x + 1$ kann äquivalent auch als Relation $\{(x, y) \in \mathbb{R}^2 \mid y = 2x + 1\}$ geschrieben werden.

Wir merken an, dass zwei Funktionen $f : X \to Y$ und $g : X \to Y$ genau dann gleich sind, wenn $f(x) = g(x)$ für alle $x \in X$ gilt. Wir schreiben dafür $f = g$.

Grundlagen Mathematik | 04.01: Einführung Funktionen

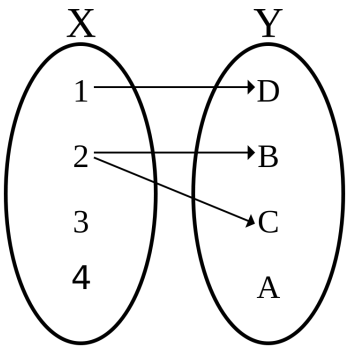


Abbildung 4.1: Das Diagramm repräsentiert die Menge von geordneten Paaren $\{(1, D), (2, B), (2, C)\}$. Dies ist jedoch *keine* Abbildung, denn das Element 2 kommt in mehreren geordneten Paaren im ersten Eintrag vor, nämlich $(2, B)$ und $(2, C)$. Zwei weitere Gründe sind, dass weder 3 noch 4 erste Einträge eines geordneten Paares der Menge sind. Quelle: Wikimedia Commons, Bin im Garten / CC BY-SA.

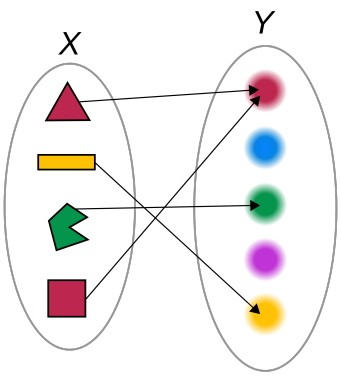


Abbildung 4.2: Eine Abbildung, die vier farbige Formen (rotes Quadrat, rotes Dreieck, gelbes Rechteck, grünes Polygon) auf 5 Farben (rot, blau, grün, violett, gelb) abbildet. Quelle: Wikimedia Commons, Wvbailey / CC BY-SA.

Grundlagen Mathematik | 04.02: Mengen von Abbildungen und Graph einer Abbildung. Man beachte: Die Achsenbeschriftung im Video für Abbildung 4.3 ist nicht korrekt. Dies spielt aber keine Rolle.

Definition 4.0.5. Es seien X und Y zwei Mengen. Die Menge aller Abbildungen von X nach Y bezeichnet man mit Y^X, d. h. es gilt

$$Y^X := \{f \mid f \text{ ist eine Abbildung von } X \text{ nach } Y\}.$$

Sind X und Y endliche Mengen, so gilt $|Y^X| = |Y|^{|X|}$. Dies werden wir später in Korollar 6.1.3 sehen.

Beispiel 4.0.6. Es gilt

$$\{0,1\}^{\mathbb{N}} = \big\{f \mid f \text{ ist eine Funktion von } \mathbb{N} \text{ nach } \{0,1\}\big\}.$$

Ebenso ist

$$\mathbb{N}^{\mathbb{N}} = \{f \mid f \text{ ist eine Funktion von } \mathbb{N} \text{ nach } \mathbb{N}\}.$$

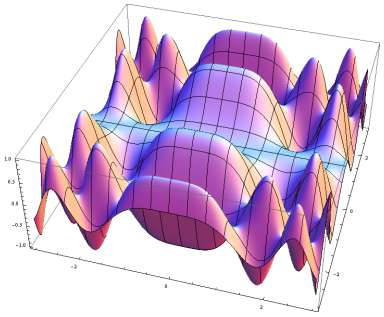


Abbildung 4.3: Ein Ausschnitt des Graphen der Funktion $f\colon \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ mit $f(x,y) := \sin(x^2)\cos(y^2)$. Man beachte, dass man eigentlich $f\big((x,y)\big)$ für den Funktionswert des geordneten Paares (x,y) schreiben müsste. Die übliche Konvention ist jedoch, diese doppelten Klammern wegzulassen. Eine solche Abbildung nennt man dann auch *Funktion in mehreren Variablen*. Quelle: Wikimedia Commons, dino / CC BY-SA.

Definition 4.0.7. Ist $f\colon X \to Y$ eine Funktion, so ist ihr *Graph* die Menge

$$G_f := \big\{\big(x, f(x)\big) \,\big|\, x \in X\big\}.$$

Im Spezialfall, wenn X und Y Teilmengen der reellen Zahlen sind, kann ein Element $(x,y) \in G_f$ mit einem Punkt in der Kartesischen Ebene identifiziert werden, der die Koordinaten x, y hat. Zum Beispiel besteht der Graph der Quadratfunktion $x \mapsto x^2$ aus alle Punkten mit den Koordinaten (x, x^2) für $x \in \mathbb{R}$. Man erhält, wenn man diese in eine Kartesische Ebene zeichnet die bekannte Normalparabel.

An dieser Stelle ist vermutlich anzumerken, dass die üblichen Graphen, die man in der Schule gezeichnet hat, wie z. B. auch nebenstehend in Abbildung 4.2 und Abbildung 4.4 abgebildet, im strengen mathematischen Sinne eigentlich eine Teilmenge von G_f darstellen. Zudem sei erwähnt, dass Graphen von Funktionen und Graphen, die einen Relation darstellen, zwei unterschiedliche Konzepte sind.

Grundlagen Mathematik | 04.03: Bilder und Urbilder von Mengen unter Abbildungen

Definition 4.0.8.
1. Das *Bild einer Teilmenge* $A \subseteq X$ ist definiert durch
$$f(A) := \{y \in Y \mid \exists x \in A \,:\, y = f(x)\}.$$
2. Wir nennen $f(X)$ das *Bild* von f.
3. Das *Urbild* einer Menge $B \subseteq Y$ ist die Menge aller $x \in X$, deren Bilder in B liegen, also
$$f^{-1}(B) := \{x \in X \mid f(x) \in B\}.$$

Man beachte, dass wir also sowohl das Bild eines Elementes $x \in X$ unter f definiert haben, als auch das Bild einer Teilmenge $A \subseteq X$ unter f.

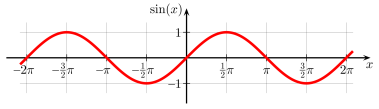


Abbildung 4.4: Ein Ausschitt des Graphen der Funktion $x \mapsto \sin x$.

Beispiel 4.0.9. Sei $f\colon \mathbb{N} \to \mathbb{N}$ gegeben durch $f(n) = 2n$. Mit $M := \{1,3\}$ und $N := \{8,13,14,100\}$ gilt $f(M) = \{f(1), f(3)\} = \{2,6\}$; $f(\mathbb{N})$ ist die Menge aller geraden natürlichen Zahlen; und $f^{-1}(N) = \{4,7,50\}$.

Beispiel 4.0.10. Für die aus der Schule bekannte Funktion $f\colon \mathbb{R} \to \mathbb{R}$

mit $f(x) := \sin x$ (die wir hier nicht formal eingeführt haben) gilt

$$f(\mathbb{R}) = [-1,1] := \{x \in \mathbb{R} \mid -1 \leq x \leq 1\}, \quad f^{-1}\big(\{0\}\big) = \{k\pi \mid k \in \mathbb{Z}\}.$$

Mit anderen Worten: Die Menge $f^{-1}\big(\{0\}\big)$ enthält gerade die Nullstellen von $\sin x$.

Definition 4.0.11. Es sei $f\colon X \to Y$ eine Funktion und $A \subseteq X$. Die Abbildung

$$f|_A\colon A \to Y \text{ mit } f|_A(x) := f(x) \text{ für } x \in A$$

nennt man *Restriktion* oder *Einschränkung* von f auf A.

Grundlagen Mathematik | 04.04: Restriktion und Komposition von Funktionen

Die Komposition von Funktionen ist erklärt durch die Komposition der zugehörigen Relationen.

Definition 4.0.12. Sind $f\colon X \to Y$ und $g\colon Y \to Z$ zwei Funktionen, so ist die *Komposition* $g \circ f$ die Abbildung $g \circ f\colon X \to Z$ mit

$$(g \circ f)(x) := g\big(f(x)\big) \text{ für alle } x \in X.$$

Beispiel 4.0.13. Ist

$$f\colon \mathbb{R} \to \mathbb{R}, \text{ mit } f(x) := x + 2,$$
$$g\colon \mathbb{R} \to \mathbb{R}, \text{ mit } g(y) := y^2 - 3y,$$

dann ist

$$\begin{aligned}(g \circ f)(x) = g\big(f(x)\big) &= f(x)^2 - 3f(x)\\ &= (x+2)^2 - 3(x+2)\\ &= x^2 + 4x + 4 - 3x - 6\\ &= x^2 + x - 2.\end{aligned}$$

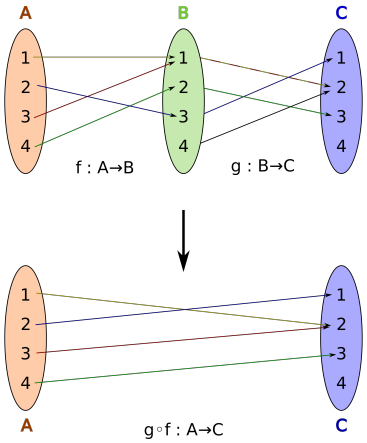


Abbildung 4.5: Funktionenkomposition graphisch veranschaulicht. Quelle: Wikimedia Commons, Stephan Kulla / CC BY-SA.

Wie bei Relationen gilt *im Allgemeinen nicht* $f \circ g = g \circ f$, d. h. es existieren Funktionen $f, g\colon X \to X$, sowie mindestens ein Punkt $x \in X$ mit $(f \circ g)(x) \neq (g \circ f)(x)$. Versuchen Sie es mal mit den Funktionen aus Beispiel 4.0.13.

Lemma 4.0.14. *Die Komposition von Abbildungen ist assoziativ, d. h. sind f, g, h verknüpfbare Abbildungen (etwa $f\colon M \to N$, $g\colon N \to L$, $h\colon L \to K$), so gilt*

$$h \circ (g \circ f) = (h \circ g) \circ f.$$

Das Lemma sagt uns, dass wir auch kurz $h \circ g \circ f$ schreiben können.

Beweis. Übung. □

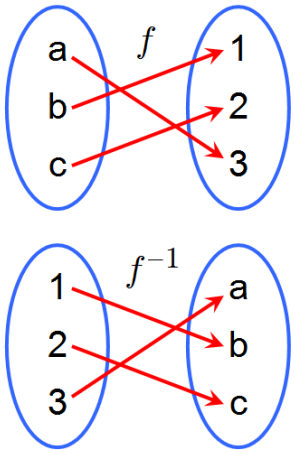


Abbildung 4.6: Eine Funktion f und ihre Umkehrfunktion f^{-1} veranschaulicht. Da z. B. f das Element a auf 3 abbildet, bildet die Inverse f^{-1} das Element 3 zurück auf a ab.

Lemma 4.0.15. *Ist $f\colon X \to Y$ eine Abbildung, so ist die inverse Relation $f^{-1} = \{(y,x) \in Y \times X \mid y = f(x)\}$ genau dann eine Abbildung von Y nach X, wenn der* horizontale Linientest *gilt:*

Zu jedem $y \in Y$ gibt es genau ein Urbild $x \in X$ mit $y = f(x)$.

An dieser Stelle eine Warnung: Man verwechsle die Umkehrfunktion f^{-1} nicht mit der Funktion $x \mapsto 1/f(x)$.

4.1 Injektivität, Surjektivität

Grundlagen Mathematik | 04.05: Injektivität, Surjektivität, Bijektivität mit vielen Beispielen

Definition 4.1.1.

1. Eine Abbildung $f : X \to Y$ heißt *injektiv*, falls jedes Element $y \in Y$ höchstens ein Urbild $x \in X$ besitzt. Mit anderen Worten gilt für alle $x, x' \in X$:
$$f(x) = f(x') \implies x = x',$$
bzw.
$$x \neq x' \implies f(x) \neq f(x').$$
2. Eine Abbildung $f : X \to Y$ heißt *surjektiv*, wenn jedes $y \in Y$ mindestens ein Urbild $x \in X$ besitzt, d. h.
$$\forall y \in Y \; \exists x \in X \; : \; y = f(x);$$
mit anderen Worten, wenn $f(X) = Y$ gilt.
3. Ist eine Abbildung f sowohl injektiv, als auch surjektiv, so heißt sie *bijektiv*. Mit anderen Worten: f besitzt eine Inverse $f^{-1} : Y \to X$. Man nennt f dann auch eine *Bijektion*.

Auf der spanischen Wikipedia (siehe Abbildung 4.7) gibt es ein sehr anschauliches Beispiel in Form einer Tabelle von injektiven / nicht injektiven und surjektiven / nicht surjektiven Funktionen, bei dem Malfarben Hunden zugeordnet werden. Die spanischen Begriffe sind ähnlich genug zu den deutschen, um das Bild und die Tabelle zu verstehen. Man sollte allerdings anmerken, dass die blauen Pinsel in irgendeiner Art und Weise unterscheidbar sein müssen (erinner Sie sich an die Definition einer Menge nach Cantor?).

Injektiv bedeutet anschaulich, dass kein Element im Wertebereich der Funktion doppelt getroffen werden darf; surjektiv bedeutet, dass alle getroffen werden müssen.

Wir können die Begriffe noch anders formulieren: Die Funktion $f : X \to Y$ ist

- injektiv genau dann, wenn
$$\forall y \in Y : \; |\{x \in X \mid f(x) = y\}| \leq 1.$$
- surjektiv genau dann, wenn
$$\forall y \in Y : \; |\{x \in X \mid f(x) = y\}| \geq 1.$$
- bijektiv genau dann, wenn
$$\forall y \in Y : \; |\{x \in X \mid f(x) = y\}| = 1.$$

Sei $f : X \to Y$. Wenn f injektiv ist, hat die Gleichung $f(x) = y$ für jedes gegebene $y \in Y$ höchstens eine Lösung $x \in X$. Ist f surjektiv, so hat die

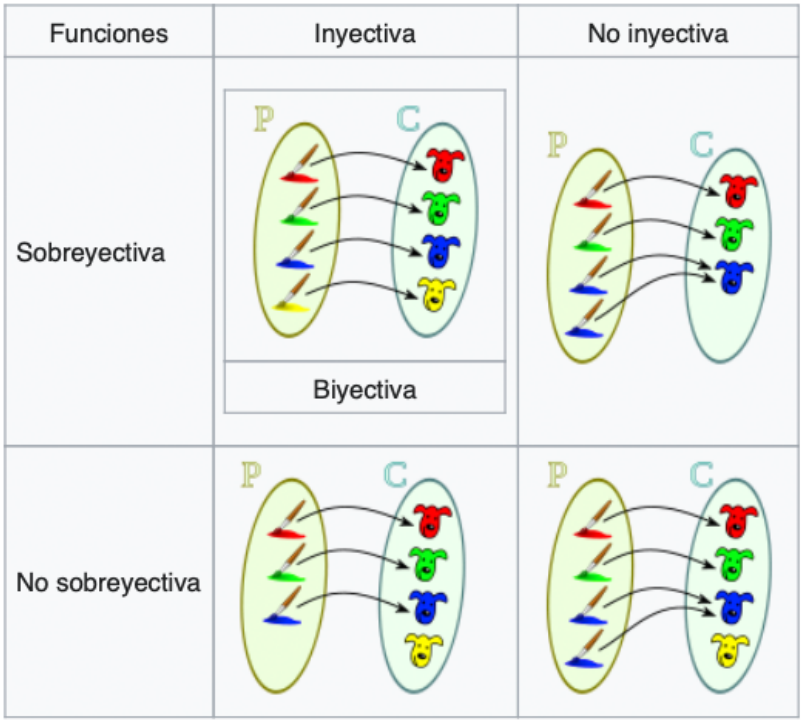


Abbildung 4.7: Quelle: Wikimedia Commons, HiTe / CC BY-SA.

Gleichung für jedes $y \in Y$ eine Lösung $x \in X$. Ist f schließlich bijektiv, dann gibt es für jedes $y \in Y$ immer eine eindeutige Lösung $x \in X$.

Beispiel 4.1.2.

1. Die Funktion $f\colon \mathbb{R} \to \mathbb{R}$ mit $f(x) := x^2$ ist nicht surjektiv, da es zu $y = -1$ kein $x \in \mathbb{R}$ mit $f(x) = y$ gibt. Zudem ist f nicht injektiv, da $f(1) = f(-1) = 1$ gilt.
2. Die Funktion $f\colon \mathbb{R} \to \mathbb{R}$ mit $f(x) := 3x + 1$ ist injektiv denn gilt $f(x_1) = f(x_2)$, so haben wir $3x_1 + 1 = 3x_2 + 1$, also $3x_1 = 3x_2$, also $x_1 = x_2$. Die Funktion ist auch surjektiv, denn ist $y \in \mathbb{R}$ beliebig gewählt, dann setze $x := \frac{y-1}{3}$. Damit gilt dann

$$f(x) = 3x + 1 = 3\left(\frac{y-1}{3}\right) + 1 = y.$$

Insgesamt ist f also bijektiv; die Umkehrabbildung $f^{-1}\colon \mathbb{R} \to \mathbb{R}$ ist gegeben durch $f^{-1}(y) = \frac{y-1}{3}$.

Beispiel 4.1.3. Wir betrachten im Beispiel die Funktion $f\colon X \to Y$ mit $f(x) := x^2$.

1. Ist $X = Y = \mathbb{R}$, so haben wir in obigem Beispiel bereits gesehen, dass f weder injektiv noch surjektiv ist.
2. Ist $X = \mathbb{R}$ und $Y = \mathbb{R}_0^+ := \{y \in \mathbb{R} \mid y \geq 0\}$, dann ist f surjektiv, aber nicht injektiv.
3. Wählt man $X = \mathbb{R}_0^+$ und $Y = \mathbb{R}$, dann ist f injektiv, aber nicht surjektiv.
4. Wählt man schließlich $X = Y = \mathbb{R}_0^+$, dann ist f bijektiv.

Versuchen Sie die Funktionen aus dem Beispiel graphisch darzustellen. Wie würden Sie die Richtigkeit der Aussagen argumentieren?

Grundlagen Mathematik | 04.06: Charakterisierung der Bijektivität durch Links- und Rechtsinverse

Satz 4.1.4. *Eine Abbildung $f: X \to Y$ ist genau dann bijektiv, wenn es eine Abbildung $g: Y \to X$ gibt, so dass $g \circ f = \mathrm{id}_X$ und $f \circ g = \mathrm{id}_Y$ gilt. Die Funktion g ist in diesem Fall die Umkehrabbildung von f.*

Beweis.

1. Es sei f bijektiv. Dann existiert $f^{-1}: Y \to X$. Setze $g := f^{-1}$. Dann gilt $g \circ f = \mathrm{id}_X$ und $f \circ g = \mathrm{id}_Y$, denn für alle $x \in X$ gilt
$$(g \circ f)(x) = (f^{-1} \circ f)(x) = f^{-1}\big(f(x)\big) = x = \mathrm{id}_X(x).$$
Die andere Behauptung zeigt man analog.
2. Es gebe eine Abbildung $g: Y \to X$ mit den beschriebenen Eigenschaften. Zu zeigen ist, dass f bijektiv ist.
   - f ist injektiv: Sei $f(x_1) = f(x_2)$. Dann gilt $g\big(f(x_1)\big) = g\big(f(x_2)\big)$. Also
   $$x_1 = \mathrm{id}_X(x_1) = (g \circ f)(x_1) = (g \circ f)(x_2) = \mathrm{id}_X(x_2) = x_2,$$
   d. h. $x_1 = x_2$.
   - f ist surjektiv: Sei $y \in Y$ gegeben. Dann gilt für $x := g(y) \in X$:
   $$f(x) = f\big(g(y)\big) = (f \circ g)(y) = \mathrm{id}_Y(y) = y. \qquad \square$$

Das Studium dieser Eigenschaften wird in der Analysis weiter vertieft werden. An dieser Stelle benötigen wir aktuell nicht mehr.

4.2 Folgen

Grundlagen Mathematik | 04.07: Definition Folgen und Folgenkonvergenz anschaulich erklärt

Das Thema Folgen werden Sie sehr intensiv in der Analysis behandeln. Wir wollen hier nicht sehr tief darauf eingehen. Dennoch brauchen wir Folgen als Hilfsmittel im weiteren Verlauf.

Definition 4.2.1. Eine *Folge* $f = (a_n)_{n\in\mathbb{N}} = (a_n)_{n=0}^{\infty}$ von Elementen einer Menge X ist eine Abbildung $f: \mathbb{N} \to X$ mit $n \mapsto f(n) =: a_n$.

Beispiel 4.2.2. Wir betrachten die Folge $f: \mathbb{N} \to \mathbb{N}$ mit $f(n) := 2n$. Die Folge könnten wir auch als „Tupel mit unendlich vielen Komponenten“ notieren, also in der Form $(0, 2, 4, 6, 8, \dots)$. Im Unterschied zu einer Menge kommt es hier auf die Reihenfolge der Elemente an. Obige Folge unterscheidet sich z. B. von der Folge $(2, 0, 4, 6, 8, \dots)$.

Definition 4.2.3. Eine Folge $(a_n)_{n\in\mathbb{N}}$ mit Werten aus $\mathbb{R}$ heißt konvergent, falls es eine Zahl $a \in \mathbb{R}$ gibt, sodass für jedes $\varepsilon > 0$ ein $N = N(\varepsilon) \in \mathbb{N}$ existiert mit
$$|a_n - a| < \varepsilon \text{ für alle } n \in \mathbb{N} \text{ mit } n \geq N.$$
Die Zahl a heißt dann *Grenzwert* von $(a_n)_{n\in\mathbb{N}}$. Man schreibt
$$a = \lim_{n\to\infty} a_n.$$

Man beachte, dass in der obigen Definition $\forall \epsilon > 0$ eine gängige Kurzschreibweise für $\forall \epsilon \in (0, +\infty) := \{x \in \mathbb{R} \mid 0 < x < +\infty\}$ ist. Weiter gehört der Satzteil „für alle $n \in \mathbb{N}$ mit $n \geq N$ eigentlich vor die Ungleichung im „Quantorspruch". Man ändert diese Reihenfolge aber gelegentlich zur besseren Lesbarkeit im Deutschen ab.

Beispiel 4.2.4. Wir behaupten, dass $\lim_{n\to\infty} \frac{1}{10^n} = 0$ gilt. Sei dazu $\varepsilon > 0$. Wähle $N := \left\lceil \frac{\log \frac{1}{\epsilon}}{\log 10} \right\rceil + 1 \in \mathbb{N}$. Dann gilt für alle $n \geq N$:

$$\left| \frac{1}{10^n} - 0 \right| = \frac{1}{10^n} < \epsilon.$$

Zur Erinnerung: Ein Element $x \in M$ heißt *kleinstes Element* der Menge M, wenn gilt: $\forall y \in M : x \leq y$. Analog dazu heißt x *größtes Element* von M, falls gilt: $\forall y \in M : y \leq x$. Wir notieren dies mit $\min M$ bzw. $\max M$.

Für eine reelle Zahl $x \in \mathbb{R}$ können wir dann die Gauß-Klammern wie folgt definieren:

$$\lfloor x \rfloor := \max\{k \in \mathbb{Z} \mid k \leq x\},$$
$$\lceil x \rceil := \min\{k \in \mathbb{Z} \mid k \geq x\}.$$

Beispiele wurden in der Videolektion gegeben.

4.3 Rekursive Definitionen

Eine Folge $(a_n)_{n\in\mathbb{N}^+}$ ist *rekursiv definiert*, wenn für ein $k \in \mathbb{N}^+$ die ersten k Folgenglieder $a_1, \ldots, a_k$ festgelegt werden und es eine Funktion $g\colon \mathbb{R}^k \to \mathbb{R}$ gibt, sodass für $n \geq k$ gilt:

$$a_{n+1} := g(a_{n-k+1}, \ldots, a_n).$$

Beispiel 4.3.1.

1. Sei $(a_n)_{n\in\mathbb{N}^+}$ definiert durch $a_1 := 1$ und $a_{n+1} := 2a_n + 1$ für alle $n \in \mathbb{N}$. In diesem Fall ist $k = 1$ und $g(x) = 2x + 1$.
2. Die *Fibonacci-Folge*

   $$0, 1, 1, 2, 3, 5, 8, 13, 21, \ldots$$

   ist definiert durch
   $$f_0 := 0, \quad f_1 := 1$$
   und
   $$f_{n+1} := f_{n-1} + f_n \quad \text{für alle } n \in \mathbb{N}^+.$$
   Hier ist $k = 2$ und $g(x, y) = x + y$.
3. Durch $a_1 := 1$ und $a_{n+1} := (n + 1) \cdot a_n$ für alle $n \in \mathbb{N}$ wird die *Fakultätsfunktion* $n!\colon \mathbb{N}^+ \to \mathbb{N}$ rekursiv definiert.

Grundlagen Mathematik | 04.08: Rekursive Definitionen, Summenzeichen, Produkt, Fakultät, Fibonacci

Definition 4.3.2. Die Summe

$$a_1 + \cdots + a_n = \sum_{k=1}^{n} a_k = \sum_{k=1,\ldots,n} a_k = \sum_{k\in[n]} a_k$$

von Elementen aus $\mathbb{R}$ ist rekursiv definiert durch

$$\sum_{k=1}^{1} a_k := a_1,$$

und

$$\sum_{k=1}^{n+1} a_k = \left(\sum_{k=1}^{n} a_k\right) + a_{n+1}.$$

Allgemeiner definiert man $\sum_{k=m}^{m} a_k := a_m$ für $m \in \mathbb{N}^+$ und

$$\sum_{k=m}^{n+1} a_k := \left(\sum_{k=m}^{n} a_k\right) + a_{n+1}$$

für $n \in \mathbb{Z}$ mit $n \geq m$. Für $n < m$ verwenden wir die Konvention $\sum_{k=m}^{n} a_k := 0$. Ebenso ist dies für das Produkt möglich.

Das Produkt

$$a_1 \cdot \ldots \cdot a_n = \prod_{k=1}^{n} a_k$$

von k Elementen aus $\mathbb{R}$ ist rekursiv definiert durch

$$\prod_{k=1}^{1} a_k := a_1,$$

und

$$\prod_{k=1}^{n+1} a_k := \left(\prod_{k=1}^{n} a_k\right) \cdot a_{n+1}.$$

Dabei heißt k jeweils der *Laufindex*, 1 ist die *untere Summations-/Produktgrenze* und n die *obere*. Für $n = 0$ definieren wir die *leere Summe* $\sum_{k=1}^{0} a_k := 0$ und das *leere Produkt* $\prod_{k=1}^{0} a_k := 1$.

In ähnlicher Weise wollen wir die Summe bzw. das Produkt über eine Menge von Summationsindizes verstehen: Ist $(a_k)_{k\in\mathbb{N}}$ eine Folge reeller Zahlen und $I \subseteq \mathbb{N}$ eine endliche Teilmenge der Indizes, so ist $\sum_{k\in I} a_k$ die Summer aller Zahlen a_k mit $k \in I$. Z. B. ist $\sum_{A\in\mathcal{P}(\{1,2\})} |A| = |\emptyset|+|\{1\}|+|\{2\}|+|\{1,2\}| = 0+1+1+2 = 4$. Die leere Summe kann durch die Setzung $\sum_{k\in\emptyset} a_k := 0$ konsistent verallgemeinert werden. Ebenso geht man mit dem leeren Produkt vor.

Der Laufindex muss dabei natürlich nicht mit k bezeichnet werden und mit 1 beginnen, so ist z. B.

$$\sum_{k=-2}^{3} 2^{k+1} = 2^{-1} + 2^0 + 2^1 + 2^2 + 2^3 + 2^4 = \sum_{i=1}^{6} 2^{i-2}.$$

Man kann zeigen, dass für die rekursiv definierte Fakultätsfunktion gilt:

$$n! = \prod_{k=1}^{n} k \text{ für alle } n \in \mathbb{N}^+.$$

Mit der Definition des leeren Produktes kann man deshalb festlegen:

$$0! := 1.$$

Für Sie als Referenz haben wir einige wichtige Rechenregeln für das Summenzeichen zusammengefasst. Diese werden Sie in allen Vorlesungen gebrauchen können.

Satz 4.3.3 (Rechenregeln für Summenzeichen)**.** *Seien $n, m \in \mathbb{Z}$, $\alpha, \beta \in \mathbb{R}$ und reelle Zahlen $a_m, a_{m+1}, \ldots$ und $b_m, b_{m+1}, \ldots$ gegeben. Dann gilt:*

- *Unabhängigkeit vom Summationsindex:* $\sum_{k=m}^{n} a_k = \sum_{\ell=m}^{n} a_\ell$.
- *Indexverschiebung:* $\sum_{k=m}^{n} a_k = \sum_{k=m+\ell}^{n+\ell} a_{k-\ell}$ *für jedes* $\ell \in \mathbb{Z}$.
- *Rückwärtssummation:* $\sum_{k=m}^{n} a_k = \sum_{k=m}^{n} a_{n+m-k}$.
- *Linearität:* $\sum_{k=m}^{n} (\alpha a_k + \beta b_k) = \alpha \left(\sum_{k=m}^{n} a_k\right) + \beta \left(\sum_{k=m}^{n} b_k\right)$.
- *Teleskopsumme:* $\sum_{k=m}^{n} (a_{k+1} - a_k) = a_{n+1} - a_m$.
- *Index aufteilen:* $\sum_{k=m}^{n} a_k = \sum_{k=m}^{s} a_k + \sum_{k=s+1}^{n} a_k$, *für* $s \in \mathbb{Z}$ *mit* $m \le s \le n$.

4.4 Abzählbarkeit

Grundlagen Mathematik | 04.09: Abzählbarkeit, Hilberts Hotel, Kardinalität, Q abzählbar unendlich

Wir wollen Mengen gemäß ihrer „Größe“ bzw. der Anzahl der enthaltenen Elemente vergleichen. Dazu müssen wir den in Kapitel 2 etwas lax eingeführten Begriff der Kardinalität präzisieren (für endliche Mengen war uns dieser intuitiv bisher klar).

Definition 4.4.1. Zwei Mengen X und Y heißen *gleichmächtig*, wenn es eine bijektive Abbildung $f : X \to Y$ gibt.

Sehr empfehlenswert hierzu ist das PBS Infinite Series Video *A Hierarchy of Infinities*, das unter https://youtu.be/i7c2qz7s00I abrufbar ist. Ebenfalls äußerst empfehlenswert ist die SWR2 Produktion *SWR2 Wissen | Geniale Mathematiker (1/6) | Georg Cantor und das Universum der Unendlichkeiten* unter https://www.swr.de/swr2/wissen/broadcastcontrib-swr-11868.html.

Sind X und Y also gleichmächtig, so kann man jedem Element aus X ein eindeutiges Element aus Y zuordnen. D. h. man kann feststellen, dass zwei Mengen gleichmächtig sind, ohne ihre Elemente zu zählen (wie würde man das auch bei unendlichen Mengen machen?). Es genügt eine Paarung ihrer Elemente. Will man bspw. prüfen, dass man an jeder Hand gleich viele Finger hat, so muss man diese nicht zählen, sondern jeden Finger der linken Hand auf einen Finger der rechten Hand legen. Das ist das Prinzip hinter obiger Definition.

Doch was sind denn nun endliche und unendliche Mengen? Und gibt es Unterschiede bei unendlichen Mengen?

Intuitiv denkt man vielleicht, dass man die Unendlichkeit versteht. Dass die Intuition beim Hantieren mit der Unendlichkeit keinen Platz hat, zeigt folgendes Beispiel.

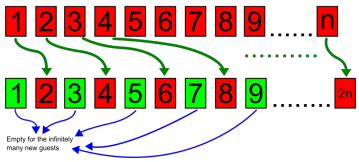


Abbildung 4.8: Hilberts Hotel bei der Anreise eines Busses. Für alle Personen aus dem Bus kann ein Platz geschaffen werden. Quelle: Wikimedia Commons, Jan Beránek / CC BY-SA.

Beispiel 4.4.2 (Hilberts Hotel)**.** David Hilbert betreibt ein Hotel mit unendlich vielen durchnummerierten Zimmern. Jedes Zimmer hat eine Zimmernummer aus $\mathbb{N}^+$. Das Hotel ist sehr beliebt, deshalb sind bereits alle Zimmer belegt.

Ein neuer Gast kommt am Hotel an und möchte gerne noch ein Zimmer. Das ist möglich! Der Portier löst einen Gong aus, und der Gast aus Zimmer 1 wechselt ins Zimmer 2, der aus Zimmer 2 ins Zimmer 3, usw.

Nun kommt sogar eine ganze Wandertruppe – endlich viele Personen – am Hotel an, sagen wir $m \in \mathbb{N}^+$ Auch diese können untergebracht werden. Der Portier wiederholt das obige Verfahren einfach m-mal.

Hilbert betreibt auch ein Busunternehmen. Die Reisebusse können unendlich viele Personen transportieren und haben Sitze, die mit Sitzplatznummern aus $\mathbb{N}^+$ versehen sind. Nun kommt ein solcher vollbeladener Bus am randvollen Hotel an. Der Portier reagiert sofort und schickt alle Hotelgäste mit Zimmernummer n in Zimmer $2n$. Nun ausreichend Platz für die Herde an Busgästen im Hotel.

Hilbert wäre nicht Hilbert, hätte er nur einen solchen Bus: Hilbert hat eine ganze unendliche Flotte dieser Reisebusse, alle haben eine Reisebusnummer aus $\mathbb{N}^+$. Als unendlich viele Reisebusse mit jeweils unendlich vielen Busgästen am Hotel ankommen, kommt der Portier im randvollen Hotel kurz ins Schwitzen. Dann hat er den rettenden Einfall: Zuerst macht er alle ungeraden Zimmernummern frei, indem er, wie oben, alle Gäste von Zimmer n in Zimmer $2n$ schickt. Busgäste aus Bus 1 werden in die Zimmer $3, 9, 27, \ldots$ geschickt; Busgäste aus

Bus 2 in die Zimmer $5, 25, 125, 625, \ldots$; $\ldots$; Busgäste aus Bus i in die Zimmer mit den Nummern

$$p_i, p_i^2, p_i^3, \ldots,$$

wobei p_i die $(i+1)$-te Primzahl ist. Damit wurden alle Gäste untergebracht. Und es sind sogar noch Zimmer frei! Das Zimmer 15 hat z. B. noch keinen Gast, da 15 keine Primzahlpotenz ist.

Definition 4.4.3.
1. Die Menge X heißt *endlich*, wenn es ein $n \in \mathbb{N}$ und eine bijektive Abbildung $f\colon [n] \to X$ gibt. Die *Mächtigkeit* oder *Kardinalität* von X ist in diesem Fall definiert durch $|X| := n$.
2. Die Menge X heißt *abzählbar unendlich*, falls es eine bijektive Abbildung $f\colon \mathbb{N} \to X$ gibt. Die Mächtigkeit von X bezeichnen wir in diesem Fall mit dem Symbol $+\infty$.
3. Die Menge X heißt *abzählbar*, falls X endlich oder abzählbar unendlich ist. Ist eine Menge nicht abzählbar, so sagen wir, dass sie *überabzählbar* ist.

Lemma 4.4.4. *Ist X eine endliche Menge, so ist $|X|$ wohldefiniert (d. h. eindeutig bestimmt).*

Beweis. Es seien $g\colon [n] \to X$ und $h\colon [m] \to X$ zwei bijektive Abbildungen. Wir müssen zeigen, dass $n = m$ ist. Betrachte dazu die Abbildung

$$f := h^{-1} \circ g\colon [n] \to [m].$$

Dann ist f ebenfalls eine bijektive Abbildung. Es folgt $n = m$. □

Hier noch einige Eigenschaften, die wir immer wieder implizit benutzen werden. Wir verzichten auf einen Beweis.

Proposition 4.4.5.
1. *Es seien $m, n \in \mathbb{N}^+$ positive natürliche Zahlen mit $m < n$.*
   a) *Es gibt keine injektive Abbildung $\{1, \ldots, n\} \to \{1, \ldots, m\}$.*
   b) *Es gibt keine surjektive Abbildung $\{1, \ldots, m\} \to \{1, \ldots, n\}$.*
2. *Es sei $n \in \mathbb{N}^+$. Für jede Abbildung*
   $$f\colon \{1, \ldots, n\} \to \{1, \ldots, n\}$$
   gilt:
   $$f \text{ injektiv} \iff f \text{ surjektiv}.$$
3. *Es sei $n \in \mathbb{N}^+$ und $M \subseteq \{1, \ldots, n\}$ eine nicht-leere Teilmenge.*
   a) *Es gibt ein $m \in \mathbb{N}^+$, $m \leq n$, und eine bijektive Abbildung*
   $$f\colon \{1, \ldots, m\} \to M.$$
   b) *Es gilt*
   $$M \neq \{1, \ldots, n\} \iff m < n.$$

Ist X eine echte Teilmenge einer endlichen Menge Y, gilt also $X \subsetneq Y$ (d. h. $X \subseteq Y$ und $X \neq Y$), dann hat X eine geringere Kardinalität als Y, in Zeichen $|X| < |Y|$. Das Überraschende: Das gilt jedoch nicht für

unendliche Mengen. So sind z. B. die Mengen der geraden natürliche Zahlen $2\mathbb{N} := \{2n \mid n \in \mathbb{N}\}$ und die Menge aller naürlichen Zahlen $\mathbb{N}$ gleichmächtig.

Beweisen Sie dies formal als Übung!

Wir kennen auch andere unendliche Mengen. So sind uns die Inklusionen

$$\mathbb{N} \subseteq \mathbb{Z} \subseteq \mathbb{Q} \subseteq \mathbb{R}$$

bekannt. Sind diese Mengen größer als $\mathbb{N}$ oder sind sie auch noch abzählbar, d. h. gleichmächtig wie $\mathbb{N}$? Auf den ersten Blick sehen diese Mengen größer aus als $\mathbb{N}$. Doch man kann leicht zeigen, dass $\mathbb{Z}$ gleichmächtig wie $\mathbb{N}$ ist. Insbesondere die Menge $\mathbb{Q}$ scheint um einiges größer als $\mathbb{N}$ zu sein. Man kann immerhin für zwei gegebene rationale Zahlen $q, p \in \mathbb{Q}$ immer eine weitere rationale Zahl r in deren Mitte finden $q < r < p$. Aber auch hier trügt uns die Intuition!

Auch das sollten Sie zur Übung tun.

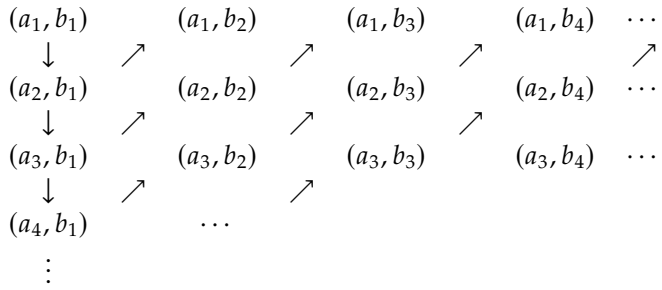
Geben Sie eine explizite Konstruktionsvorschrift für diese Zahl r an.

Theorem 4.4.6. *$\mathbb{Q}$ ist abzählbar unendlich.*

Wir zeigen allgemeiner folgendes Lemma.

Lemma 4.4.7. *Sind X und Y zwei abzählbar unendliche Mengen, so ist das Kartesische Produkt $X \times Y$ wieder eine abzählbar unendliche Menge.*

Beweis von Lemma 4.4.7. Es seien $f : \mathbb{N}^+ \to X$ mit $f(n) = a_n$ und $g : \mathbb{N}^+ \to Y$ mit $g(n) = b_n$ die zwei bijektiven Abzählungen von X und Y. Wir betrachten das Schema



Das *Cauchysche Diagonalverfahren* liefert diese Abzählung in Pfeilrichtung der Elemente von $X \times Y$. Etwas genauer: Wir erhalten eine Bijektion $\Psi : \mathbb{N}^+ \to X \times Y$ mit $\Psi(1) = (a_1, b_1)$, $\Psi(2) = (a_2, b_1)$, $\Psi(3) = (a_1, b_2)$, $\Psi(4) = (a_3, b_1)$, $\Psi(5) = (a_2, b_2)$, $\Psi(6) = (a_1, b_3)$, $\Psi(7) = (a_4, b_1)$, ... □

Auf gut zwei Seiten könnte man sich hierzu eine formale Abbildung definieren und deren Bijektivität zeigen. Wir begnügen uns mit der Intuition. Interessierte Leser seien auf die nicht-klausurrelevante Seite `https://de.wikipedia.org/wiki/Cantorsche_Paarungsfunktion` verwiesen.

Beweisskizze von Theorem 4.4.6 (Cantors erstes Diagonalargument). Die positiven rationalen Zahlen $\mathbb{Q}^+ := \{p/q \mid p, q \in \mathbb{N}^+\}$ können wir die geordneten Paare nach dem Cauchyschen Diagonalverfahren aufgelistet werden, wodurch man eine surjektive Abbildung von $\mathbb{N}^+$ in $\mathbb{Q}^+$ erhält. Durch Weglassen von erweiterten Brüchen, die bereits aufgetreten sind (also, wenn p und q einen gemeinsamen Teiler $t \neq 1$ haben), erhält man eine Liste der echten Brüche (für welche p und q teilerfremd sind). Man kann das Verfahren abschließend auf alle anderen „Quadranten“ von $\mathbb{Q}$ wiederholt anwenden. □

Dieses Verfahren sollte man in einem formalen Beweis noch deutlich präzisieren!

Auch dieses Argument sollte man in einem formalen Beweis noch präzisieren.

↓ **Ende der 6. Vorlesungswoche**

Bisher haben wir schon ab und an die reellen Zahlen – also die „Kommazahlen“ – in Beispiel benutzt, ohne diese jedoch formal eingeführt zu

haben. Um dies im Schnellverfahren nachzuholen, die nächsten Definitionen.

Grundlagen Mathematik | 04.10: Warum 0.999... = 1 ist - Dezimalbruchentwicklung unter der Lupe

Definition 4.4.8. Es sei $(d_k)_{k\in\mathbb{N}}$ eine Zahlenfolge mit $d_0 \in \mathbb{Z}$ und $d_k \in \{0,1,2,3,\dots,9\}$ für alle $k \in \mathbb{N}^+$. Dann heißt die Folge $(a_n)_{n\in\mathbb{N}}$ mit

$$a_n := d_0, d_1 d_2 \dots d_n := d_0 + \sum_{k=1}^{n} \frac{d_k}{10^k}$$

eine *Dezimalbruchentwicklung*. Sie heißt *eigentlich*, falls keine „Periode 9“ vorliegt, d. h. es kein $N \in \mathbb{N}^+$ mit $d_k = 9$ für alle $k \in \mathbb{N}^+$ mit $k \geq N$ gibt.

Man kann – mit einigen Hilfsmittel aus der Analysis – folgendes zeigen:

- Jede Dezimalbruchentwicklung $(a_n)_{n\in\mathbb{N}}$ ist konvergent. Den Grenzwert $a = \lim_{n\to\infty} a_n$ bezeichnen wir mit $a = d_0, d_1 d_2 \dots$. Man könnte auch formulieren: Das Symbol $d_0, d_1 d_2 \dots$ kennzeichnet den Wert der *Reihe* $\sum_{k=0}^{\infty} \frac{d_k}{10^k} := \lim_{n\to\infty} \sum_{k=0}^{n} \frac{d_k}{10^k}$.
- Für jede reelle Zahl $a \in \mathbb{R}$ gibt es genau eine eigentlich Dezimalbruchdarstellung $a = d_0, d_1 d_2 \dots$.

Was steckt hinter dem Verbot von „Periode 9“ in der eigentlichen Dezimalbruchentwicklung?

Schließt man dies nicht aus, so gilt z. B. $0,\overline{9} = 0,9999999\ldots = 1$, also hat die Zahl 1 zwei verschiedene Darstellungen; denn

$$0,9999999\ldots = \lim_{n\to\infty} \sum_{k=1}^{n} \frac{9}{10^k} = \lim_{n\to\infty} \left(1 - \frac{1}{10^n}\right) = 1 - \lim_{n\to\infty} \frac{1}{10^n} = 1.$$

Die Grenzwertrechenregeln, die wir hier anwenden haben Sie bereits in der Analysis kennengelernt. Genauer berechnen hier den Wert einer Reihe.

Diese Zweideutigkeit in der Darstellung wollen wir unterbinden.

Mit diesem kleinen Exkurs kann man folgenden Satz zeigen.

Grundlagen Mathematik | 04.11: Die reellen Zahlen sind überabzählbar unendlich – Beweis

Theorem 4.4.9. *Die Menge der reellen Zahlen $\mathbb{R}$ ist überabzählbar.*

Unsere Idee, um das Theorem zu zeigen, wird ein Widerspruchsargument sein. Anschaulich und etwas lax gesprochen: Egal wie ich es anstellen würde, eine Liste aller Dezimalzahlen zu erstellen, Sie können immer eine Dezimalzahl nennen, die sich nicht auf meiner Liste findet.

Beweis (Cantors zweites Diagonalargument). Es genügt bereits zu zeigen, dass das *Intervall* $[0,1) := \{x \in \mathbb{R} \mid 0 \leq x < 1\}$ überabzählbar ist. Nehmen wir, um einen Widerspruch zu erhalten, an, dass $[0,1)$ abzählbar ist. Nach Definition muss es dann eine Folge $(a_n)_{n\in\mathbb{N}^+}$ geben mit $[0,1) = \{a_n \mid n \in \mathbb{N}^+\}$ – anschaulich gesprochen ist die Folge also eine Auflistung aller möglichen Dezimalzahlen im Intervall $[0,1)$. Für jedes der Folgenglieder a_n betrachten wir die eigentliche Dezimalbruchentwicklung: Wir schreiben $a_n = 0, d_{n,1} d_{n,2} d_{n,3} \dots$ (d. h. $d_{n,i}$ ist die i-te Nachkommastelle der Zahl a_n). Das erlaubt es uns, unsere vermeintliche

Liste aller Dezimalzahlen in einem Schema aufzuschreiben:

$$\begin{array}{lllllll} a_1 &= 0, & d_{1,1} & d_{1,2} & d_{1,3} & d_{1,4} & \dots \\ a_2 &= 0, & d_{2,1} & d_{2,2} & d_{2,3} & d_{2,4} & \dots \\ a_3 &= 0, & d_{3,1} & d_{3,2} & d_{3,3} & d_{3,4} & \dots \\ a_4 &= 0, & d_{4,1} & d_{4,2} & d_{4,3} & d_{4,4} & \dots \\ \vdots & & \vdots & \vdots & \vdots & \vdots & \ddots \end{array}$$

Für $k \in \mathbb{N}^+$ setzen wir nun

$$d_k := \begin{cases} 0, \text{ falls } d_{k,k} \neq 0, \\ 1, \text{ falls } d_{k,k} = 0. \end{cases}$$

Damit unterscheidet sich die Folge der d_k's von allen Diagonalelementen in unserem Schema:

$$d_k \neq d_{k,k} \text{ für alle } k \in \mathbb{N}^+.$$

Also ist die Zahl $a := 0, d_1 d_2 d_3 \cdots \neq a_n$ für alle $n \in \mathbb{N}^+$, weil $0, d_1 d_2 d_3 \dots$ eine eigentlich Dezimalbruchentwicklung ist. Also: $a \in [0, 1)$, aber $a \neq a_n$ für alle $n \in \mathbb{N}^+$ – ein Widerspruch zur Annahme $[0, 1) = \{a_n \mid n \in \mathbb{N}^+\}$. □

Theorem 4.4.10 (Diagonalisierung)**.** *Ist A eine beliebige Menge, so existiert keine Surjektion $f : A \to \mathcal{P}(A)$.*

Grundlagen Mathematik | 04.12: Cantors Diagonalargument zur Potenzmenge und Nicht-Berechenbarkeit

Beweis. Angenommen, $f : A \to \mathcal{P}(A)$ ist surjektiv. Setze

$$D := \{a \in A \mid a \notin f(a)\}.$$

Offensichtlich ist $D \in \mathcal{P}(A)$. Weil f surjektiv ist, existiert also ein $a_0 \in A$ mit $f(a_0) = D$. Es gilt natürlich: entweder $a_0 \in D$ oder $a_0 \notin D$. Aber nach Definition von D gilt

$$a_0 \in D \iff a_0 \notin f(a_0) = D,$$

ein Widerspruch! □

Insbesondere ist $\mathcal{P}(\mathbb{N})$ überabzählbar, denn wenn diese Menge abzählbar wäre, so gäbe es eine Bijektion $\mathbb{N} \to \mathcal{P}(\mathbb{N})$, aber es gibt nicht einmal eine Surjektion, wie wir gerade gezeigt haben.

Das Interessante daran: Wir können die Elemente aus $\mathcal{P}(\mathbb{N})$ in natürlicher Weise mit den Elementen aus $\{0, 1\}^{\mathbb{N}}$ identifizieren: Ist $f \in \{0, 1\}^{\mathbb{N}}$ eine Abbildung, so assoziieren wir mit dieser die Menge $f^{-1}(\{1\})$. Umgekehrt lässt sich zu jeder Menge $A \in \mathcal{P}(\mathbb{N})$ eine Abbildung $f \in \{0, 1\}^{\mathbb{N}}$ definieren mit $f(n) = 1 \iff n \in A$. Man kann zeigen: Diese Zuordnung ist eine Bijektion. Also ist $\{0, 1\}^{\mathbb{N}}$ überabzählbar.

Haben diese abstrakten Überlegungen überhaupt Anwendungen in der Informatik? Sehr wichtige sogar! Eine grundlegende Frage der theoretischen Informatik ist, welche Funktionen wir (z. B. mit einem Java-, C- oder Python-Programm) überhaupt berechnen können.

Korollar 4.4.11. *Es gibt (überabzählbar viele) Abbildungen $f : \mathbb{N} \to \{0,1\}$, die nicht durch ein Programm berechnet werden können.*

Beweis. Egal in welcher Programmiersprache wir schreiben: Jedes Programm ist eine *endliche* Folge von Symbolen aus einer *endlichen* Menge (dem natürlichen Alphabet und Sonderzeichen, wie geschweiften Klammern, etc.). Daher ist die Menge aller Programme abzählbar.

Demnach ist die Menge $B \subseteq \{0,1\}^{\mathbb{N}}$ aller Abbildungen $\mathbb{N} \to \{0,1\}$, welche von einem Programm berechnet werden können, abzählbar. Nach Theorem 4.4.10 ist $\{0,1\}^{\mathbb{N}}$ überabzählbar. Deshalb ist $\{0,1\}^{\mathbb{N}} \setminus B$ nicht-leer, sondern sogar überabzählbar. □

Um dies nachzuvollziehen, versuchen Sie zu zeigen, dass

$$\mathcal{E}(\mathbb{N}) := \{X \subseteq \mathbb{N} \mid X \text{ endlich}\}$$

abzählbar ist.

Der letzte Schritt in unserer Argumentation müsste eigentlich noch weiter formal untermauert werden, indem man zeigt, dass die abzählbare Vereinigung abzählbarer Mengen wieder abzählbar ist. Dies ist letztendlich aber ein einfaches sog. Dovetailing-Argument wie im Chauchyschen Diagonalverfahren.

4.5 Permutationen

Definition 4.5.1 (Permutationen)**.** Ist $f : X \to X$ eine Bijektion und die Menge X endlich und nicht-leer, dann heißt f *Permutation* auf X. Die Menge der Permuationen auf X bezeichnen wir mit $\mathfrak{S}(X)$, d. h. es ist

$$\mathfrak{S}(X) := \{f : X \to X \mid f \text{ ist bijektiv}\}.$$

Grundlagen Mathematik | 04.13: Einführung in Permutationen und das Problem der ENIGMA

In der Praxis kann man sich in der Regel auf den Fall $X := [n]$ und damit auf $\mathfrak{S}_n := \mathfrak{S}(\{1,\dots,n\})$ beschränken.

Ein Element $\sigma \in \mathfrak{S}_n$ ist durch das Tupel $\big(\sigma(1),\dots,\sigma(n)\big)$ eindeutig bestimmt. Umgekehrt gibt es zu jedem Tupel $(x_1,\dots,x_n)$ von paarweise verschiedenen Zahlen $x_i \in \{1,\dots,n\}$ genau eine Permutation $\sigma \in \mathfrak{S}_n$ mit $\sigma(i) = x_i$ für alle $i = 1,\dots,n$. Dies rechtfertigt die Schreibweise

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & \dots & n \\ \sigma(1) & \sigma(2) & \sigma(3) & \dots & \sigma(n) \end{pmatrix}.$$

Beispiel 4.5.2. Ein Beispiel für ein Element $\sigma \in \mathfrak{S}_6$ ist die Permutation

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 1 & 6 & 5 & 2 \end{pmatrix}.$$

In Beispiel 4.5.2 wird 1 auf 3 abgebildet, und 3 auf 1. Das nennt man einen *Zyklus der Länge 2*. Ferner wird 2 auf 4, und 4 auf 6, und 6 auf 2 abgebildet. Dies stellt einen Zyklus der Länge 3 dar. Die Zahl 5 wird nicht verändert. Diese Überlegungen rechtfertigen die *Zyklendarstellung* einer Permutation. In diesem Fall würden wir schreiben:

$$\sigma = (1\,3)\,(2\,4\,6)\,(5).$$

Die Zyklen können in beliebiger Reihenfolge geschrieben werden. Weiter kann innerhalb eines Zyklus die Reihenfolge zyklisch vertauscht werden. Wir können obige Permutation also z. B. auch wie folgt notieren:

$$\sigma = (6\,2\,4)\,(5)\,(3\,1).$$

Beispiel 4.5.3. Gegeben seien die Permutationen $\sigma, \tau \in \mathfrak{S}_9$, wobei

$$\sigma = \begin{pmatrix} 1&2&3&4&5&6&7&8&9 \\ 4&6&2&3&1&5&7&8&9 \end{pmatrix}$$

in Standard-Darstellung und

$$\tau = (5\,3\,1\,8)\,(6)\,(4\,2)\,(7\,9)$$

in Zyklendarstellung gegeben ist. Damit ergibt sich

$$\tau \circ \sigma = \begin{pmatrix} 1&2&3&4&5&6&7&8&9 \\ 2&6&4&1&8&3&9&5&7 \end{pmatrix} = (1\,2\,6\,3\,4)\,(5\,8)\,(7\,9).$$

Beispiel 4.5.4. In Abschnitt 4.4 haben wir gelernt, dass es zu einer endlichen Menge A immer eine natürliche Zahl $n \in \mathbb{N}$ und eine bijektive Abbildung $a\colon \{1, \ldots, n\} \to A$ gibt, die man als die *Aufzählung* von A betrachten kann. Man kann dann schreiben

$$A = \{a_1, \ldots, a_n\} = \{a_k \mid k = 1, \ldots, n\}.$$

Es kommt auf die Reihenfolge der Aufzählung jedoch nicht an: Hat man eine andere Aufzählung, so gibt es eine Permutation $\sigma \in \mathfrak{S}_n$, so dass die andere Aufzählung die Form

$$k \mapsto a_{\sigma(k)}, \quad k = 1, \ldots, n$$

hat. Damit ist

$$A = \{a_1, \ldots, a_n\} = \{a_{\sigma(1)}, \ldots, a_{\sigma(n)}\}.$$

Folgende Aussage werden wir später beweisen. Sie ist eine fundamental Aussage der abzählenden Kombinatorik und spielt überall dort eine wichtige Rolle, wo es darum geht, die Anzahl der möglichen Anordnungen von endlich vielen Objekten zu bestimmen. Typische Anwendungen findet man z. B. in der Wahrscheinlichkeitsrechnung.

Proposition 4.5.5. *Für alle $n \in \mathbb{N}^+$ gilt $|\mathfrak{S}_n| = n!$.*

Beweis. Im Kapitel über Kombinatorik. □

Wenn eine Permutation aus einem einzigen Zyklus besteht, z. B.

$$\sigma = (2\,5\,6\,8\,1\,3\,4\,9\,7),$$

so heißt σ *zyklisch*.

Proposition 4.5.6. *Die Menge $\mathfrak{S}_n$ enthält $(n-1)!$ zyklische Permutationen.*

Beweis. Man kann o. B. d. A. den Zyklus mit 1 beginnen lassen. Die restlichen $n-1$ Zahlen kann man beliebig permutieren. Die Behauptung folgt mit Proposition 4.5.5. □

Ohne Beschränkung der Allgemeinheit wird abgekürzt als *o. B. d. A.*

Definition 4.5.7. Eine Stelle x mit $\sigma(x) = x$ nennt man *Fixpunkt* der Permutation $\sigma \in \mathfrak{S}_n$. Eine Permutation ohne Fixpunkte heißt *fixpunktfrei* (d. h. es gilt $\sigma(i) \neq i$ für alle $i = 1, \ldots, n$).

Beispiel 4.5.8. Die identische Abbildung mit $\sigma(i) = i$ für $i = 1, \ldots, n$ ist ebenfalls eine Permutation. Sie besitzt die Zyklendarstellung

$$\sigma = (1)\,(2)\,(3)\,\ldots\,(n).$$

Jede Zahl $i = 1, \ldots, n$ ist Fixpunkt dieser Permutation.

Eine fixpunktfreie Permutation ist also eine Permutation, bei der kein Element seine Ausgangsposition behält. Mit anderen Worten tritt kein Zyklus der Länge 1 auf.

Das Beispiel 4.5.9 ist nicht klausurrelevant, aber von enormer historischer Bedeutung, wie sie in Beispiel 4.5.10 erfahren werden.

Beispiel 4.5.9. Wir bezeichnen mit d_n die Anzahl der fixpunktfreien Permutationen in $\mathfrak{S}_n$. Ist $\sigma \in \mathfrak{S}_n$ mit $n \geq 3$ eine fixpunktfreie Permutation, dann gilt per Definition $\sigma(1) \neq 1$. Nun können wir die folgenden zwei Fälle unterschieden:

- Befindet sich die Zahl 1 an der Stelle $j = \sigma(1)$, dann können die übrigen $n - 2$ Zahlen auf d_{n-2} Möglichkeiten fixpunktfrei auf die verbleibenden Plätze verteilt werden.
- Ansonsten betrachtet man die Menge $\{1, \ldots, n\} \setminus \{j\}$. Diese Zahlen müssen nun die Positionen $2, 3, \ldots, n$ einnehmen, sodass keine der Zahlen festbleibt und zudem die 1 nicht an der Stelle j steht. Die Anzahl der Möglichkeiten dies zu erreichen ist gerade d_{n-1}.

Nachdem es $n - 1$ mögliche Werte für j gibt, folgt daraus die lineare Rekurrenz

$$d_n = (n-1)(d_{n-1} + d_{n-2})$$

mit $d_1 = 0$ (die einzige Permutation in $\mathfrak{S}_1$ bildet 1 auf 1 ab und ist damit nicht fixpunktfrei) und $d_2 = 1$ (in $\mathfrak{S}_2$ gibt es zwei Permutationen, wovon nur eine fixpunktfrei ist).

Mit der Technik der Induktion, die wir im nächsten Kapitel kennenlernen, kann man zeigen, dass der Anteil der fixpunktfreien Permutationen für $n \geq 4$ bei etwa 37 % liegt. Etwas genauer gilt

$$\lim_{n\to\infty} \frac{d_n}{n!} = \frac{1}{\mathrm{e}},$$

wobei $\mathrm{e} := \lim_{n\to\infty} \left(1 + \frac{1}{n}\right)^n \approx 2{,}71828\ldots$ die *Eulersche Zahl* ist – eine der wichtigsten mathematischen Konstanten.

Warum interessiert man sich überhaupt für fixpunktfreie Permutationen. Dazu schauen wir uns zwei Beispiele an.

Beispiel 4.5.10. Die Verschlüsselungsmaschine ENIGMA, die während des Zweiten Weltkriegs bei den Nazis zum Einsatz kam, führte konstruktionsbedingt fixpunktfreie Permutationen durch, die zudem selbstinvers sind (d. h. für die $\sigma = \sigma^{-1}$ gilt). Eine spezielle Walze, nämlich die ganz links liegende Umkehrwalze, bewirkte, dass der

Strom den Walzensatz zweimal durchfloss, einmal in Hinrichtung und einmal in Rückrichtung. Dadurch konnte ein Buchstabe nicht mehr in sich selbst verschlüsselt werden, was zwar die Konstruktion und Bedienung der Maschine vereinfachte, da Verschlüsselung und Entschlüsselung hierdurch gleich waren, zugleich allerdings eine signifikante kryptographische Schwächung bewirkte. Wir haben gesehen, dass damit nur noch ca. 37 % der möglichen Schlüssel in Frage kamen.

Beispiel 4.5.11. Das *Wichteln* ist ein vorweihnachtlicher Brauch, bei dem eine Gruppe von Personen auf zufällige Weise Geschenke austauscht (und es meist viele Enttäuschte gibt). Nimmt man dabei an, dass sich keine Person selbst beschenkt, kann der Austausch der Geschenke mathematisch als fixpunktfreie Permutation der Personen beschrieben werden.

Es ist für alle Anwendungen in der Informatik sehr wichtig, sich das unglaublich starke Wachstum der Mengen $\mathfrak{S}_n$ vor Augen zu führen. So ist z. B. die Anzahl der Permutationen einer Menge mit 20 Elemente gleich

$$|\mathfrak{S}_{20}| = 20! = 2\,432\,902\,008\,176\,640\,000.$$

Es ist sogar

$$70! > 10^{100},$$

d. h. die Anzahl der Permutationen auf 70 Elementen übersteigt die Anzahl der Atome im Universum. Kein Computer (auch kein zukünftiger!) kann daher jemals alle diese Permutationen gleichzeitig im Speicher halten. Ebenso problematisch wäre es, wenn man ein Programm schreibt, das alle diese Permutationen nacheinander auf eine bestimmte Eigenschaft überprüft. Die Laufzeit wäre viel zu gigantisch! In der Informatik geht es auch darum „schlauere Lösungen“ für solche Probleme zu finden, bzw. ggfs. zu zeigen, dass man ein bestimmtes Problem nicht schneller lösen kann. Betrachten wir dazu ein Beispiel:

GdM | 04.14: Traveling Salesman Problem, Erzeugung aller Permutationen in lexikographischer Ordnung

Heutzutage machen wir einen Großteil unserer Einkäufe im Internet. Die Lieferdienste stehen vor dem Problem, dass die Vielzahl von Paketen möglichst effizient ausgeliefert werden. Im simpelsten Fall haben wir also folgendes Problem: Ein Fahrer muss während seiner Tour die Kunden $A, B, C, \ldots$ anfahren und am Ende zum Depot zurückfahren. Je nachdem in welcher Reihenfolge die Kunden angefahren werden, legt er eine kürzere oder längere Strecke zurück. Kürzer ist natürlich besser, denn dann wird weniger Benzin verbraucht und der Fahrer kann vielleicht noch eine zweite Tour starten, und so mehr Pakete pro Tag ausliefern. Dies ist das *Traveling Salesman Problem* (Problem des Handlungsreisenden), kurz TSP. Auf den ersten, mathematischen Blick erscheint das TSP trivial. Es gibt nur endlich viele Rundtouren, d. h. es ist theoretisch möglich alle aufzulisten und die kürzeste auszuwählen. Realistisch ist dieses Vorgehen jedoch nicht. Bereits für nur 11 Städte gibt es etwa 2 Millionen Touren! Eine Problemstellung in der Informatik (genauer: der kombinatorischen Optimierung) ist es daher einen effizienten Algorithmus zu konstruieren, der in vertretbarer Zeit die beste oder zumindest eine gute Rundtour findet.

Abbildung 4.9: Kürzester Rundreiseweg durch die 15 größten Städte Deutschlands (14 sind genannt, Dortmund fehlt). Insgesamt sind $14!/2 = 43\,589\,145\,600$ verschiedene Wege möglich.

Tatsächlich kann man zeigen, dass TSP ein sog. NP-*vollständiges* Problem ist. Ein erstes solches Problem hatten wir in der Videolektion zum Quine-McCluskey Verfahren gesehen. Falls Sie Interesse haben, ist folgendes Video sehr empfehlenswert, um mehr über Komplexität von Problemen und eines der spannendsten Probleme der Informatik und Mathematik zu erfahren: `https://youtu.be/YX40hbAHx3s`.

Um allerdings einen ersten Eindruck von den Schwierigkeiten und Herausforderungen in der kombinatorischen Optimierung zu erhalten, und um einen ersten Algorithmus kennenzulernen und zu analysieren, werden wir einen naiven Algorithmus für das TSP formulieren. Dieser Algorithmus ist der offensichtliche: Wir listen alle Touren auf und suchen die beste heraus. Dazu muss das Problem zunächst modelliert werden, d. h. auf das mathematisch Wesentliche reduziert werden.

TSP

Instanz: Ein Menge von Punkten $p_1, \ldots, p_n \in \mathbb{R}^2$.
Aufgabe: Finde eine Permutation $\sigma \in \mathfrak{S}_n$, so dass

$$\sum_{i=1}^{n-1} \operatorname{dist}(p_{\sigma(i)}, p_{\sigma(i+1)}) + \operatorname{dist}(p_{\sigma(n)}, p_{\sigma(1)})$$

minimal ist.

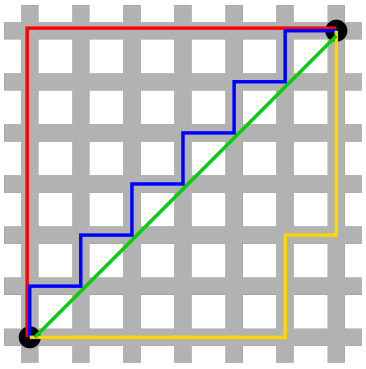


Abbildung 4.10: Die Linien in rot, blau und gelb sind drei Beispiele für die Manhattan-Distanz zwischen den zwei schwarzen Punkten (je 12 Einheiten lang); die grüne Linie stellt zum Vergleich den Euklidischen Abstand dar, der eine Länge von $6\sqrt{2} \approx 8.5$ Einheiten hat.

Hierbei sind verschiedene Abstandsfunktionen denkbar, etwa der *euklidische Abstand*

$$\operatorname{dist}\left(\binom{x_1}{x_2}, \binom{y_1}{y_2}\right) = \left\|\binom{x_1}{x_2}, \binom{y_1}{y_2}\right\|_2 := \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2}$$

oder der Abstand in der sogenannten *Manhattan-Metrik*,

$$\operatorname{dist}\left(\binom{x_1}{x_2}, \binom{y_1}{y_2}\right) = \left\|\binom{x_1}{x_2}, \binom{y_1}{y_2}\right\|_\infty := |y_1 - x_1| + |y_2 - x_2|.$$

Wenn wir das naive Durchtesten aller Routen auf einem Computer implementieren wollen, müssen wir es also schaffen, alle Permutationen zu erzeugen. In einem zweiten Schritt, den wir hier weglassen, da er trivial ist, muss dann die Permutation ermittelt werden, die die kürzeste Rundtour gibt.

Ein ganz naiver Weg, alle Permutationen zu mit einem Computer erzeugen wäre ein sogenannter *Top-Down* Ansatz mit *Rekursion*. Das könnte wie folgt funktionieren: Das Programm wählt das erste Element und führt dann eine sogenannte *Rekursion* durch, in dem es das zweite Element der Permutation aus den verbliebenen Zahlen auswählt, usw. Das Problem an diesem Vorgehen: Es ist gar nicht so leicht zu programmieren: Man muss sich um die Rekursion kümmern, den Rekursionsspeicher geschickt verwalten und dafür sorgen, dass doppelte Elemente übersprungen werden. Außerdem ist nicht wirklich klar, in welcher Reihenfolge die Permutationen erzeugt werden (das ist für unsere Anwendung nicht direkt wichtig, man kann sich aber auch Anwendungen vorstellen, wo dies eine Rolle spielt).

Daher wollen wir unsere Permutationen bzgl. der lexikographischen Ordnung sortiert ausgeben. Dazu benötigen wir ein kleines Hilfswerkzeug, und zwar die *lexikographische Ordnung* auf n-Tupeln reeller Zahlen.

Definition 4.5.12. Sind $a = (a_1, \ldots, a_n)$ und $b = (b_1, \ldots, b_n)$ zwei verschiedene Elemente in $\mathbb{R}^n$, so schreiben wir $a \prec_{\text{lex}} b$, falls es ein $i \in \mathbb{N}^+$ gibt, so dass $a_j = b_j$ für alle $j \in [i-1]$ aber $a_i < b_i$. Hierbei sei $[0] := \emptyset$.

Beispielsweise ist $(1,2,3) \prec_{\text{lex}} (1,3,3)$ und $(1,2,3,4,5) \prec_{\text{lex}} (1,2,3,5,1)$.

Die folgenden Ideen gehen auf den Mathematiker Narayana Panditas im Indien des 14. Jahrhunderts zurück.

Wir wollen zuerst überlegen, wie wir alle „Umsortierungen“ einer Folge, z. B.

$$(0,1,2,5,3,3,0),$$

die wir als fortlaufendes Beispiel benutzen wollen, in lexikographischer Ordnung erzeugen können. Danach werden wir zeigen, dass wir unser Problem auch richtig gelöst haben. Das ist prinzipiell ein zu empfehlendes Vorgehen beim Algorithmenentwurf.

Wie gelingt es uns jetzt nun die nächste Umsortierung zu erzeugen? Das ist die Kernidee unseres Algorithmus: Wir müssen die Folge bzgl. der lexikographischen Ordnung „so wenig wie möglich“ erhöhen. Das funktioniert so ähnlich wie beim Zählen in Zahlensystemen zur einer Basis b (siehe Tutorium 03). So macht es bspw. keinen Sinn das erste Element der Folge von einer 0 zu einer 1 umzuwandeln, denn wenn wir den *Präfix* der Folge von $(0,1)$ zu $(0,2)$ abändern, erhalten wir eine noch „nähere“ Umsortierung. Tatsächlich macht es auch keinen Sinn das zweite Element zu ändern. Aus dieser Beobachtung werden wir unsere Idee entwickeln.

Wir identifizieren zunächst den längsten *Suffix*, der nicht-wachsend ist. Man beachte, dass ein solcher Suffix immer mindestens ein Element hat, denn jeder *Substring* aus einem einzelnen Element ist trivialerweise nicht-wachsend. In unserem laufenden Beispiel wäre das $(5,3,3,0)$. Dieser Suffix ist bereits die bzgl. der lexikographischen Ordnung größte Umsortierung der Zahlen $5,3,3$ und 0. D. h. durch Modifikation dieses Suffixes wird es uns nicht gelingen die Umsortierung $(0,1,2,5,3,3,0)$ lexikographisch zu erhöhen. Wir müssen dazu also ein Element zur Linken des Suffix ändern.

Als nächstes betrachten wir das Element unmittelbar zur Linken des gefundenen Suffixes. In unserem fortlaufenden Beispiel wäre dies 2. Wir nennen dieses Element *Pivot*. Falls es solch ein Element nicht mehr geben sollte, dann haben wir bereits die letzte Umsortierung bzgl. der lexikographischen Ordnung gefunden. Das Pivot-Element ist notwendigerweise geringer als das erste Element des Suffixes (im Beispiel war dies 5). Das bedeutet aber, dass es mindestens ein Element im Suffix geben muss, das größer als das Pivot-Element ist. Wenn wir das Pivot-Element mit dem kleinsten Element im Suffix tauschen, das immer noch größer als das Pivot-Element ist, dann haben wir den Präfix (also alles in der Folge bis auf den Suffix) minimiert. Im unserem Beispiel würden wir den neuen Präfix $(0,1,3)$ und den neuen Suffix $(5,3,2,0)$ erhalten. Man beachte: Wenn der Suffix mehrere Kopien des neuen Pivot-Elements enthält, dann sollte man dasjenige wählen, das am weitesten Rechts steht. Das spielt in unserem nächsten Schritt eine Rolle.

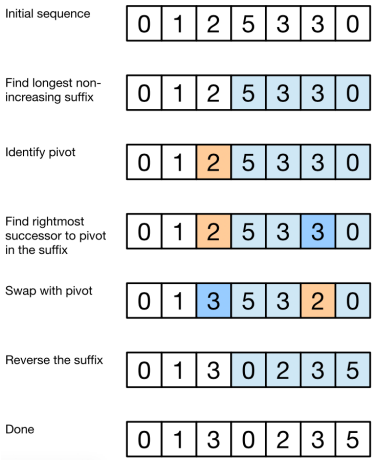


Abbildung 4.11: Eine Skizze zur Algorithmenidee. Die Skizze und die dazugehörige Herleitung stammen von Project Nayuki. Nayuki hat uns freundlicherweise die Lizenz hierzu für diese Vorlesung gegeben.

Abschließend sortieren wir das Suffix in nicht-fallender Ordnung – denn wir haben ja den Präfix vergrößert, also wollen wir den Suffix so klein wie möglich halten (wie beim Zählen in Zahlensystemen). Tatsächlich muss man dazu nicht extra ein Programm schreiben, dass diese Zahlen sortiert. Es genügt, die Reihenfolge des Suffixes umzudrehen, denn das ausgetauschte Element fügt sich in die nicht-wachsende Ordnung ein. Wir erhalten also die Folge

$$(0, 1, 3, 0, 2, 3, 5),$$

welche die nächste Umsortierung ist, die wir erhalten wollten.

Was wir also gemacht haben, ist kurz gefasst folgendes: Wir hatten ein sogenanntes *Array* mit Zahlen als Einträge. Dann haben wir folgende Schritte ausgeführt:

- Finde den größten Index i mit $\text{array}[i] < \text{array}[i+1]$.
  (Falls kein solches i existiert, haben wir bereits die letzte Umsortierung gefunden).
- Finde den größten Index ℓ mit $\ell \geq i$ und $\text{array}[i] < \text{array}[\ell]$.
- Tausche $\text{array}[i]$ mit $\text{array}[\ell]$.
- Invertiere den Suffix, der mit $\text{array}[i+1]$ beginnt.

Notieren wir das ganze zur Analyse noch einmal in durchnummerierter Form. Dabei beschränken wir uns ab hier wieder auf Permutationen. Dazu dient Zeile 1.

1: Setze $a_i = i$ für $i = 0, \ldots, n$
2: Gebe $a_1, \ldots, a_n$ aus
3: Ermittle maximales $i \leq n-1$ mit $a_i < a_{i+1}$
4: Halte an, falls $i = 0$
5: Ermittle maximales ℓ mit $a_i < a_\ell$
6: Tausche $a_i \leftrightarrow a_\ell$
7: Invertiere die Reihenfolge $a_{i+1}, \ldots, a_n$
8: Gehe zu Zeile 2

Warum tut der Algorithmus aber das, was wir von ihm wollen? Das sollten wir zuerst beweisen! Man sagt auch, dass man die *Korrektheit* des Algorithmus beweist.

Damit können wir nun beweisen, dass der Algorithmus alle Permutationen auf $\{1, \ldots, n\}$ ausgibt.

Beweis. Wir wollen sogar beweisen, dass der Algorithmus die Permutationen gemäß der lexikographischen Ordnung sortiert ausgibt. Wir stellen dazu zuerst fest, dass die „Startpermutation“ $(1, 2, 3, \ldots, n)$ tatsächlich die kleinste bezüglich $<_{\text{lex}}$ ist. Zudem hält der Algorithmus genau dann, wenn er auf die Permutation $(n, n-1, \ldots, 2, 1)$ stößt. Dies ist die lexikographisch größte Permutation.

Sei nun $a = (a_1, \ldots, a_n)$ eine vom Algorithmus in Zeile 2 ausgegebene Permutation und sei $a' = (a'_1, \ldots, a'_n)$ die nächste ausgegebene Permutation. Wir wollen nachvollziehen, dass a und a' aufeinanderfolgend in

der lexikographischen Ordnung sind. Nach Wahl von i und ℓ gilt:

$$\begin{array}{rl} a = a_1, \ldots, a_{i-1}, & \boldsymbol{a_i} < a_{i+1} > \ldots > a_{\ell-1} > \boldsymbol{a_\ell} > a_{\ell+1} > \ldots > a_n \\ & \wedge \qquad\qquad \downarrow \text{ Zeile 6} \\ a_1, \ldots, a_{i-1}, & \boldsymbol{a_\ell} < a_{i+1} > \ldots > a_{\ell-1} > \boldsymbol{a_i} > a_{\ell+1} > \ldots > a_n \\ & \qquad\qquad \downarrow \text{ Zeile 7} \\ a' = a_1, \ldots, a_{i-1}, & \boldsymbol{a_\ell} > \boldsymbol{a_n} < \ldots < \boldsymbol{a_{\ell+1}} < \boldsymbol{a_i} < \boldsymbol{a_{\ell-1}} < \ldots < \boldsymbol{a_{i+1}} \end{array}$$

Nach Ausführung von Zeile 6 ist $a_\ell < a_{i+1}$, denn sonst wäre $a_\ell > a_{i+1}$, ein Widerspruch (denn wir wissen $a_{i+1} > \ldots > a_{\ell-1} > a_\ell$). Ebenso ist $a_{\ell-1} > a_i$, denn sonst wäre $a_{\ell-1} < a_i$, was ebenfalls ein Widerspruch ist (denn Zeile 5 liefert zusammen mit der Initialisierung von a: $a_i < a_\ell < a_{\ell-1}$).

Bei Betrachtung des Schaubilds stellen wir fest: $a \prec_{\text{lex}} a'$.

Nehmen wir nun an, es gäbe eine Permutation b mit $a \prec_{\text{lex}} b \prec_{\text{lex}} a'$. Dann können wir schreiben $b = (a_1, \ldots, a_{i-1}, b_i, \ldots, b_n)$ für gewisse $b_i, \ldots, b_n$, da b in den ersten $i-1$ Stellen mit a und a' übereinstimmen muss. Da nun aber $a \prec_{\text{lex}} b$ gilt, gibt es einen Index $r \geq i$ mit $a_t = b_t$ für $t = 1, \ldots, r-1$ und $a_r < b_r$. Angenommen, $r > i$. Die Ungleichung $b_r \neq a_r$ und der Fakt, dass b eine Permutation ist, bedingen $b_r \in \{a_{r+1}, \ldots, a_n\}$. Die Zahlen $a_{i+1} > a_{i+2} > \ldots > a_n$ sind jedoch nach Definition von i in Zeile 3 absteigend sortiert, woraus $a_r > b_r$ folgt. Mit diesem Widerspruch erhalten wir $a_i < b_i$.

Betrachten wir nun b und a'. Weil $b \prec_{\text{lex}} a'$, gibt es einen Index $s \geq i$, so dass $b_t = a'_t$ für alle $t = 1, \ldots, s-1$ und $b_s < a'_s$. Angenommen, $s > i$. Wiederum ist $b_s \in \{a'_{s+1}, \ldots, a'_n\}$, da es sich bei b um eine Permutation handelt. Jedoch ist $a'_{i+1} < a'_{i+2} < \ldots < a'_n$ eine aufsteigende Folge, was $b_s < a'_s$ unmöglich macht. Folglich ist $b_i < a'_i$.

Insgesamt erhalten wir $a_i < b_i < a'_i = a_\ell$. Doch a_ℓ ist nach Wahl das kleinste Element aus $a_{i+1}, \ldots, a_n$, das immer noch größer als a_i ist. Da $b_i \in \{a_i, \ldots, a_n\}$ ist also $a_i < b_i < a_\ell$ unmöglich. □

Obiger Algorithmus ist natürlich ein wenig informell. Wenn wir ihn tatsächlich auf einem Computer ausführen wollten, müssten wir, je nach Programmiersprache, einige Details hinzufügen. Dennoch ist es ein leichtes obige Skizze in einen (größtenteils) sprach-unabhängigen Pseudocode umzuformulieren. Dabei wollen wir eine leichte Verallgemeinerung vornehmen: Auf Eingabe von Zahlen $(a_1, \ldots, a_n)$ soll unser Programm alle Permutationen von $\Xi(\{a_1, \ldots, a_n\})$ erzeugen. Dabei nehmen wir an, dass die Zahlen bereits aufsteigend sortiert sind, also $a_1 \leq \cdots \leq a_n$ gilt. Hierbei sei zugelassen, dass manche Zahlen gleich sind.

PERMUTATIONS$(a_1, \ldots, a_n)$

```
1  repeat
2      VISIT(a_1, ..., a_n)
3      setze i := n − 1
4      while a_i ≥ a_{i+1} do setze i := i − 1
5      if i > 0 then
6          setze ℓ := n
7          while a_i ≥ a_ℓ do setze ℓ := ℓ − 1
8          vertausche a_i ↔ a_ℓ
9          setze r := i + 1 und s := n
10         while r < s do
11             vertausche a_r ↔ a_s
12             setze r := r + 1 und s := s − 1
13 until i = 0
```

Statt *setze* $i := n - 1$ schreibt man auch $i \leftarrow n - 1$.

Die aktuell berechnete Permutation wird an das Unterprogramm Visit übergeben. In Visit ist dann eine entsprechende Anwendung programmiert. Dies könnte zum Beispiel einfach eine Print-Anweisung für den Bildschirm sein, oder die Kalkulation der Länge der resultierenden Rundtour durch alle größeren Dörfer Finnlands.

Kleiner Fun-Fact: Wird die Eingabesequenz als eine Zahl zur Basis $a_n + 1$ gelesen, hier also $(123)_4 = 27$, so ist dies die kleinste Zahl, die man mit den gegebenen Zahlen bilden kann. Die jeweils nächste Permutation ist immer die nächst größere Zahl, die man bilden kann.

Sei beispielsweise $a_i := i$ für $i = 1, \ldots, n$, wie zuvor. Dann erzeugt Permutations$(1, 2, 3)$ die sechs Permutationen aus $\mathfrak{S}_6$ in folgender Reihenfolge:

$$123, \quad 132, \quad 213, \quad 231, \quad 312, \quad 321.$$

Es wäre nun ein leichtes Pseudocode in eine echte Programmiersprache zu übersetzen. Wir haben das hier beispielhaft für die Sprache C getan, die Sie gerade in der Vorlesung *Strukturierte Programmierung* lernen.

Die Sprache C ist für diese Vorlesung nicht klausurrelevant. Diese Ausführungen hier dienen nur der Veranschaulichung.

```c
#include <stdio.h>
#include <stdbool.h>
#include <stddef.h>

/*
* Computes the next lexicographical permutation of the specified
* array of integers in place, returning a Boolean to indicate
* whether a next permutation existed.
* (Returns false when the argument is already the last possible
* permutation.)
*/
bool next_permutation(int array[], size_t length) {
    // Find non-increasing suffix
    if (length == 0)
        return false;
    size_t i = length - 1;
    while (i > 0 && array[i-1] >= array[i])
        i--;
    if (i == 0)
        return false;

    // Find successor to pivot
    size_t l = length - 1;
    while (array[l] <= array[i - 1])
        l--;
    int temp = array[i - 1];
    array[i - 1] = array[l];
    array[l] = temp;

    // Reverse suffix
    size_t r = i;
    size_t s = length - 1;
    while (r < s) {
        temp = array[r];
        array[r] = array[s];
        array[s] = temp;
        r++;
        s--;
    }
    return true;
}
```

Der Aufruf kann wie folgt erfolgen:

```
43  int main() {
44      // Must start at lowest "permutation"!
45      int array[5] = {0, 1, 1, 1, 4};
46
47      size_t len = sizeof(array)/sizeof(int);
48
49      do {
50          for(int a = 0; a < len; a++) {
51              printf("%d ", array[a]);
52          }
53      printf("\n");
54      } while (next_permutation(array,len));
55
56      return 0;
57  }
58
```

Man kann das Programm z. B. online auf https://onlinegdb.com/S1gBMXQbj8 testen, oder per Download der main.c Datei unter den Modulen.

4.6 O-Notation

Grundlagen Mathematik | 04.15: O-Notation zur Algorithmenanalyse erklärt

Eine Analyse eines Algorithmus besteht aus zwei Teilen. Zunächst muss überprüft werden, dass der Algorithmus *korrekt* ist, d. h. dass er das ihm gestellte Problem auch löst. Zweitens sind wir an der Laufzeit des Algorithmus interessiert.

Doch wie misst man die Laufzeit eines Algorithmus? Man könnte ihn in einer echten Programmiersprache (z. B. Java oder Python) implementieren und auf verschieden Instanzen testen. Dann könnte man die Messergebnisse in Diagrammen visualisieren. Dieser Weg hat jedoch schwerwiegende Nachteile: Um zwei verschiedene Algorithmen zu vergleichen, müssten beide unter exakt identischen Bedingungen ausgeführt werden (d. h. auf der gleichen Maschine, mit den gleichen Instanzen, es muss gewährleistet sein, dass die Implementierungen beider Algorithmen mit der gleichen Sorgfalt gemacht wurde). Außerdem hängen die Laufzeiten dann immer noch von den konkret ausgewählten Testinstanzen ab. Dies alles schwächt die Aussagekraft solcher Vergleiche erheblich. Wir werden stattdessen einen theoretischeren Ansatz wählen.

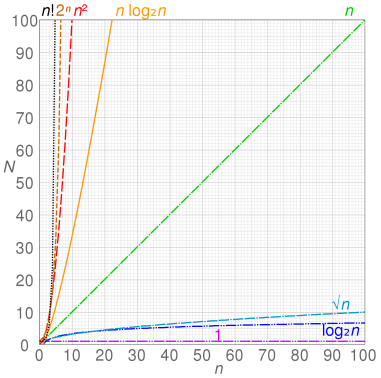


Abbildung 4.12: Die Graphen von Funktionen, die häufig in der Analyse von Algorithmen verwendet werden. Abgetragen ist die Anzahl an Operationen bei einer gegebenen Input-Größe. Quelle: Wikimedia Commons, Cmglee / CC BY-SA (https://creativecommons.org/licenses/by-sa/4.0)

Dazu stellen wir uns vor, dass wir unsere Algorithmen auf idealisierten Computern implementieren. Diese zählen nur die Anzahl der elementaren Operationen. Und davon wird uns auch nur das asymptotische Verhalten interessieren, also wie sich die Laufzeit bei immer größer werdenden Eingaben verhält. Das genaue Modell führt uns in die Tiefen der theoretischen Informatik. Deshalb nehmen wir an dieser Stelle lediglich an, dass das verwendete Modell vernünftig ist und über unendlich viel Speicher verfügt. Unter anderem nehmen wir an, unser Rechner kann folgende *elementare Operationen* in einer Zeiteinheit ausführen:

- Arithmetische Operationen auf zwei ganzen oder rationalen Zahlen (von beschränkter Größe, wie Addition, Subtraktion, Multiplikation, Division, Division mit Rest usw.)
- Speicheroperationen (beschränkter Größe): Laden, speichern, kopieren usw.
- Kontrolloperationen: if-then, Schleifen usw.

Damit lässt sich für unseren Algorithmus folgender Satz beweisen.

Satz 4.6.1. *Es gibt eine Konstante $c \in \mathbb{R}$, so dass der Algorithmus höchstens $c \cdot n \cdot n!$ elementare Operationen ausführt.*

Beweis. Bei den Zeilennummern beziehen wir uns auf die erste Version des Algorithmus im Pseudocode. Zeile 1 benötigt sicher höchstens 10(n+1) elementare Operationen. Die Schleife Zeile 2 – 8 wird genau einmal per Permutation ausgeführt, also $n!$-mal. Die Ausgabe in Zeile 2 benötigt höchstens $10n$ Operationen. Jede while-Schleife wird in höchstens $10n$ Operationen ausgeführt. Alle anderen Zeilen brauchen sogar höchstens 10 elementare Operationen. Insgesamt also haben wir $\leq 100nn!$ Operationen. □

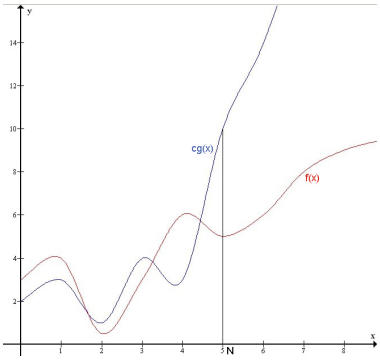


Abbildung 4.13: Skizze zur O-Notation.

In Satz 4.6.1 haben wir keine Mühe gemacht, die Konstante c genauer zu bestimmen, und es wäre auch unsinnig. Wie viele Operationen der Algorithmus nun genau benötigt, hängt sehr eng mit dem genauen Rechnermodell zusammen, das wir noch nicht einmal richtig präzisiert haben. Anstatt dies nachzuholen und das Rechnermodell bis ins kleinste zu diskutieren, schlagen wir einen anderen Weg ein. Wir vereinbaren, dass wir nur an der *Asymptotik* der Laufzeiten interessiert sind. Das bedeutet, dass ein Algorithmus, der bei einer Eingabe der Größe n, nach n Schritten anhält für uns genau so gut ist wie ein Algorithmus, der $10^{10}n$ Operationen benötigt. Und letzterer Algorithmus soll für uns sogar überlegen sein gegenüber einem Algorithmus, der $n^{1.01}$ Operationen braucht.

Hierzu gibt man die Laufzeit in O-Notation an. Siehe Abbildung Abbildung 4.13.

Man stelle sich vor, dass diese Funktionen die Anzahl der Operationen je Eingabegröße zählen.

Definition 4.6.2. Es seien $f, g \colon \mathbb{N}^+ \to \mathbb{R}^+ := \{x \in \mathbb{R} \mid x > 0\}$ zwei Funktionen. Wir schreiben $f \in \mathrm{O}(g)$, falls es eine reelle Konstante $c > 0$ und ein $N \in \mathbb{N}^+$ gibt, so dass $f(n) \leq cg(n)$ für alle $n \geq N$.

Man kann $f \in \mathrm{O}(g)$ so verstehen, dass f höchstens so stark wie g wächst (ab einem gewissen Punkt).

Beispiel 4.6.3. Es gilt $2n + 1 \in \mathrm{O}(n)$. Betrachte dazu z. B. $c = 3$ und $N = 1$. Es gilt nämlich $2n + 1 \leq 3n$ für alle $n \geq 1$.

Beispiel 4.6.4. $10^{90}n^5 \in \mathrm{O}(n^5)$, $10^{1000}n^5 \in \mathrm{O}(n^6)$, und $\log n \in \mathrm{O}(\sqrt{n})$.

Manchmal will man ausdrücken, dass f mindestens so stark wächst wie g. Man schreibt $f \in \Omega(g)$, falls es eine reelle Konstante $c > 0$ und ein $N \in \mathbb{N}^+$ gibt, so dass $cg(n) \leq f(n)$ für alle $n \geq N$.

Schließlich setzt man $\Theta(g) := \mathrm{O}(g) \cap \Omega(g)$.

Damit können wir Satz 4.6.1 umformulieren: Unser Algorithmus zur Erzeugung aller Permutationen führt $O(nn!)$ elementare Operationen durch.

Hier ein weiteres Beispiel zu dieser Notation: Gegeben sei die endliche Menge $A = \{a_1, \ldots, a_n\}$ auf der eine binäre Relation $R \subseteq A \times A$ in Form eines Graphen gegeben ist. Wir wollen die transitive Hülle $M^+ = (m^+_{i,j})_{i,j=1,\ldots,n}$ der Adjazenzmatrix $M = (m_{i,j})_{i,j=1,\ldots,n}$ berechnen, welche die Relation R darstellt. Zur Erinnerung: Es ist $m_{i,j} = 1 \iff a_i R a_j$. Es soll später gelten: $m^+_{i,j} = 1$ genau dann, wenn ein Pfad von a_i nach a_j existiert. Wir haben schon erfahren, dass wir das mit dem (Floyd)-Warshall-Algorithmus tun können.

WARSHALL(M)

```
1  for k = 1 to n do
2    for i = 1 to n do
3      if m_{i,k} = 1 then
4        for j = 1 to n do
5          if m_{k,j} = 1 then m_{i,j} := 1
```

Die Laufzeit (Komplexität) des Algorithmus liegt in $O(n^3)$, weil für die drei Variablen k, i, j die Werte von 1 bis n durchlaufen werden müssen.

Freiwilliges Übungsblatt 04

Bearbeiten Sie diese Woche bitte das vierte freiwillige Übungsblatt, das auf Canvas eingestellt wurde. Bleiben Sie am Ball!

↓ **Ende der 7. Vorlesungswoche**

Besprechung Blatt 04

Für das Freiwillige Blatt 04 erhalten Sie wieder eine handschriftliche Musterlösung (unter den Modulen in Canvas).

5 Induktion

5.1 Die Peano-Axiome

In Beispiel 3.5.5 und in Aufgabe 3 des dritten freiwilligen Übungsblattes haben wir gesehen, wie man mittels der Quotientenmenge einer clever gewählten Äquivalenzrelationen die Menge der rationalen Zahlen $\mathbb{Q}$ aus der Menge der ganzen Zahlen $\mathbb{Z}$ herleiten kann. Mit einem ähnlichen Konstrukt kann man auch $\mathbb{Z}$ aus $\mathbb{N}$ definieren (dies wurde in der Bemerkung in der Lösung des dritten freiwilligen Übungsblattes besprochen).

Zusammenfassend haben wir also $\mathbb{N}$ benutzt, um $\mathbb{Z}$ zu definieren, und $\mathbb{Z}$, um $\mathbb{Q}$ zu definieren. Dabei haben wir uns bisher immer auf den Standpunkt zurückgezogen, dass die natürlichen Zahlen aus der Schule bekannt sind. Schließlich weiß jedes Kind, wie man zählt.

Das ist natürlich eine sehr unübliche mathematische Vorgehensweise. Daher wollen wir an dieser Stelle nachreichen, wie man $\mathbb{N}$ exakt definieren kann. Wir werden in diesem Kapitel dann auch sehen, dass sich dies enorm lohnt. Aus der formalen Definition erhalten wir ein äußert mächtiges Beweiswerkzeug: das der vollständigen Induktion. Dieses brauchen wir oft, wenn wir eine Aussage für *alle* natürliche Zahlen zeigen wollen.

Definition 5.1.1. Die *Menge* $\mathbb{N}$ wird durch folgendes Axiomensystem (nach Peano 1889) charakterisiert:

(P1) $0 \in \mathbb{N}$
(P2) Jedes $n \in \mathbb{N}$ hat einen Nachfolger $n' \in \mathbb{N}$ und nur einen.
(P3) 0 ist kein Nachfolger.
(P4) Natürliche Zahlen mit gleichem Nachfolger sind gleich: $n, m \in \mathbb{N}$ und $n' = m' \Longrightarrow n = m$.
(P5) *Induktionsaxiom*: Ist $M \subseteq \mathbb{N}$ eine Teilmenge der natürlichen Zahlen mit den Eigenschaften
  (I1) $0 \in M$,
  (I2) $n \in M \Longrightarrow n + 1 := n' \in M$,
  so gilt $M = \mathbb{N}$.

Grundlagen Mathematik | 05.01: Peano-Axiome zum Einführen der Natürlichen Zahlen

5.2 Das Beweisprinzip der Induktion

Grundlagen Mathematik | 05.02: Das Beweisprinzip der vollständigen Induktion und der Kleine Gauß

Axiom (P5) beschreibt das „Dominoprinzip“ und bildet die Grundlage für das Prinzip der vollständigen Induktion:

Wenn der n-te Dominostein in der Reihe fällt, so auch der $n + 1$-te. Das gewährleistet den Induktionsschritt. Jetzt fällt der erste Dominostein. Folgerung: Schließlich werden alle Steine umgefallen sein.

Theorem 5.2.1 (Prinzip der vollständigen Induktion)**.** *Für jedes $n \in \mathbb{N}$ sei eine Aussage $A(n)$ gegeben. Ferner gelte:*

- Induktionsanfang (IA)*: $A(0)$ ist wahr.*
- Induktionsschritt (IS)*: Wenn $A(n)$ für ein $n \in \mathbb{N}$ wahr ist, dann ist auch $A(n+1)$ wahr.*

Dann ist $A(n)$ für alle $n \in \mathbb{N}$ wahr.

Beweis. Es sei

$$M := \{n \in \mathbb{N} \mid A(n) \text{ ist wahr}\}.$$

Dann ist natürlich $M \subseteq \mathbb{N}$. Nach (IA) ist $0 \in M$. Wegen (IS) gilt: $n \in M \Longrightarrow n+1 \in M$. Also gelten (I1) und (I2) in (P5). Dieses liefert $M = \mathbb{N}$. Daraus folgt die Behauptung. □

Bemerkung 5.2.2. *Das Induktionsprinzip ist auch für die Menge $\mathbb{N}^+$ gültig.*

Eine häufige Anwendung des Prinzips findet sich im Beweisen von Summen- und Produktformeln.

Beispiel 5.2.3. Wir wollen den *Kleinen Gauß* zeigen: Für alle $n \in \mathbb{N}$ gilt

$$\sum_{k=1}^{n} k = 1 + 2 + \cdots + n = \frac{n(n+1)}{2}.$$

Wir zeigen dies per Induktion nach n. Dazu bezeichne $A(n)$ eben obige Gleichheit.

IA: $n = 0$: Es gilt $\sum_{k=1}^{0} k = 0$ (leere Summe) und $\frac{0\cdot(0+1)}{2} = 0$. Also ist $A(0)$ wahr.

Induktionshypothese: Die Aussage $A(n)$ sei für *ein* $n \in \mathbb{N}$ wahr.

Induktionsschritt: Es gilt

$$\begin{aligned}\sum_{k=1}^{n+1} k &= \left(\sum_{k=1}^{n} k\right) + (k+1) \\ &\overset{\text{IH}}{=} \frac{n(n+1)}{2} + (n+1) \\ &= \frac{1}{2}(n^2 + n + 2n + 2) \\ &= \frac{1}{2}(n+1)(n+2).\end{aligned}$$

Also gilt $A(n+1)$.

Theorem 5.2.1 liefert die Behauptung.

Grundlagen Mathematik | 05.03: Induktiver Beweis der Geometrischen Summe und Sparplan Berechnungen

Beispiel 5.2.4. Es sei $q \in \mathbb{R} \setminus \{1\}$. Dann gilt für alle $n \in \mathbb{N}$:

$$\sum_{k=0}^{n} q^k = \frac{q^{n+1} - 1}{q - 1}.$$

Diese Aussage haben Sie bereits in Aufgabe 6 auf Tutorienblatt 03

gezeigt. Wir zeigen sie hier nocheinmal mit Induktion:

IA: $n = 0$: Es gilt $\sum_{k=0}^{0} q^k = q^0 = 1$. Andererseits ist $\frac{q^{0+1}-1}{q-1} = \frac{q-1}{q-1} = 1$. Also ist $A(0)$ wahr.

IH: Es gelte $A(n)$ für *ein* $n \in \mathbb{N}$.

IS: Es ist

$$\begin{aligned}\sum_{k=0}^{n+1} q^k &= \left(\sum_{k=0}^{n} q^k\right) + q^{n+1} \\ &\overset{\text{IH}}{=} \frac{q^{n+1}-1}{q-1} + q^{n+1} \\ &= \frac{q^{n+1}-1+q^{n+1}(q-1)}{q-1} \\ &= \frac{q^{(n+1)+1}-1}{q-1}.\end{aligned}$$

Also gilt $A(n+1)$. Damit folgt die Behauptung mit Theorem 5.2.1.

Die geometrische Summenformel lässt sich dazu verwenden, wie viel Geld man bis zu Rente sparen kann. Angenommen, man würde per Aktien-Sparplan jedes Jahr 2000 Euro anlegen. Wir gehen davon aus, dass diese mit 5 % per annum „verzinst" werden. Wie viel hätte man nach 10 Jahren gespart? Die ersten 2000 Euro, die man einzahlt, werden 10-mal verzinst, die zweiten werden 9-mal verzinst, die dritten werden 8-mal verzinst und so weiter. Damit ergibt sich der Betrag des Ersparten E zu

Der Autor des vorliegenden Skriptes ist sich darüber bewusst, dass Aktien keine zinstragenden Wertpapiere sind (trotz unsinniger Sprichwörter wie „Dividenden sind die neuen Zinsen"). Der Einfachheit halber gehen wir aber von solchen festen Renditen aus.

$$\begin{aligned}E &= 2000 \cdot 1{,}05^{10} + 2000 \cdot 1{,}05^{9} + \cdots + 2000 \cdot 1{,}05^{1} \\ &= \sum_{k=1}^{10} 2000 \cdot 1{,}05^k \\ &= 2000 \cdot 1{,}05 \cdot \sum_{k=0}^{9} 1{,}05^k \\ &= 2000 \cdot 1{,}05 \cdot \frac{1{,}05^{9+1}-1}{1{,}05-1} \\ &= 26\,413{,}57 \text{ Euro}.\end{aligned}$$

Beispiel 5.2.5. Die *Türme von Hanoi* sind ein mathematisches Knobelspiel. Es besteht aus drei Stäben und mehreren gelochten Scheiben, welche alle verschieden groß sind und auf die Stäbe gesteckt werden können. Die drei Stäbe werden im Weiteren mit A, B und C bezeichnet. Ziel des Spiels ist es, alle Scheiben vom Stab A (die dort der Größe nach geordnet mit der größten Scheibe unten) unter Zuhilfenahme von Stab B auf Stab C zu versetzen. In jedem Zug darf jedoch nur die oberste Scheibe eines Stabes auf einen anderen gelegt werden, aber nur wenn dort keine kleinere Scheibe liegt. D. h. die „Hochzeitstortenform" muss auf jedem Stab erhalten bleiben – die Scheiben sind stets der Größe nach geordnet. Animationen dazu finden Sie unter `https://de.wikipedia.org/wiki/Türme_von_Hanoi`,

Wir wollen diese Aufgabe mit möglichst wenig Zügen lösen. Ist $n \in \mathbb{N}^+$

Grundlagen Mathematik | 05.04: Die Türme von Hanoi und induktiver Beweis der optimalen Zugfolge

die Anzahl der der Scheiben, die zu Beginn des Spiels auf Stab A platziert sind, so bezeichne $Z(n)$ die Anzahl der Züge in einer kürzesten Zugfolge zum Lösen des Problems (man kann sogar zeigen, dass es nur eine solche Zugfolge geben kann). Wir behaupten, dass für alle $n \in \mathbb{N}^+$ gilt:

$$Z(n) = 2^n - 1.$$

Dies zeigen wir mit vollständiger Induktion über die Anzahl n der Scheiben:

IA: $n = 1$: Es liegt nur eine Scheibe auf Stab A. Offensichtlich besteht die optimale Zugfolge darin, diese Scheibe von Stab A auf Stab C zu legen. Daher ist

$$Z(1) = 1 = 2^1 - 1,$$

womit der Induktionsanfang gesichert ist.

IH: Es gelte $Z(n) = 2^n - 1$ für *ein* $n \in \mathbb{N}^+$.

IS: Wir wissen nach Induktionsvoraussetzung bereits, dass man n Scheiben mit $2^n - 1$ Zügen von einem Stab auf einen anderen bewegen kann und dies auch nicht schneller möglich ist. Wir erhalten die optimale Zugfolge für $n + 1$ Scheiben wie folgt:

a) Die obersten n Scheiben werden von Stab A auf Stab B umgeschichtet,
b) von Stab A wird die unterste und größte Scheibe des Spiels auf den leeren Stab C verschoben,
c) die n Scheiben von Stab B werden auf Stab C umgeschichtet.

Folglich ist

$$Z(n+1) = 2Z(n) + 1 \overset{\text{IH}}{=} 2(2^n - 1) + 1 = 2^{n+1} - 1.$$

Bei optimaler Zugfolge werden also $2^n - 1$ Züge zur Lösung der Aufgabe benötigt, wobei n die Anzahl der Scheiben ist. Es liegt also ein *exponentielles Wachstum* der Komplexität des Problems vor (vgl. Abschnitt 4.6). Damit ist eine praktische Umsetzung der Lösung nur für kleine n möglich. Die Tabelle zeigt die Dauer unter der Annahme, dass eine Scheibe pro Sekunde verschoben wird. Vermutlich wurde das Spiel 1883 vom französischen Mathematiker Édouard Lucas erfunden. Er dachte sich dazu die Geschichte aus, dass indische Mönche im großen Tempel zu Benares, im Mittelpunkt der Welt, einen Turm aus 64 goldenen Scheiben versetzen müssten, und wenn ihnen das gelungen sei, wäre das Ende der Welt gekommen. Wir scheinen bis dahin wohl noch etwas Zeit für mehr Mathematik zu haben...

Anzahl Scheiben	Benötigte Zeit
5	31 Sekunden
10	17,1 Minuten
20	12 Tage
30	34 Jahre
40	348 Jahrhunderte
60	36,6 Milliarden Jahre
64	585 Milliarden Jahre

Bemerkung 5.2.6. *Man kann den IA nicht weglassen! Betrachten wir dazu das Beispiel*

$$A(n): \ n > n + 1.$$

Diese Aussage ist offensichlich für alle natürliche Zahlen n falsch. Dennoch könnte man aus der Gültigkeit von $A(n)$ für ein $n \in \mathbb{N}$ auch $n + 1 > n + 2$ schließen, also $A(n + 1)$.

Weitere Aussagen, die man zum Üben durch Induktion zeigen kann: Für alle $n \in \mathbb{N}^+$ gilt:

- Die Potenzmenge einer endlichen Menge mit n Elementen enthält 2^n Elemente.
- Jede Teilung der euklidischen Ebene mit n vielen Geraden kann mit zwei Farben eingefärbt werden kann, sodass je zwei Teile mit einer gemeinsamen Kante nie von der gleichen Farbe sind.
- Man kann n verschiedene Objekte auf $n!$ Arten anordnen.

5.3 Starke Induktion

Grundlagen Mathematik | 05.05: Starke Induktion und die Formel von Moivre-Binet für Fibonacci-Zahlen

Wir formulieren noch eine Variante der vollständigen Induktion.

Theorem 5.3.1 (Variante von Theorem 5.2.1; Starke Induktionsvoraussetzung)**.** *Es sei $n_0 \in \mathbb{N}$ und für jedes $n \in \mathbb{N}$ mit $n \geq n_0$ sei eine Aussage $A(n)$ gegeben. Ferner gelte:*

- Induktionsanfang (IA*)*: $A(n_0)$ ist wahr.*
- Induktionsschritt (IS*)*: Wenn $A(n_0) \wedge A(n_0 + 1) \wedge \cdots \wedge A(n)$ für ein $n \in \mathbb{N}$ mit $n \geq n_0$ wahr ist, dann ist auch $A(n + 1)$ wahr.*

Dann ist $A(n)$ für alle $n \in \mathbb{N}$ mit $n \geq n_0$ wahr.

Beweis. Wir zeigen die Aussage per Induktion. Setze dazu

$$B(k) := A(n_0) \wedge \cdots \wedge A(n_0 + k - 1)$$

für $k \in \mathbb{N}^+$. Wir zeigen, dass $B(k)$ für alle $k \in \mathbb{N}^+$ eine wahre Aussage ist.

IA: $k = 1$: $B(1) = A(n_0)$ ist wahr nach (IA*).
IH: Es gelte $B(k)$ für *ein* $k \in \mathbb{N}^+$.
IS: Es sei $n := n_0 + k - 1$. Dann ist also $A(n_0) \wedge \cdots \wedge A(n)$ wahr. Nach (IS*) ist dann $A(n + 1)$ wahr. Nun ist $B(k + 1) = B(k) \wedge A(n + 1)$ wahr, denn $B(k)$ ist wahr nach IV und $A(n + 1)$ ist ebenfalls wahr, wie wir gerade gesehen haben. Nach Theorem 5.2.1 gilt somit $B(k)$ für alle $k \in \mathbb{N}^+$.

Hieraus folgt die Behauptung. □

Beispiel 5.3.2. Wir erinnern uns an die Fibonacci-Folge $(f_n)_{n\in\mathbb{N}}$ aus Beispiel 4.3.1: Diese war durch das *rekursive* Bildungsgesetz

$$f_n := f_{n-1} + f_{n-2} \quad \text{für } n \geq 2$$

mit den Anfangswerten

$$f_0 = 0 \text{ und } f_1 = 1$$

gegeben.

Das *explizite* Bildungsgesetz für die Glieder der Fibonacci-Folge wurde unabhängig voneinander von den französischen Mathematikern Abraham de Moivre im Jahr 1718 und Jacques Philippe Marie Binet im Jahr 1843 entdeckt. Dazwischen war sie aber auch den Mathematikern Leonhard Euler und Daniel Bernoulli bekannt. Letzterer lieferte 1728 auch den vermutlich ersten Beweis. Die Fibonacci-Zahlen lassen sich direkt mittels

$$f_n = \frac{\Phi^n - \Psi^n}{\Phi - \Psi} \quad \text{für } n \in \mathbb{N}$$

berechnen, wobei Φ und Ψ die beiden Lösungen der *charakteristischen Gleichung*

$$x^2 - x - 1 = 0$$

sind. Mit

$$\Phi := \frac{1+\sqrt{5}}{2} \approx 1{,}618\ldots,$$
$$\Psi := 1 - \Phi = -\Phi^{-1} = \frac{1-\sqrt{5}}{2} \approx -0{,}618\ldots$$

ist explizit

$$f_n = \frac{\Phi^n - \Psi^n}{\sqrt{5}} = \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right).$$

Die Zahl Φ ist der *goldene Schnitt*.

Wir werden diese Formel nicht *herleiten* (dies könnten wir z. B. durch ein Eigenwertproblem, das man mittels Methoden aus der Linearen Algebra löst, was allerdings diese Vorlesung weit übersteigt). Wir begnügen uns damit die Formel zu *beweisen*. Einer der einfachsten Beweise hierfür gelingt induktiv:

IA*: Wegen $\frac{\Phi^0-\Psi^0}{\sqrt{5}} = 0 = f_0$ und $\frac{\Phi^1-\Psi^1}{\sqrt{5}} = 1 = f_1$ ist der Induktionsanfang erfüllt.

IH*: Angenommen, die Formel gelte für alle Werte von 0 bis n (starke Induktionsvoraussetzung).

IS*: Wir zeigen, dass sie dann notwendigerweise auch für $n+1$ gelten

muss:

$$\begin{aligned} f_{n+1} &= f_{n-1} + f_n \\ &\overset{\text{IH}^*}{=} \frac{\Phi^{n-1} - \Psi^{n-1} + \Phi^n - \Psi^n}{\sqrt{5}} \\ &= \frac{\Phi^{n-1}(1+\Phi) - \Psi^{n-1}(1+\Psi)}{\sqrt{5}} \\ &\overset{\star}{=} \frac{\Phi^{n-1}(\Phi^2) - \Psi^{n-1}(\Psi^2)}{\sqrt{5}} \\ &= \frac{\Phi^{n+1} - \Psi^{n+1}}{\sqrt{5}}. \end{aligned}$$

Dabei haben wir im Schritt ($\star$) benutzt, dass Φ und Ψ der charakteristischen Gleichung $x^2 = x + 1$ genügen; es ist nämlich z. B.

$$\begin{aligned} \Phi + 1 = \frac{1+\sqrt{5}}{2} + 1 = \frac{1+\sqrt{5}+2}{2} = \frac{2+2\sqrt{5}+4}{4} \\ = \frac{1+2\sqrt{5}+5}{4} = \left(\frac{1+\sqrt{5}}{2}\right)^2 = \Phi^2. \end{aligned}$$

Es folgt (wie im Feedback gewünscht) noch ein etwas komplizierteres Beispiel. Es ist völlig in Ordnung, wenn man dieses beim ersten Lesen überspringt.

Grundlagen Mathematik | 05.06: Existenz im Zeckendorf-Theorem induktiv bewiesen

Beispiel 5.3.3. Das Zeckendorf-Theorem (benannt nach dem belgischen Amateur-Mathematik Edouard Zeckendorf) besagt, dass jede positive natürliche Zahl $N \in \mathbb{N}^+$ eindeutig als Summe (einer oder mehrerer) voneinander verschiedener, nicht direkt aufeinanderfolgender Fibonacci-Zahlen f_i mit Indizes $i \geq 2$ geschrieben werden kann. D. h. für jedes $N \in \mathbb{N}$ gibt es eine eindeutige Darstellung der Form

$$N = \sum_{i=2}^{k} c_i f_i$$

mit $c_i \in \{0, 1\}$ und $c_i c_{i+1} = 0$ für alle i. Diese Summe wird *Zeckendorf Repräsentation* von N genannt. Zum Beispiel ist

$$64 = 55 + 8 + 1.$$

Es gibt noch andere Arten 64 als Summe von Fibonacci-Zahlen darzustellen, z. B.

$$\begin{aligned} 64 &= 34 + 21 + 8 + 1, \\ 64 &= 55 + 5 + 3 + 1, \end{aligned}$$

doch diese sind keine Zeckendorf Repräsentationen, denn 34 und 21 sind aufeinanderfolgende Fibonacci-Zahlen, ebenso wie 5 und 3.

In diesem Beispiel wollen wir nur die Existenz (nicht jedoch die Eindeutigkeit) einer Zeckendorf Repräsentation für jede positive natürliche Zahl N nachweisen. Dies geschieht über Induktion. Für $N = 1, 2, 3$ ist die Aussage trivialerweise richtig (denn diese Zahlen sind selbst

Fibonacci-Zahlen). Für $N = 4$ hat man $4 = 3 + 1$ als Zeckendorf Repräsentation.

Ist N eine Fibonacci-Zahl, so sind wir fertig. Andernfalls gibt es einen Index j mit $f_j < N < f_{j+1}$. Es sei angenommen, dass jedes Zahl $a < N$ eine Zeckendorf Repräsentation besitzt (Induktionshypothese). Wir betrachten die Zahl $a := N - f_j$. Da $a < N$ ist, hat a nach Induktionshypothese eine Zeckendorf Repräsentation. Gleichzeitig ist $a < f_{j+1} - f_j = f_{j-1}$, d. h. die Zeckendorf Repräsentation von a kann f_{j-1} nicht enthalten. Daher kann N dargestellt werden als die Summe von f_j und der Zeckendorf Repräsentation von a (die f_{j-1} nicht enthält). Diese Darstellung ist eine Zeckendorf Repräsentation von N.

5.4 Strukturelle Induktion

Der Abschnitt über strukturelle Induktion ist nicht klausurrelevant (weshalb es auch keine Videolektion dazu gibt). Aber er wird Ihnen sicher in einigen weiteren Vorlesungen Ihres weiteren Studiums weiterhelfen.

Der Beginn des Abschnitt ist dem Wikipedia-Artikel https://de.wikipedia.org/wiki/Strukturelle_Induktion entlehnt.

Die strukturelle Induktion ist ein Beweisverfahren, das unter anderem in der Logik, der theoretischen Informatik und der Graphentheorie eingesetzt wird. Es handelt sich um eine allgemeinere Form der vollständigen Induktion. Mit dem Verfahren lassen sich Aussagen über die Elemente von rekursiv aufgebauten Mengen (zum Beispiel Mengen von Listen, Formeln, Graphen) beweisen.

Bei der vollständigen Induktion werden mit natürlichen Zahlen numerierte Aussagen bewiesen; bei der strukturellen Induktion werden Eigenschaften für Mengen bewiesen, deren Elemente aus Grundelementen durch eine endliche Anzahl von Konstruktionsschritten (unter Verwendung bereits konstruierter Elemente) bzw. mittels eines Erzeugungssystems entstehen. Es gibt also minimale (auch: einfachste oder Grund-)Elemente und rekursiv definierte (oder: rekursiv gebildete) Elemente der Menge. Bei den natürlichen Zahlen ist das Grundelement 0 und der Konstruktionsschritt ist der Übergang von einer Zahl n zum Nachfolger n'.

Um eine Aussage für die Elemente einer Menge zu beweisen, zeigt man im Induktionsanfang die Gültigkeit der Aussage für die einfachsten Elemente und im Induktionsschluss die Gültigkeit der Aussage für die rekursiv gebildeten Elemente unter der Voraussetzung, dass die Aussage für die in der Konstruktion verwendeten Elemente gilt. Ist beides erfüllt, so gilt die Aussage für alle Elemente. Man führt die Induktion also über den strukturellen Aufbau der Elemente.

Beispiel 5.4.1 (Beispiel für eine Definition durch strukturelle Induktion)**.** Die Menge der aussagenlogischen Formeln lässt sich mittels struktureller Induktion wie folgt definieren:

- IA: Falls A eine atomare aussagenlogische Formel ist (also eine Variable), ist A eine aussagenlogische Formel.
- IS1: Falls F eine aussagenlogische Formel ist, ist auch $\neg F$ eine aussagenlogische Formel.
- IS2: Falls F und G aussagenlogische Formeln sind, ist auch $(F \wedge G)$ eine aussagenlogische Formel.
- IS3: Falls F und G aussagenlogische Formeln sind, ist auch $(F \vee G)$ eine aussagenlogische Formel.

IS4: Falls F und G aussagenlogische Formeln sind, ist auch $(F \to G)$ eine aussagenlogische Formel.
IS5: Falls F und G aussagenlogische Formeln sind, ist auch $(F \leftrightarrow G)$ eine aussagenlogische Formel.

Nach dieser Definition sind z. B. die folgenden Terme aussagenlogische Formeln:

- $\neg(A \wedge B)$,
- $\big(A \to (B \vee \neg C)\big)$,
- $\big((A \vee \neg B) \wedge (\neg A \vee B)\big)$.

Viele Klammern können weggelassen werden, wenn man die Assoziativität von $\wedge$ und $\vee$ ausnutzt und eine Operatorrangfolge vereinbart.

Beispiel 5.4.2 (Beispiel für einen Beweis durch strukturelle Induktion)**.** Bewiesen wird der Satz: Für jede aussagenlogische Formel F gibt es eine äquivalente aussagenlogische Formel $[F]$, in der als einzige Operatoren $\neg$ und $\wedge$ vorkommen.

Der Beweis:

IA: Falls $F = A$ für eine atomare aussagenlogische Formel A ist, so ist $[F] = A$.
IS1: Falls $F = \neg G$ für eine aussagenlogische Formel G gilt, ist $[F] = \neg[G]$.
IS2: Falls $F = (G \wedge H)$ für aussagenlogische Formeln G und H gilt, ist $[F] = ([G] \wedge [H])$.
IS3: Falls $F = (G \vee H)$ für aussagenlogische Formeln G und H gilt, ist $[F] = \neg(\neg[G] \wedge \neg[H])$.
IS4: Falls $F = (G \to H)$ für aussagenlogische Formeln G und H gilt, ist $[F] = \neg([G] \wedge \neg[H])$.
IS5: Falls $F = (G \leftrightarrow H)$ für aussagenlogische Formeln G und H gilt, ist $[F] = \neg(\neg([G] \wedge [H]) \wedge \neg(\neg[G] \wedge \neg[H]))$.

Das Gleichheitszeichen steht hier für *syntaktische Gleichheit*, d. h. Gleichheit Zeichen für Zeichen. In jedem Induktionsschritt wird vorausgesetzt, dass für G und H jeweils die äquivalenten Formeln $[G]$ und $[H]$ existieren, die nur $\neg$ und $\wedge$ verwenden (Induktionsvoraussetzung).

In einer konkreten Konstruktion kann man die Beweisschritte auch in der umgekehrten Reihenfolge, also „von außen nach innen“, anwenden. Für die mittlere der oben angegebenen aussagenlogischen Formeln gelten z. B. die folgenden Äquivalenzen:

$$\begin{aligned}
\big(A \to (B \vee \neg C)\big) &\equiv \Big[\big(A \to (B \vee \neg C)\big)\Big] \\
&\overset{\text{IS4}}{=} \neg\Big([A] \wedge \neg\big[(B \vee \neg C)\big]\Big) \\
&\overset{\text{IS3}}{=} \neg\big([A] \wedge \neg\neg(\neg[B] \wedge \neg[\neg C])\big) \\
&\overset{\text{IS1}}{=} \neg\big([A] \wedge \neg\neg(\neg[B] \wedge \neg\neg[C])\big) \\
&\overset{3\times\text{IA}}{=} \neg\big(A \wedge \neg\neg(\neg B \wedge \neg\neg C)\big)
\end{aligned}$$

Dass sich die letzte Formel noch zu $\neg(A \wedge (\neg B \wedge C))$ vereinfachen lässt, ist hier übrigens unerheblich.

Boolesche Schaltkreise sind ebenfalls nicht klausurrelevant.

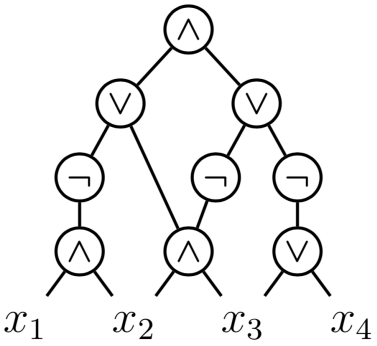
Definition 5.4.3 (Boolesche Schaltkreise). 1. Ein *Boolescher Schaltkreis* über den Variablen $x_1, \ldots, x_n$ ist ein Tupel $s = (g_1, \ldots, g_m)$ von *Gattern*

$$g_\ell \in \{0, 1, x_1, \ldots, x_n\} \cup \bigcup_{1 \le j < \ell} \{(\neg, j)\} \cup \bigcup_{1 \le j,k < \ell} \{(\wedge, j, k), (\vee, j, k)\}.$$

2. Der *Funktionswert der am Gatter g_ℓ berechneten n-stellige Funktion* ist für $a \in \{0,1\}^n$ induktiv wie folgt definiert:

g_ℓ	0	1	x_i	$(\neg, j)$	$(\wedge, j, k)$	$(\vee, j, k)$
$g_\ell(a)$	0	1	a_i	$\neg g_j(a)$	$g_j(a) \wedge g_k(a)$	$g_j(a) \vee g_k(a)$

3. Der *Wert des Schaltkreises* bei *Eingabe* $a \in \{0,1\}^n$ definiert als $s(a) := g_m(a)$.
4. Der Schaltkreis s heißt *erfüllbar*, falls eine *Eingabe* $a \in \{0,1\}^n$ existiert mit $s(a) = 1$.



Beispiel 5.4.4. Der Schaltkreis

$$\begin{aligned} s = \big(&x_1,\ x_2,\ x_3,\ x_4,\\ &(\wedge,1,2),\ (\wedge,2,3),\ (\vee,3,4),\\ &(\neg,5),\ (\neg,6),\ (\neg,7),\\ &(\vee,6,8),\ (\vee,9,10),\\ &(\wedge,11,12)\big) \end{aligned}$$

ist nebenstehend graphisch dargestellt.

Die Anzahl der Eingänge eines Gatters g wird als *Fanin* von g bezeichnet, die Anzahl der Ausgänge von g (d. h. die Anzahl der Gatter, die g als Eingabe benutzen) als *Fanout*.

↓ **Ende der 8. Vorlesungswoche**

5.5 Planare Graphen

Der Abschnitt über planare Graphen ist zum Selbststudium gedacht.

Ein Graph heißt *planar*, wenn er sich in der Ebene so zeichnen lässt, dass sich keine zwei Kanten überkreuzen (Kanten dürfen hierbei, wenn nötig, als krumme Linien gezeichnet werden).

Wird ein planarer Graph in der Ebene kreuzungsfrei gezeichnet (was auf verschiedene Arten möglich sein kann), so wird die Zeichnungsebene hierbei in verschiedene *Regionen* zerlegt, wobei wir auch die „äußere" unendlich große Region mitzählen. Der folgende Satz von Euler sagt, dass es für die Anzahl der Regionen keine Rolle spielt, wie der Graph gezeichnet wird, und ferner stellt er einen Zusammenhang zwischen Regionenzahl, Knotenzahl und Kantenzahl her.

Satz 5.5.1 (Spezialfall des Eulerschen Polyedersatzes). *In jedem zusammenhängenden, planaren Graphen gilt:*

$$\textit{Knotenzahl - Kantenzahl + Regionenzahl} = 2.$$

Allgemeiner gilt der Satz für dreidimensionale geometrische Körper, die aus identischen, regelmäßigen Flächenstücken zusammengesetzt sind. Das wohl bekannteste Beispiel eines solchen *Platonischen Körpers* ist der Würfel, dieser besteht aus sechs Quadraten. Der Eulersche Polyedersatz besagt, dass für jeden konvexen Polyeder (= Vielflächer ohne einspringende Ecken)

$$E - K + F = 2$$

gilt, wobei E die Anzahl der Ecken, F die Anzahl der Flächen und K die Anzahl der Kanten bezeichnet. Diesen Satz könnte man wie folgt beweisen: Durch Verwendung einer *Zentralprojektion* des gegebenen Polyeders können wir diesen als planaren Graphen darstellen. Anschaulich wird dabei eine Fläche entfernt und die Kanten nach außen gezogen, und das Objekt glattgebügelt, wodurch wir den Körper auf die Ebene projizieren. D. h. es genügt obigen Spezialfall für planare Graphen zu zeigen.

Beweis von Satz 5.5.1. Im Folgenden sei immer n die Knotenzahl, m die Kantenzahl, und r die Regionenzahl.

Es könne hilfreich sein auf einem Blatt ein paar Skizzen beim Lesen des Beweises anzufertigen.

Wir stellen uns vor, das kreuzungsfreie Zeichnen eines zusammenhängenden, planaren Graphen geschehe schrittweise. Als mögliche Schritte kommen in Frage:

1. (Als erster Schritt:) Das Platzieren eines Knotens.
2. Die Unterteilung einer Kante durch einen neuen Knoten.
3. Die Verbindung zweier schon vorhandener Knoten durch eine Kante, die hierbei keine andere Kante schneidet.
4. Das Ansetzen einer neuen Kante mit einem neuen Endknoten an einen schon vorhandenen Knoten.

Nach dem ersten Schritt haben wir $r = n = 1$ und $m = 0$, also $n - m + r = 2$. Bei Schritt 2 erhöhen sich n und m um 1. Ebenso bei Schritt 4. Der Wert von $n - m + r$ bleibt dabei konstant bei 2. Bei Schritt 3 erhöhen sich m und r um 1, also bleibt auch hier der Wert von $n - m + r$ konstant. Strukturelle Induktion liefert die Behauptung □

5.6 Schleifeninvarianten

Mit vollständiger Induktion lässt sich auch die Korrektheit von Algorithmen nachweisen. Will man die Werte verfolgen, die die Variablen beim Ablauf eines Algorithmus annehmen, dann stellen Schleifen die eigentlich Herausforderung dar. Um dies in den Griff zu bekommen stellt man *Schleifeninvarianten* auf. Dies sind Eigenschaften einer Schleife, die bei jedem Schleifendurchlauf erhalten bleiben. Wir betrachten drei Beispiele, die Umrechnung einer Dezimalzahl in eine Binärzahl, die Exponentiation einer Zahl und die binäre Suche.

Dezimal zu Binär

Grundlagen Mathematik | 05.07: Schleifeninvarianten – Dezimal zu Binär

Die Darstellung einer natürlichen Zahl n im Zahlensystem zur Basis $b \geq 2$ hat die Form

$$n = n_{\ell-1} b^{\ell-1} + n_{\ell-2} b^{\ell-2} + \cdots + n_2 b^2 + n_1 b + n_0, \tag{5.1}$$

wobei die *Ziffern* $n_i \in \{0, 1, \ldots, b-1\}$ sind. Wir schreiben dann kürzer

$$n = (n_{\ell-1}\, n_{\ell-2} \cdots n_1\, n_0)_b.$$

Im Dezimalsystem, also für $b = 10$, lässt man üblicherweise die Basis weg und schreibt $n = n_{\ell-1}\, n_{\ell-2} \cdots n_1\, n_0$. Ist die *höchstwertige* Ziffer $n_{\ell-1} \neq 0$, dann heißt die Anzahl ℓ der Stellen von n auch die *Länge von n* (*zur Basis b*).

Wir wollen die Dezimaldarstellung einer Zahl in ihre Binärdarstellung umrechnen. D. h. auf die Eingabe einer Zahl $n = n_{\ell-1}\, n_{\ell-2} \cdots n_1\, n_0$ in Dezimaldarstellung soll die Binärdarstellung $n = (b_{k-1}\, b_{k-2} \cdots b_1\, b_0)_2$ von n berechnet werden.

Eine Möglichkeit dafür wäre, nach der größten Zweierpotenz zu suchen, die $\leq n$ ist, und diese dann von n abzuziehen. Dies wiederholt man dann mit der Differenz. Zum Beispiel für $n = 10$ ist $2^3 = 8$ die größte Zweierpotenz ≤ 10. Die Differenz ist $10 - 8 = 2$. Die nächst kleinere Zweierpotenz ist $2^2 = 4$ und ist > 2. Dann kommt $2^1 = 2$. Dies ist ≤ 2. Die Differenz ist 0. Damit sind wir fertig. Die Binärdarstellung von 10 ist folglich $(1010)_2$.

Es gibt eine elegantere Methode. Dazu betrachten wir die Binärdarstellung von n wie in (5.1). Alle Summanden sind Vielfache von 2, mit Ausnahme von b_0. Wir können also schreiben

$$\begin{aligned} n &= b_{k-1}2^{k-1} + b_{k-2}2^{k-2} + \cdots + b_1 2 + b_0 \\ &= (b_{k-1}2^{k-2} + b_{k-2}2^{k-3} + \cdots + b_1) \cdot 2 + b_0. \end{aligned}$$

D. h. wenn wir n durch 2 ganzzahlig dividieren, so bleibt b_0 als Rest übrig. Ist n gerade, dann ist $b_0 = 0$, ist n ungerade, dann ist $b_0 = 1$. Der ganzzahlige Quotient ist $(b_{k-1}\, b_{k-2} \cdots b_1)_2$. Mit dem können wir analog verfahren und so der Reihe nach $b_1, b_2, \ldots$ bestimmen. Für $n = 10$ erhalten wir:

$$\begin{aligned} 10 \div 2 &= 5 \text{ Rest } 0 \\ 5 \div 2 &= 2 \text{ Rest } 1 \\ 2 \div 2 &= 1 \text{ Rest } 0 \\ 1 \div 2 &= 0 \text{ Rest } 1 \end{aligned}$$

Die Binärdarstellung von 10 ist also $(1010)_2$.

Ausgeschrieben betrachten wir somit folgende Darstellung von n:

$$n = \big(\cdots(b_{k-1} \cdot 2 + b_{k-2}) \cdot 2 + \cdots + b_1\big) \cdot 2 + b_0. \tag{5.2}$$

Zum Beispiel für $n = 10$ bekommen wir

$$\begin{aligned} 10 &= 2^3 + 2 \\ &= (2^2 + 1) \cdot 2 \\ &= \big((1 \cdot 2 + 0) \cdot 2 + 1\big) \cdot 2 + 0. \end{aligned}$$

Die Schreibweise in (5.2) entpricht der Darstellung eines Polynoms gemäß dem *Hornerschema*. Für die Binärdarstellung betrachten wir das Polynom p mit

$$p(x) = b_{k-1}x^{k-1} + b_{k-2}x^{k-2} + b_1 x + b_0,$$

wobei die Koeffizienten b_i die Ziffern der Binärdarstellung sind, d.h. $b_i \in \{0, 1\}$. Das Hornerschema klammert gleiche Potenzen so weit wie möglich aus:

$$p(x) = \big(\cdots(b_{k-1}\,x + b_{k-2})\,x + \cdots + b_1\big)\,x + b_0$$

Für $x = 2$ erhalten wir die Darstellung in (5.2).

Der folgende Algorithmus Dec2Bin führt diese Methode auf einer Eingabe $n > 0$ aus und schreibt die Reste in ein Feld b. Am Ende steht in b die Binärdarstellung von n.

```
Dec2Bin(n)  (* n > 0 *)
1  t ← n
2  c ← 0
3  k ← 0
4  while t > 0 do
5      b[k] ← t mod 2
6      c ← c + b[k] · 2^k
7      k ← k + 1
8      t ← ⌊t/2⌋
9  return b
```

In Zeile 1 wird die Eingabe n in der Variablen t gespeichert. Mit t werden dann die Divisionen durchgeführt. Dadurch bleibt n unverändert. Die Variable c wird eigentlich nicht benötigt, sie dient nur zur Erklärung. Man kann die Zeilen 2 und 6 also unbeschadet aus dem Programm streichen. Variable k indiziert die Bits b_k der Binärdarstellung.

Mit obiger Herleitung wissen wir bereits, dass Dec2Bin korrekt arbeitet. Wir zeigen dies im Folgenden nocheinmal mit Hilfe von Schleifeninvarianten. In diesem Fall sind das die folgenden beiden Gleichungen. Wir zeigen, dass sie jedesmal gelten wenn Zeile 4 erreicht wird.

(i) $c = (b_{k-1} \cdots b_0)_2$

(ii) $n = t \cdot 2^k + c$

Wir zeigen die Schleifeninvarianten mit vollständiger Induktion darüber, wie oft Zeile 4 erreicht wurde. Beim Induktionsanfang betrachten wir den Fall, dass wir zum ersten Mal Zeile 4 erreichen, d. h. der Algorithmus hat die Zeilen 1–3 durchlaufen. Für die Variablen gilt dann $t = n$, $c = 0$ und $k = 0$.

Wir verifizieren die Invarianten:

- Gleichung (i) gilt, da $c = 0$ und ebenfalls

$$(b_{k-1} \cdots b_0)_2 = (b_{-1} \cdots b_0)_2 = 0.$$

- Gleichung (ii) gilt, da

$$t \cdot 2^k + c = n \cdot 2^0 + 0 = n.$$

Im Induktionsschritt zeigen wir, dass wenn die Invarianten bei einem Erreichen von Zeile 4 gelten, dann gelten sie auch beim nächsten Mal. Damit wir die Werte einer Variablen vorher und nachher auseinander

halten können geben wir ihnen im Beweis verschiedene Namen. Seien t, c und k die aktuellen Werte in Zeile 4. Für diese gelten nach Induktionsvoraussetzung die Invarianten. Die neuen Werte der Variablen, wenn wir das nächste mal Zeile 4 erreichen, bezeichnen wir mit t', c' und k'. Die Induktionsbehauptung ist, dass auch für die neuen Werte die Invarianten gelten.

Wir betrachten also einen Schleifendurchlauf. Der Wert von b_k in Zeile 5 hängt davon ab, ob t gerade oder ungerade ist. Wir unterscheiden die beiden Fälle.

Fall 1. t ist gerade. Wenn wir die Zeilen 5 – 8 ausführen, erhalten wir

$$\begin{aligned} b_k &= 0, \\ c' &= c + 0 \cdot 2^k = c, \\ k' &= k + 1, \\ t' &= t/2. \end{aligned}$$

Wir verifizieren die Invarianten für die neuen Werte. Gleichung (i):

$$\begin{aligned} (b_{k'-1} \cdots b_0)_2 &= (b_k \cdots b_0)_2 \\ &= (0\, b_{k-1} \cdots b_0)_2 \\ &= (b_{k-1} \cdots b_0)_2 \quad \text{führende Null kann man weglassen} \\ &= c \quad \text{nach Induktionsvoraussetzung} \\ &= c'. \end{aligned}$$

Gleichung (ii):

$$\begin{aligned} t' \cdot 2^{k'} + c' &= \frac{t}{2} \cdot 2^{k+1} + c \\ &= t \cdot 2^k + c \\ &= n \quad \text{nach Induktionsvoraussetzung.} \end{aligned}$$

Fall 2. t ist ungerade. Nach den Zeilen 5 – 8 gilt

$$\begin{aligned} b_k &= 1, \\ c' &= c + 2^k, \\ k' &= k + 1, \\ t' &= \frac{t-1}{2}. \end{aligned}$$

Wir verifizieren wieder die Invarianten für die neuen Werte. Gleichung (i):

$$\begin{aligned} (b_{k'-1} \cdots b_0)_2 &= (1\, b_{k-1} \cdots b_0)_2 \\ &= (b_{k-1} \cdots b_0)_2 + 2^k \quad \text{führende Eins hat Wert } 2^k \\ &= c + 2^k \quad \text{nach Induktionsvoraussetzung} \\ &= c'. \end{aligned}$$

Gleichung (ii):

$$\begin{aligned} t' \cdot 2^{k'} + c' &= \frac{t-1}{2} \cdot 2^{k+1} + c + 2^k \\ &= (t-1) \cdot 2^k + c + 2^k \\ &= t \cdot 2^k + c \\ &= n \quad \text{nach Induktionsvoraussetzung.} \end{aligned}$$

Damit haben wir die gezeigt, dass die Invarianten jedes mal gelten, wenn Zeile 4 erreicht wird. Insbesondere gilt dies auch beim letzten Mal. Dies ist, wenn $t = 0$ ist. Die Invarianten liefern dann $n = c = (b)_2$. Das zeigt, dass der Algorithmus Dec2Bin korrekt ist.

Wir analysieren abschließend die Laufzeit von Dec2Bin. In einem Schleifendurchlauf werden konstant viele Rechenoperationen ausgeführt. Die Rechenzeit ist also proportional zur Anzahl der Schleifendurchläufe. Da die Variable k mit jedem Schleifendurchlauf um Eins hochgezählt wird, gibt k am Ende die Anzahl der Durchläufe an. Gleichzeitig ist k auch die Länge der Binärdarstellung von n, also $k = \lfloor \log_2 n \rfloor + 1$ (dies haben Sie auf Tutoriumsblatt 03, Aufgabe 7 (b) gezeigt).

Die Laufzeit wird als Funktion in der *Länge der Eingabe* angegeben. Wenn wir n als Dezimalzahl eingeben, hat die Eingabe die Länge $\ell = \lfloor \log_{10} n \rfloor + 1$. Verschiedene Logarithmen unterscheiden sich nur durch einen konstanten Faktor. Es gibt also eine Konstante $c \geq 1$, so dass $k \leq c\ell$. D.h. Dec2Bin hat nach Abschnitt 4.6 lineare Laufzeit und ist damit sehr effizient.

Wir wollen nochmal betonen, dass bei Algorithmen deren Eingabe eine Zahl n ist, die Länge von n als Maß für die Rechenzeit genommen wird, also $\log n$, und nicht n. Betrachten wir beispielsweise einen Algorithmus der aus einer Schleife besteht, die von der Eingabe n (in Dezimaldarstellung) in jedem Durchlauf Eins abzieht. Dieser Algorithmus macht also n Schritte und hat damit *exponentielle* Laufzeit, da $n = 10^{\log n}$. Algorithmen mit exponentieller Laufzeit sind bereits für relativ kleine Werte von n nicht mehr praktikabel. Auch schnelle Rechner brauchen dann Jahre um solch einen Algorithmus auszuführen.

Exponentiation

Grundlagen Mathematik | 05.08: Schleifeninvarianten – Schnelle Exponentiation

Auf Eingabe von Zahlen $a \in \mathbb{R}$ und $n \in \mathbb{N}_{\geq 2}$ soll a^n berechnet werden. Die naive Vorgehensweise multipliziert den Ausdruck $a \cdot a \cdot a \cdot \ldots \cdot a$ von links nach rechts. Dabei werden $n - 1$ Multiplikationen durchgeführt. Wir wollen versuchen mit weniger Multiplikationen auszukommen.

Wir vereinbaren: Ist $k \in \mathbb{N}^+$, so ist $\mathbb{N}_{\geq k} := \mathbb{N} \setminus [k] \setminus \{0\}$.

Das ist möglich weil die Multiplikation assoziativ ist. Um zum Beispiel a^4 zu berechnen können wir zunächst a quadrieren. Das ist eine Multiplikation. Dann quadrieren wir das Ergebnis mit einer weiteren Multiplikation nochmal. Die Rechnung ist also

$$a^4 = (a^2)^2$$

und kostet nur zwei Multiplikation, statt drei beim naiven Verfahren. Wenn wir nochmal quadrieren verdoppeln wir den Exponenten:

$$a^8 = \big((a^2)^2\big)^2.$$

Dies kostet nur drei Multiplikation, während es beim naiven Verfahren bereits 7 sind. Wenn wir weiter quadrieren erreichen wir alle Zweierpotenzen als Exponent. Ist also $n = 2^k$, für ein $k \geq 1$, dann können wir a^n durch k-maliges quadrieren berechnen, statt $n - 1$ Multiplikationen beim naiven Verfahren. Es ist $k = \log n$, d. h. wir haben exponentiell viele Multiplikationen eingespart!

Was machen wir, wenn n keine Zweierpotenz ist? Dazu betrachten wir nochmal die Schreibweise der Binärdarstellung von n gemäß dem Hornerschema (5.2):

$$n = \big(\cdots(b_{k-1} \cdot 2 + b_{k-2}) \cdot 2 + \cdots + b_1\big) \cdot 2 + b_0.$$

Für $n = 10 = (2 \cdot 2 + 1) \cdot 2$ im Exponent erhalten wir damit

$$a^{10} = a^{(2\cdot 2+1)\cdot 2} = \Big(\big((a^2)^2\big) \cdot a\Big)^2.$$

Für jede Ziffer b_i der Binärdarstellung von n wird also das aktuelle Zwischenergebnis quadriert. Ist $b_i = 1$, dann multiplizieren wir zusätzlich noch mit a. Bei den Zweierpotenzen sind alle Ziffern Null (bis auf die führende Eins). Deswegen reichte es hier lediglich zu quadrieren.

Der folgende Algorithmus Exp berechnet a^n nach dieser Methode.

```
Exp(a, n)
 1  b ← Dec2Bin(n) = (b_{k-1}, ..., b_0)
 2  c ← 0
 3  d ← 1
 4  for i ← k − 1 downto 0 do
 5      c ← 2c
 6      d ← d · d
 7      if b_i = 1 then
 8          c ← c + 1
 9          d ← d · a
10  return d
```

Zuerst wird die Binärdarstellung von n mit Dec2Bin berechnet. Die Variable c wird eigentlich nicht benötigt, sie dient nur zur Erklärung. Die Zeilen 2 und 5 kann man also weglassen. Variable d enthält am Ende das Ergebnis und wird mit Eins initialisiert.

Wir zeigen die Korrektheit von Exp. Es gelten folgende Invarianten bei jedem Erreichen von Zeile 4:

(i) $c = (b_{k-1} \cdots b_{i+1})_2$

(ii) $d = a^c$

Wir zeigen die Schleifeninvarianten wieder mit vollständiger Induktion darüber, wie oft Zeile 4 erreicht wurde.

Wenn wir zum ersten Mal Zeile 4 erreichen, gilt dann $c = 0$, $d = 1$ und $i = k - 1$. Wir verifizieren die Invarianten:

- Gleichung (i) gilt, da $c = 0$ und ebenfalls

$$(b_{k-1} \cdots b_{i+1})_2 = (b_{k-1} \cdots b_k)_2 = 0.$$

- Gleichung (ii) gilt, da $a^c = a^0 = 1 = d$.

Im Induktionsschritt nehmen wir wieder an, dass die Invarianten für die aktuellen Werte von c, d und i in Zeile 4 gelten. Wir zeigen, dass sie auch bei nächsten Mal in Zeile 4 für die neuen Werte c', d' und i' gelten.

Nach Zeile 5 und 6 gilt $c' = 2c$ und $d' = d^2$. Im Fall, dass $b_i = 0$ ist die Schleife bereits zu Ende. Es wird lediglich noch i runtergezählt, $i' = i - 1$.

Fall 1. $b_i = 0$. Wir verifizieren die Invarianten. Gleichung (i):

$$\begin{aligned}(b_{k-1} \cdots b_{i'+1})_2 &= (b_{k-1} \cdots b_i)_2 \\ &= (b_{k-1} \cdots b_{i+1}0)_2 \\ &= 2 \cdot (b_{k-1} \cdots b_{i+1})_2 \\ &= 2c \quad \text{nach Induktionsvoraussetzung} \\ &= c'.\end{aligned}$$

Gleichung (ii):

$$a^{c'} = a^{2c} = (a^c)^2 = d^2 = d'.$$

Fall 2. $b_i = 1$. In diesem Fall werden c' und d' in Zeile 8 und 9 noch weiter verändert. Es gilt dann $c' = 2c + 1$ und $d' = d^2 \cdot a$.

Wir verifizieren wieder die Invarianten. Gleichung (i):

$$\begin{aligned}(b_{k-1} \cdots b_{i'+1})_2 &= (b_{k-1} \cdots b_{i+1}1)_2 \\ &= 2 \cdot (b_{k-1} \cdots b_{i+1})_2 + 1 \\ &= 2c + 1 \quad \text{nach Induktionsvoraussetzung} \\ &= c'.\end{aligned}$$

Gleichung (ii):

$$a^{c'} = a^{2c+1} = (a^c)^2 \cdot a = d^2 \cdot a = d'.$$

Damit haben wir die Invarianten bewiesen. Das letzte Mal wird Zeile 4 mit $i = -1$ erreicht. Gemäß Invariante (i) gilt dann $n = c = (b)_2$. Eingesetzt in Invariante (ii) erhalten wir damit $d = a^n$. Somit ist Algorithmus Exp korrekt.

Wir schätzen die Anzahl der Multiplikationen von Exp ab. In jedem Schleifendurchlauf wird in Zeile 6 eine Multiplikation ausgeführt. Dann, abhängig vom jeweiligen Bit b_i, nochmal eine in Zeile 9. Insgesamt also $\leq 2k$ Multiplikationen.

Die Laufzeit von Exp ist proportional zu k, die Länge der Binärdarstellung von n. Damit hat auch Exp lineare Laufzeit bezüglich der Länge der Eingabe.

Grundlagen Mathematik | 05.09: Schleifeninvarianten – Binäre Suche

Binäre Suche

Der Algorithmus Binary Search bekommt als Eingabe ein sortiertes Feld A mit $n \in \mathbb{N}^+$ Elementen und ein Element a. Es gilt also

$$A[1] \leq A[2] \leq \cdots \leq A[n].$$

Wir nehmen zunächst an, dass n eine Zweierpotenz ist, d.h. $n = 2^p$, für ein $p \geq 0$. Der Algorithmus soll herausfinden, ob a in A vorkommt. Dazu vergleicht er a mit dem mittleren Element von A und sucht je nach dem in der unteren oder der oberen Hälfte von A weiter.

Binary Search(A, a)

```
1  l ← 1
2  r ← n
3  k ← 0
4  while l < r do
5      k ← k + 1
6      m ← (l+r−1)/2
7      if A[m] < a then l ← m + 1
8      else r ← m
9  return A[l] = a
```

Korrektheit

Kommt a nicht in A vor, dann ist Binary Search korrekt, da dies am Ende durch den Test in Zeile 9 sicher gestellt wird. Nehmen wir im Folgenden also an, dass a in A vorkommt. Wir müssen zeigen, dass Binary Search dann a findet. Es gelten folgende Schleifeninvarianten in Zeile 4:

(i) $r - l + 1 = n/2^k = 2^{p-k}$
(ii) $A[l] \leq a \leq A[r]$

Wir beweisen die Schleifeninvarianten durch Induktion über k: Die Variable k wird ansonsten eigentlich nicht benötigt, sie ist nur zu Erklärungszwecken da.

Für $k = 0$ ist $r - l + 1 = n - 1 + 1 = n = n/2^0$ und $A[1] \leq a \leq A[n]$. Der Induktionsanfang gilt also. Sei nun $k \geq 0$ und gelten die Invarianten für die Werte r, l und k. Wir betrachten einen Schleifendurchlauf, Zeile 5 bis 8. Die neu berechneten Werte bezeichnen wir wieder mit r', l' und k'. Nach Zeile 5 und 6 gilt also $k' = k + 1$ und $m = \frac{l+r-1}{2}$. Der Vergleich in Zeile 7 ergibt zwei Fälle.

Fall 1. $A[m] < a$. Dann ist $l' = m + 1$ und $r' = r$ nach Zeile 7. Wir zeigen, dass die Invarianten für die neuen Werte gelten:

$$r' - l' + 1 = r - (m+1) + 1 = r - \frac{l+r-1}{2} = \frac{r-l+1}{2} = \frac{n}{2^{k+1}} = \frac{n}{2^{k'}}.$$

Da $r' = r$ ist, gilt $a \leq A[r]$. Da $A[m] < a$, ist auch $A[l'] = A[m+1] \leq a$.

Fall 2. $A[m] \geq a$. Dann ist $l' = l$ und $r' = m$ nach Zeile 8. Wir zeigen, dass die Invarianten wieder gelten:

$$r' - l' + 1 = m - l + 1 = \frac{l+r-1}{2} - l + 1 = \frac{r-l+1}{2} = \frac{n}{2^{k'}}.$$

Da $l' = l$ ist, gilt $A[l'] \leq a$. Da $a \leq A[m]$ und $r' = m$, ist auch $a \leq A[r']$.

Binary Search stoppt, wenn die Bedingung der while-Schleife nicht mehr erfüllt ist, wenn also $l = r$ gilt. Nach Bedingung (ii) gilt $A[l] \leq a \leq A[r]$. Folglich ist dann $A[l] \leq a \leq A[l]$ und somit $A[l] = a$. Binary Search gibt also in Zeile 9 die richtige Antwort.

Laufzeit

Wir zählen die Vergleiche $V(n)$ von Binary Search zwischen Elementen von A. Der einzige Vergleich findet in Zeile 7 statt (abgesehen vom Schluss in Zeile 9), d.h. ein Vergleich pro Schleifendurchlauf. Es gilt also $V(n) = k$. Damit ist auch die Laufzeit von Binary Search proportional zu $V(n)$.

Am Anfang gilt $r - l + 1 = n$. Gemäß der Invarianten (i) halbiert sich dieser Wert bei jedem Schleifendurchlauf. Die Schleife wird abgebrochen wenn $l = r$ gilt, also wenn $r - l + 1 = 1$ ist. Folglich muss $n/2^k = 2^{p-k} = 1$ sein, und somit $k = p$. Da $p = \log n$ macht Binary Search also insgesamt $V(n) = \log n$ Vergleiche.

Allgemeines n

Wenn n keine keine Zweierpotenz ist, ist der Ausdruck $\frac{l+r-1}{2}$ in Zeile 6 im Allgemeinen keine ganze Zahl. Da m ein Index im Feld A ist, muss m auf jeden Fall ganzzahlig sein. Wir runden den Ausdruck einfach auf die nächst kleinere ganze Zahl ab. D.h. wir ändern Zeile 6 wie folgt:

$6'$ $\quad m \leftarrow \left\lfloor \frac{l+r-1}{2} \right\rfloor$

Der Rest von Binary Search bleibt unverändert.

Damit bleibt Binary Search natürlich korrekt. Für die Anzahl der Vergleiche $V(n)$ war oben entscheidend, dass sich die aktuelle Intervalllänge $r - l + 1$ in jeder Runde halbierte (Invariante (i)). Durch das Abrunden von m stimmt das zwar noch ungefähr, aber eventuell nicht mehr exakt.

Die Anzahl der Vergleiche von Binary Search verhält sich monoton wachsend: Nehmen wir an, wir vergrößern die Eingabe, d.h. wir geben ein Feld mit $n' > n$ Elementen ein. Dann vergrößern sich auch die Intervalle $r - l + 1$ in jeder Runde, und damit die Anzahl der Runden.

Sei $2^p < n < 2^{p+1}$. Wenn wir also statt n Elemente die nächst größere Zweierpotenz $n' = 2^{p+1}$ an Elementen eingeben, machen wir exakt $p + 1$ Vergleiche und dies sind mindestens so viele wie bei Eingabe von n Elementen. Allerdings machen wir auch mehr wie p Vergleiche, da dies die exakte Anzahl bei 2^p Elementen war. Da $p + 1 = \lceil \log n \rceil$, macht Binary Search im Allgemeinen folglich $V(n) = \lceil \log n \rceil$ Vergleiche.

↓ Ende der 9. Vorlesungswoche

Kombinatorik 6

Die Anzahl der Möglichkeiten verschiedene Dinge miteinander zu kombinieren ist ein zentrales Thema der *Kombinatorik*. Typische Fragestellungen sind:

- Wie viele Möglichkeiten gibt es drei Kugeln Eis in der Eisdiele auszuwählen?
- Wie viele Möglichkeiten gibt es für die 6 Zahlen einer Lottoziehung?

Die Lottoziehung motiviert ein grundlegendes kombinatorisches Experiment. Wir betrachten eine Urne mit n Bällen. Die Bälle seien von 1 bis n nummeriert. Wir ziehen k der Bälle. Bei der Lottoziehung ist $n = 49$ und $k = 6$.

Wir betrachten verschiedene Varianten einer solchen Ziehung. Beim Lotto wird ein einmal gezogener Ball nicht wieder in die Urne zurückgelegt. Wir werden auch die Variante betrachten, bei der ein gezogener Ball wieder zurückgelegt wird. Außerdem kommt es im Lotto nicht auf die Reihenfolge an in der die Bälle gezogen werden. Das Ergebnis wird in der Regel einfach gemäß aufsteigender Ballnummer angegeben, also z.B. als 3, 4, 19, 25, 31, 39. Die Reihenfolge in der die Bälle gezogen wurden kann aber eine ganz andere sein. Wir werden beide Varianten betrachten, also mit und ohne Beachtung der Reihenfolge.

Die Varianten bzgl. Zurücklegen und Reihenfolge ergeben 4 verschiedene Experimente die wir nun der Reihe nach untersuchen wollen.

6.1 Ziehen mit Zurücklegen und mit Reihenfolge

Grundlagen Mathematik | 06.01: Ziehen mit Zurücklegen und mit Reihenfolge

Wir betrachten eine Ziehung, bei der der gezogene Ball nach jedem Zug zurückgelegt wird. Außerdem beachten wir die Reihenfolge, in der die Bälle gezogen wurden. D. h. wir unterscheiden beispielsweise die Ziehung 1, 2, 3 von der Ziehung 3, 2, 1.

Für kleine Werte für n und k kann man die möglichen Ergebnisse noch übersichtlich auflisten. Nehmen wir als Beispiel $n = 3$ und $k = 2$. Folgende Tabelle zeigt alle möglichen Ergebnisse der Ziehungen:

$$\begin{array}{lll} (1,1) & (1,2) & (1,3) \\ (2,1) & (2,2) & (2,3) \\ (3,1) & (3,2) & (3,3) \end{array}$$

Die Ergebnisse $(1,1),\ (2,2),\ (3,3)$ sind möglich, da wir den gezogenen Ball zurücklegen. Insgesamt gibt es also 9 mögliche Resultate einer Ziehung.

Beim Ziehen von k Bällen aus einer Urne mit n Bällen, die mit $1, \ldots, n$ nummeriert sind, mit Zurücklegen und unter Beachtung der Reihenfolge ergibt sich als *Grundmenge* der möglichen Ereignisse

$$\begin{aligned}\Omega &= \big\{(a_1, \ldots, a_k) \mid a_i \in \{1, \ldots, n\} \text{ für alle } i = 1, \ldots, k\big\} \\ &= \{1, \ldots, n\}^k.\end{aligned}$$

Wir versuchen eine allgemeine Formel für die Anzahl der Möglichkeiten aufzustellen. Dazu betrachten wir die Anzahl der möglichen Ergebnisse pro Ball den wir ziehen: Wenn wir den ersten Ball ziehen, gibt es offenbar n Möglichkeiten. Da wir den gezogenen Ball wieder zurücklegen gibt es beim zweiten Ball ebenfalls n mögliche Ergebnisse. Da wir auch die Reihenfolge beachten, ergibt dies $n \cdot n = n^2$ mögliche Kombinationen für die ersten beiden Bälle. Wenn wir den dritten Ball ziehen gibt es wiederum n mögliche Ergebnisse, und die Gesamtzahl der Kombinationen für die ersten drei Bälle steigt auf $n \cdot n \cdot n = n^3$. So geht es weiter bis zum k-ten Ball bei dem es immer noch n Möglichkeiten gibt. Das ergibt dann insgesamt n^k mögliche Ergebnisse.

Theorem 6.1.1. *Bei einer Ziehung von k aus n Bällen, mit Zurücklegen und mit Beachtung der Reihenfolge, gibt es n^k mögliche Ergebnisse.*

Für die Randfälle, wenn n oder k Null sind, legen wir folgende Konvention fest:

$$n^0 := 1, \text{ für alle } n \in \mathbb{N}.$$

Insbesondere ist damit auch $0^0 = 1$. Dagegen ist $0^k = 0$ für $k \neq 0$. Diese Festlegung gilt übrigens nicht nur für natürliche Zahlen sondern auch reelle k und n.

Bei der Anwendung dieser Formel besteht die Kunst darin, eine evtl. ganz anders formulierte kombinatorische Aufgabe als das Ziehen-von-Bällen darzustellen. Als Beispiel wollen wir die Anzahl der Wörter der Länge 5 über dem Alphabet a, b, c, . . . , z bestimmen. Das hat zunächst nichts mit Bällen zu tun. Aber wir können das Problem in ein Ziehen-von-Bällen übersetzen: Wir nehmen $n = 26$ Bälle und beschriften sie mit a, b, c, . . . , z anstatt mit $1, 2, \ldots, 26$. Um ein Wort der Länge 5 zu erhalten ziehen wir nun $k = 5$ mal und setzen die Buchstaben auf den gezogenen Bällen in der Reihenfolge ihrer Ziehung zu einem Wort zusammen. Die Ziehung erfolgt

- mit Zurücklegen, da jeder Buchstabe auch mehrfach in dem Wort vorkommen darf, und
- mit Beachtung der Reihenfolge, da z.B. “spass” und “passs” verschiedene Wörter sind.

Es gibt genauso viele der gesuchten Wörter wie Ergebnisse dieser Ziehung. Dies sind nach Theorem 6.1.1 also $n^k = 26^5 = 11.881.376$. Wenn wir k variabel lassen haben wir also das Ergebnis: es gibt 26^k Wörter der Länge k über dem Alphabet a, b, c, . . . , z.

Wir ändern obige Aufgabe leicht ab und betrachten statt dem Alphabet a, b, c, . . . , z mit 26 Zeichen das binäre Alphabet das nur aus den zwei Zeichen 0 und 1 besteht. Dann können wir analog die Anzahl der 0-1-Wörter der Länge k berechnen. Der einzige Unterschied ist, dass wir nur

noch $n = 2$ Bälle haben die wir mit 0 und 1 beschriften. Das ergibt dann 2^k 0-1-Wörter der Länge k. Diese Wörter kann man auch als Binärzahlen auffassen.

Korollar 6.1.2. *Die Anzahl der k-stelligen Binärzahlen ist* 2^k.

Als ein weiteres Beispiel berechnen wir die Anzahl der Funktionen $f : A \to B$ auf zwei endlichen Mengen A und B. Sei $A = \{a_1, a_2, \dots a_k\}$ und $B = \{b_1, b_2, \dots, b_n\}$. Zum Beispiel für $A = [5]$ und $B = [3]$ können wir eine Funktion $f : A \to B$ durch folgende Wertetabelle darstellen.

A	1	2	3	4	5
f	3	2	1	2	3

Die Frage ist nun, wie viele solche Funktionen es gibt. Oder anders formuliert: Wie viele Möglichkeiten gibt es eine solche Wertetabelle auszufüllen?

Auch dies lässt sich als ein Ziehen-von-Bällen darstellen: Eine Funktion $f : A \to B$ ordnet jedem Element von A ein Element von B zu. Wir nehmen n Bälle und beschriften sie mit den Elementen $b_1, b_2, \dots, b_n$ von B. Die Wertetabelle für eine Funktion f füllen wir nun durch k-maliges Ziehen aus. Den Wert auf dem ersten Ball ordnen wir a_1 zu, den Wert vom zweiten Ball a_2, und so weiter, bis zum k-ten Ball, dessen Wert a_k zugeordnet wird. Die Ziehung erfolgt

- mit Zurücklegen, da jedes Element von B beliebig oft als Funktionswert vorkommen darf, und
- mit Beachtung der Reihenfolge, da wir die Zuordnung genau in der Reihenfolge $a_1, a_2, \dots, a_k$ machen.

Es gibt genauso viele Funktionen von A nach B wie Ergebnisse dieser Ziehung, also $n^k = |B|^{|A|}$ nach Theorem 6.1.1.

Korollar 6.1.3. *Die Anzahl der Funktionen* $f : A \to B$ *ist* $|B|^{|A|}$.

Ein Spezialfall sind Boolesche Funktionen. Eine Funktion $f : \{0,1\}^k \to \{0,1\}$ heißt *k-stellige boolesche Funktion*. Z. B. kann die logische und-Verknüpfung als 2-stellige Boolesche Funktion aufgefasst werden:

$$\{0,1\}^2 \ni (x, y) \longmapsto x \wedge y \in \{0,1\}.$$

Die Menge $A = \{0,1\}^k$ besteht bekanntlich aus allen 0-1-Tupeln der Länge k. Nach Folgerung 6.1.2 ist $|A| = 2^k$. Die Menge $B = \{0,1\}$ hat zwei Elemente. Nach Folgerung 6.1.3 ist die Anzahl der k-stelligen Booleschen Funktionen $|B|^{|A|} = 2^{2^k}$. Es gibt also beispielsweise $2^{2^2} = 2^4 = 16$ zweistellige Boolesche Funktionen.

6.2 Ziehen ohne Zurücklegen und mit Reihenfolge

Grundlagen Mathematik | 06.02: Ziehen ohne Zurücklegen und mit Reihenfolge

Wir betrachten eine Ziehung, bei der der gezogene Ball *nicht* zurückgelegt wird, aber weiterhin die Reihenfolge beachtet wird, in der die Bälle

gezogen wurden. Für $n = 3$ und $k = 2$ zeigt folgende Tabelle alle möglichen Ergebnisse der Ziehungen:

$$\begin{array}{ccc} & (1,2) & (1,3) \\ (2,1) & & (2,3) \\ (3,1) & (3,2) & \end{array}$$

Im Vergleich zur Ziehung mit Zurücklegen fehlen die Diagonalelemente $(1,1)$, $(2,2)$, $(3,3)$. Diese sind jetzt nicht mehr möglich, da wir den gezogenen Ball nicht zurücklegen. Insgesamt gibt es also 6 mögliche Resultate einer Ziehung.

Die allgemeine Grundmenge ergibt sich zu

$$\Omega = \big\{(a_1, \ldots, a_k) \mid a_i \in \{1, \ldots, n\} \text{ und } a_i \neq a_j \text{ für } i \neq j\big\}.$$

Um eine allgemeine Formel für die Anzahl der Möglichkeiten herzuleiten betrachten wir wieder die Anzahl der möglichen Ergebnisse pro Ball den wir ziehen: Wenn wir den ersten Ball ziehen, gibt es offenbar n Möglichkeiten. Da wir den gezogenen Ball diesmal nicht zurücklegen gibt es beim zweiten Ball nur noch $n - 1$ Bälle in der Urne. Da wir die Reihenfolge beachten, ergibt dies $n(n-1)$ mögliche Kombinationen für die ersten beiden Bälle. Wenn wir den dritten Ball ziehen sind nur noch $n - 2$ Bälle übrig, und die Gesamtzahl der Kombinationen für die ersten drei Bälle ist $n(n-1)(n-2)$. So geht es weiter bis zum k-ten Ball bei dem noch $n - k + 1$ Bälle in der Urne sind. Das ergibt dann insgesamt $n(n-1)(n-2)\cdot\ldots\cdot(n-k+1)$ mögliche Ergebnisse.

Der Ausdruck $n(n-1)(n-2)\cdot\ldots\cdot(n-k+1)$ wird als *fallende Faktorielle von n der Länge k* bezeichnet. Das k-fache Produkt $n \cdot n \cdot \ldots \cdot n$ wird mit n^k abkürzt. Analog dazu definieren wir für die fallende Faktorielle die Abkürzung

$$n^{\underline{k}} := n(n-1)(n-2)\cdot\ldots\cdot(n-k+1)\,,$$

das k ist hier also unterstrichen. Für den Randfall $k = 0$ legen wir fest: $n^{\underline{0}} = 1$.

Beispiel 6.2.1. Es ist also $n^{\underline{2}} = n(n-1)$ und $n^{\underline{1}} = n$.

Bemerkung 6.2.2. *In der Literatur findet sich auch die Bezeichnung $(n)_k$ für die fallende Faktorielle der Länge k. Da wir allerdings mit Zahlenwertsystemen in dieser Vorlesung arbeiten, verwenden wir diese Notation nicht.*

Unsere obigen Überlegungen lassen sich nun wie folgt zusammenfassen:

Theorem 6.2.3. *Bei einer Ziehung von k aus n Bällen, ohne Zurücklegen und mit Beachtung der Reihenfolge, gibt es $n^{\underline{k}}$ mögliche Ergebnisse.*

Da wir hier nicht zurücklegen, können wir höchstens n mal ziehen, dann ist die Urne leer. In dem Ausdruck $n^{\underline{k}}$ können wir aber trotzdem Werte $k > n$ einsetzen:

$$2^{\underline{4}} = 2 \cdot 1 \cdot 0 \cdot (-1) = 0.$$

Da das Produkt dann die Null durchläuft, ist das Ergebnis immer Null. Das kann man auch kombinatorisch sinnvoll interpretieren: es gibt 0 Möglichkeiten ohne Zurücklegen 4 Bälle aus 2 zu ziehen (mit anderen Worten: $\Omega = \emptyset$). Man beachte auch, dass die Formel $n^{\underline{k}} = n(n-1)\cdots(n-k+1)$ auch dann definiert ist, wenn man für n reelle (oder komplexe) Werte einsetzt.

Auf komplexe Zahlen sind wir in dieser Vorlesung bisher nicht eingegangen. Wir verweisen auf andere Veranstaltungen.

Dann geht die kombinatorische Interpretation verloren. Man kann $n^{\underline{k}}$ dann als ein reelles Polynom vom Grad k betrachten. Diese Sichtweise ist in manchen Argumenten sehr hilfreich.

Wir betrachten noch einmal obiges Beispiel, die Anzahl der Wörter der Länge 5 über dem Alphabet a, b, c, . . . , z zu bestimmen. Im Unterschied zu oben soll jetzt aber jeder Buchstabe in einem Wort höchstens einmal vorkommen. Dies können wir in unserem Experiment dadurch modellieren, dass wir die Bälle nicht zurücklegen. D.h. wir können Theorem 6.2.3 anwenden und erhalten $26^{\underline{5}} = 26 \cdot 25 \cdot 24 \cdot 23 \cdot 22 = 7.893.600$ viele Wörter der Länge 5 mit jeweils unterschiedlichen Buchstaben.

Analog können wir in obigem Beispiel der Anzahl der Funktionen $f : A \to B$ verfahren. Dort haben wir die Bälle mit den Elementen von B beschriftet und durch die Ziehungen den Elementen von A Funktionswerte zugeordnet. Wenn wir die Bälle nun nicht zurücklegen, wird also jedes Element von B höchstens einmal als Funktionswert benutzt. Folglich ist die Funktion die dadurch entsteht *injektiv*.

Korollar 6.2.4. *Die Anzahl der injektiven Funktionen $f : A \to B$ ist $|B|^{\underline{|A|}}$.*

Einen interessanten Spezialfall erhalten wir für $A = B = [n]$. Die injektiven Funktionen von $[n]$ nach $[n]$ sind die Permutationen auf $[n]$, die wir mit $\mathfrak{S}_n$ bezeichnet haben (vgl. Proposition 4.4.5 (2)). Deren Anzahl ergibt sich nun nach Folgerung 6.2.4 als

$$n^{\underline{n}} = n(n-1)\cdot\ldots\cdot 2\cdot 1.$$

Diesen Ausdruck bezeichnet man als n *Fakultät* und er erhält eine eigene Abkürzung mit einem Ausrufezeichen nach dem n,

$$n! := n(n-1)\cdot\ldots\cdot 2\cdot 1.$$

Zur Erinnerung: Den Randfall $n = 0$ definieren wir als $0! := 1$.

Korollar 6.2.5. *Es gibt $n!$ Permutationen auf $[n]$.*

6.3 Ziehen ohne Zurücklegen und ohne Reihenfolge

Grundlagen Mathematik | 06.03: Ziehen ohne Zurücklegen und ohne Reihenfolge

Im Unterschied zum vorherigen Fall unterscheiden wir nun nicht mehr die Reihenfolge in der die Bälle gezogen werden. Dies entspricht der Lotto-Ziehung. Für $n = 3$ und $k = 2$ bleiben drei Ergebnisse übrig

$$(1,2) \quad (1,3) \quad (2,3)$$

Im Vergleich zur Ziehung mit Reihenfolge fehlen jetzt die Ergebnisse $(2,1)$, $(3,1)$, $(3,2)$ da dies nur Permutationen der bereits aufgezählten Ergebnisse sind. Dies ist auch bereits die Beobachtung, die den allgemeinen Fall löst: wenn wir k Elemente haben, dann gibt es nach Folgerung 6.2.5 genau $k!$ Permutationen dieser Elemente. Bei Beachtung der Reihenfolge sind dies auch $k!$ verschiedene Ergebnisse. Ohne Beachtung der Reihenfolge werden die $k!$ Permutationen nur als ein Ergebnis gezählt. D.h. die $n^{\underline{k}}$ Ergebnisse bei Ziehung mit Beachtung der Reihenfolge, lassen sich in Gruppen der Größe $k!$ einteilen, wobei jede Gruppe aus den Permuationen von k Zahlen besteht. Die Anzahl der Gruppen ist $\frac{n^{\underline{k}}}{k!}$, und dies ist auch die Anzahl der Ergebnisse bei einer Ziehung ohne Beachtung der Reihenfolge.

Wir definieren eine Abkürzung für diesen Ausdruck:

$$\binom{n}{k} := \frac{n^{\underline{k}}}{k!}\,.$$

Die linke Seite wird gelesen als *n über k* (engl. *n choose k*) und wird auch als *Binomialkoeffizient* bezeichnet. Woher dieser Name kommt werden wir später sehen.

Als Randfälle haben wir für alle $n \in \mathbb{N}$:

$$\binom{n}{0} = \frac{n^{\underline{0}}}{0!} = \frac{1}{1} = 1.$$

Insbesondere ist also auch $\binom{0}{0} = 1$. Eine sinnvolle Erweiterung für k auf die ganzen Zahlen ist die Festlegung $\binom{n}{k} := 0$ für $k < 0$.

Ist n eine nicht-negative ganze Zahl und $n \geq k$, so ist

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

Theorem 6.3.1. *Bei einer Ziehung von k aus n Bällen, ohne Zurücklegen und ohne Beachtung der Reihenfolge, gibt es $\binom{n}{k}$ mögliche Ergebnisse.*

In obigem Beispiel war $n = 3$ und $k = 2$ und es gilt

$$\binom{3}{2} = \frac{3^{\underline{2}}}{2!} = \frac{3 \cdot 2}{2 \cdot 1} = 3.$$

Beim Lotto haben wir $n = 49$ Bälle und ziehen $k = 6$ mal. Dies ergibt

$$\binom{49}{6} = \frac{49^{\underline{6}}}{6!} = 13.983.816,$$

also knapp 14 Millionen mögliche Ergebnisse. Die Chancen bei zufälliger Ziehung 6 Richtige zu haben sind also ungefähr 1:14 Millionen. Für den Jackpot braucht man auch noch die richtige Superzahl. Dafür gibt es noch einmal 10 Möglichkeiten. Die Gewinnchancen sinken damit auf 1:140 Millionen. Deutschland und Frankreich haben zusammen etwas über 140 Millionen Einwohner. Wenn man einen Einwohner aus diesen beiden Ländern zufällig auswählt, dann hat man ungefähr die gleiche

Chance gewählt zu werden wie im Lotto den Jackpot zu gewinnen. Wir folgern: Mathematiker spielen in der Regel kein Lotto!

Lotto-Spielen wird daher von zynischen Zungen auch als Dummen-Steuer bezeichnet.

In unseren Beispielen bei den Ziehungen mit Beachtung Reihenfolge konnten wir die Ergebnisse als Wörter oder Zahlen interpretieren, da mit der Ziehung eines Buchstabens oder einer Ziffer auch deren Platz definiert war. Ohne Reihenfolge kommt es nur noch darauf an, was überhaupt gezogen wurde. Wir können das Ergebnis also als *Menge* interpretieren. So ist hier nämlich

$$\Omega = \big\{(a_1, \dots, a_k) \mid a_i \in \{1, \dots, n\} \text{ und } a_1 < a_2 < \cdots < a_k\big\}.$$

Man kann sich Ω alternativ eben auch als die Menge aller ungeordneten k-elementigen Teilmengen von $\{1, \dots, n\}$ vorstellen, also

$$\Omega = \big\{\{a_1, \dots, a_k\} \mid a_i \in \{1, \dots, n\} \text{ und } a_i \neq a_j \text{ für } i \neq j\big\}.$$

Bei Mengen spielt es keine Rolle in welcher Reihenfolge die Elemente aufgezählt werden. Wenn wir aus einer Menge mit n Elementen k-mal ziehen, dann erhalten wir alle Teilmengen der Größe k als mögliche Ergebnisse der Ziehung.

Korollar 6.3.2. *Eine Menge mit n Elementen hat $\binom{n}{k}$ Teilmengen der Größe k.*

Nehmen wir speziell die Menge $[n]$ und interpretieren die Zahlen als Platznummern. D. h. wir ziehen k Platznummern. Damit konstruieren wir eine 0-1-Sequenz der Länge n indem wir genau auf den gezogenen Plätzen eine 1 setzen, und auf die restlichen Plätze 0. Die möglichen Ergebnisse erzeugen damit alle 0-1-Sequenzen der Länge n mit genau k Einsen.

Korollar 6.3.3. *Es gibt $\binom{n}{k}$ 0-1-Sequenzen der Länge n mit genau k Einsen.*

Nullen und Einsen kann man im obigen Argument natürlich vertauschen. Es gibt also auch $\binom{n}{k}$ 0-1-Sequenzen der Länge n mit genau k Nullen. Eine 0-1-Sequenzen der Länge n mit k Nullen hat $n - k$ Einsen. Jede Sequenz mit k Nullen definiert also eine Sequenz mit $n - k$ Einsen, und umgekehrt. Von beiden Sequenzen gibt es also gleich viele. Nach Folgerung 6.3.3 gibt es $\binom{n}{n-k}$ 0-1-Sequenzen der Länge n mit $n - k$ Einsen. D. h. für $0 \leq k \leq n$ gilt folgende *Symmetrie*

$$\binom{n}{k} = \binom{n}{n-k}. \tag{6.1}$$

Wenn wir die Mengen der 0-1-Sequenzen der Länge n mit genau k Einsen alle vereinigen, für $k = 0, 1, \dots, n$, dann erhalten wir *alle* 0-1-Sequenzen der Länge n. Davon gibt es nach Folgerung 6.1.2 insgesamt 2^n viele. Folglich muss die Summe der $\binom{n}{k}$-Werte 2^n ergeben:

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n. \tag{6.2}$$

Dies ist ein Spezialfall des Binomialtheorems / Binomischen Lehrsatzes.

Als ein weiteres Beispiel zählen wir kürzeste Wege in Gittern. Wir betrachten ein rechteckiges Gitter mit Seitenlängen a und b. Mit $w(s,t)$ bezeichnen wir die Anzahl der kürzesten Wege vom unteren linken Eckpunkt s zum oberen rechten Eckpunkt t. Abbildung 6.1 zeigt die Situation.

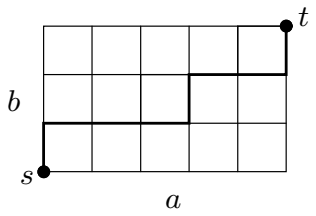


Abbildung 6.1: Ein Gitter mit Seitenlängen a und b ist ein ungerichteter Graph der sich wie abgebildet zeichnen lässt. Die Gitterpunkte sind die Knoten. Kürzeste Wege von s nach t sind Wege, die nur nach rechts oder nach oben gehen.
Wir schreiben 0 für einen Schritt nach oben und 1 für einen Rechtsschritt. Der dick eingezeichnete Weg wir dann durch die Sequenz 01110110 beschrieben.

Ein kürzester Weg von s nach t macht an jedem Gitterpunkt entweder einen Schritt nach rechts oder nach oben. Die Länge ist $a + b$, da man insgesamt a Schritte nach rechts und b Schritte nach oben machen muss. Die Wege unterscheiden sich nur in der Reihenfolge, wann nach rechts bzw. nach oben gegangen wird. Wir können die kürzesten Wege durch 0-1-Sequenzen der Länge $a + b$ kodieren: ein Schritt nach rechts wird durch eine 1 repräsentiert, ein Schritt nach oben durch eine 0. Es entsteht eine 0-1-Sequenz der Länge $a + b$ mit genau a Einsen und b Nullen. Jede solche 0-1-Sequenz repräsentiert einen kürzesten Weg von s nach t. Aus Folgerung 6.3.3 erhalten wir $w(s,t)$.

Korollar 6.3.4. *Es gibt $w(s,t) = \binom{a+b}{a}$ kürzeste Wege von s nach t in einem $a \times b$-Gitter.*

Natürlich kann man anstatt die Plätze der Rechtsschritte genauso die der Schritte nach oben betrachten. In den 0-1-Sequenzen ist dies die Lage der Nullen. Die Anzahl der Möglichkeiten bleibt dabei gleich. Dies ist die oben erwähnte Symmetrieformel (6.1). Sie lautet hier

$$\binom{a+b}{a} = \binom{a+b}{b}.$$

Formel (6.1) erhält man daraus wenn man $n = a+b$ und $k = a$ definiert.

Freiwilliges Übungsblatt 05

Bearbeiten Sie bitte das fünfte freiwillige Übungsblatt. Wir vertiefen dort unser Wissen über Graphen und Induktion weiter.

↓ **Ende der 10. Vorlesungswoche**

Besprechung Blatt 05

Für das Freiwillige Blatt 05 erhalten Sie wieder eine handschriftliche Musterlösung (unter den Modulen in Canvas).

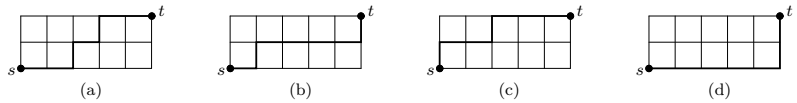


Abbildung 6.2: Ein Gitter mit Seitenlängen $k = 5$ und $n - 1 = 2$. Die dick eingezeichneten Wege repräsentieren die Ziehungen (a) 11233, (b) 12222, (c) 22333 und (d) 11111. Die Anzahl der horizontalen Schritte auf der i-ten Linie von unten gibt die Anzahl an, wie oft Ball i gezogen wurde, für $i = 1, \dots, n$.

6.4 Ziehen mit Zurücklegen und ohne Reihenfolge

Grundlagen Mathematik | 06.04: Ziehen mit Zurücklegen und ohne Reihenfolge

Da wir nun die Bälle wieder zurücklegen kommen in unserem Beispiel mit $n = 3$ und $k = 2$ die Diagonalelemente wieder hinzu:

$$\begin{array}{ccc} (1,1) & (1,2) & (1,3) \\ & (2,2) & (2,3) \\ & & (3,3) \end{array}$$

Insgesamt gibt es also 6 mögliche Resultate einer Ziehung.

Wir erweitern das Beispiel und ziehen $k = 5$ mal aus $n = 3$ Bällen. Typische Resultate einer Ziehung sind

$$\begin{array}{c} 11233 \\ 12222 \\ 22333 \\ 11111 \end{array}$$

Die Ergebnisse sind jeweils aufsteigend sortiert angegeben. Die tatsächliche Reihenfolge in der die Bälle gezogen wurden spielt hier keine Rolle. D. h. es kommt nur darauf an, wie oft eine Zahl gezogen wurde.

Wir können die Ergebnisse als kürzeste Wege in Gittern darstellen: wir betrachten ein $k \times (n-1)$ Gitter. Bei einer Höhe von $n - 1$ hat das Gitter n horizontale Linien. Diese Linien entsprechen von unten her den Bällen $1, 2, \dots, n$. Die Rechtsschritte in jeder Zeile auf einem kürzesten Weg von s nach t sind die Anzahlen, wie oft der jeweilige Ball gezogen wurde. Bei der Seitenlänge k ziehen wir dabei insgesamt k-mal. Abbildung 6.2 zeigt die Wege für obige Beispiele.

Es gibt also genauso viele Ergebnisse bei dieser Ziehung wie es kürzeste Wege in einem $k \times (n-1)$ Gitter gibt. Nach Folgerung 6.3.4 sind dies $\binom{n+k-1}{k}$ viele.

Theorem 6.4.1. *Bei einer Ziehung von k aus n Bällen, mit Zurücklegen und ohne Beachtung der Reihenfolge, gibt es* $\binom{n+k-1}{k}$ *mögliche Ergebnisse.*

In obigem Beispiel mit $n = 3$ und $k = 2$ erhalten wir $\binom{3+2-1}{2} = \binom{4}{2} = \frac{4 \cdot 3}{2 \cdot 1} = 6$, und für $k = 5$ ergibt sich $\binom{3+5-1}{5} = \binom{7}{5} = \binom{7}{2} = \frac{7 \cdot 6}{2 \cdot 1} = 21$. Dabei haben wir in der Mitte die Symmetrie (6.1) der Binomialkoeffizienten benutzt.

Die Ergebnisse bei dieser Ziehung können wir auch etwas kompakter darstellen, indem wir für jeden Ball angeben, wie oft er gezogen wurde, also durch Werte $x_1, x_2, \dots, x_n$. In obigem Beispiel mit $k = 5$ und $n = 3$ geben wir dann die Ziehung 11233 durch $x_1 = 2$, $x_2 = 1$ und

$x_3 = 2$ an. Die Anzahl der gezogenen Bälle erkennt man an der Summe: $x_1 + x_2 + x_3 = 5 = k$. Bei der Ziehung 12222 erhalten wir $x_1 = 1$, $x_2 = 4$ und $x_3 = 0$. Jede Ziehung entspricht also einer Gleichung der Form

$$x_1 + x_2 + \cdots + x_n = k, \qquad 0 \leq x_i \leq k. \tag{6.3}$$

Die Zerlegung einer Zahl k in eine feste Anzahl n von Summanden $x_i \geq 1$ nennt man eine *Zahlpartition*. In unserer Variante hier gilt lediglich $x_i \geq 0$. Wir betrachten hier eine *geordnete Zahlpartition*, da wir die Reihenfolge der x_i's beachten. Wir unterscheiden hier z.B. zwischen den Darstellungen $1 + 4 = 5$ und $4 + 1 = 5$.

Korollar 6.4.2. *Es gibt* $\binom{n+k-1}{k}$ *Möglichkeiten eine Zahl* k *als Summe von* n *Zahlen wie in Gleichung* (6.3) *zu schreiben.*

Die Zahlpartitionen lassen sich als mögliche Ergebnisse einer Wahl interpretieren: Es gibt k Wähler und n Kandidaten. Sei x_i die Anzahl der Stimmen die Kandidat i bekommen hat. Dann muss $x_1 + x_2 + \cdots + x_n = k$ gelten. Es gibt also $\binom{n+k-1}{k}$ mögliche Wahlergebnisse.

Man kann die Anzahl der Wahlergebnisse auch ohne Zahlpartitionen direkt mit dem Ziehen von Bällen erklären. Wir betrachten ein Beispiel. Die 50 Studenten eines Semesters wählen einen Semestersprecher. Es stehen 3 Kandidaten zur Wahl. Die 3 Kandidaten werden durch $n = 3$ Bälle repräsentiert. Eine Stimmabgabe bei der Wahl entspricht dann dem Ziehen eines Balls. D. h. es wird $k = 50$ mal gezogen. Die Ziehung erfolgt

- mit Zurücklegen, da jeder Kandidat beliebig oft gewählt werden kann, und
- ohne Beachtung der Reihenfolge, da die Reihenfolge in der ein Kandidat seine Stimmen erhält keine Rolle spielt.

Nach Theorem 6.4.1 ist die Anzahl der möglichen Wahlergebnisse

$$\binom{n+k-1}{k} = \binom{52}{50} = \binom{52}{2} = \frac{52 \cdot 51}{2} = 1326.$$

Dabei haben wir in der Mitte wieder die Symmetrie (6.1) benutzt.

6.5 Eigenschaften der Binomialkoeffizienten

Grundlagen Mathematik | 06.05: Pascalsches Dreieck – Eigenschaften der Binomialkoeffizienten

Die Anzahl der Ergebnisse bei den beiden Experimente ohne Beachtung der Reihenfolge der gezogenen Bälle sind Binomialkoeffizienten. Diese spielen eine wichtige Rolle in der Kombinatorik. Wir geben hier noch einmal die Definition an. Dabei sind $n, k \in \mathbb{Z}$, wobei wir bereits erwähnt haben, dass diese Voraussetzung für n nicht notwendig ist, denn n dürfte beispielsweise auch reell sein.

Definition 6.5.1. Für $n, k \in \mathbb{Z}$ ist

$$\binom{n}{k} := \begin{cases} \frac{n^{\underline{k}}}{k!} & \text{für } k > 0 \\ 1 & \text{für } k = 0 \\ 0 & \text{für } k < 0. \end{cases}$$

Folgende Tabelle zeigt die Werte $\binom{n}{k}$ für $n = 0, 1, \ldots, 10$. Für $k > n$ ist $\binom{n}{k} = 0$. Diese Werte sind zu Gunsten der Übersichtlichkeit nicht eingetragen. So entsteht das in Abbildung 6.3 gezeigte Zahlenfeld in der Form eines Dreiecks, das nach Blaise Pascal benannte *Pascalsche Dreieck*.

n	$\binom{n}{0}$	$\binom{n}{1}$	$\binom{n}{2}$	$\binom{n}{3}$	$\binom{n}{4}$	$\binom{n}{5}$	$\binom{n}{6}$	$\binom{n}{7}$	$\binom{n}{8}$	$\binom{n}{9}$	$\binom{n}{10}$
0	1										
1	1	1									
2	1	2	1								
3	1	3	3	1							
4	1	4	6	4	1						
5	1	5	10	10	5	1					
6	1	6	15	20	15	6	1				
7	1	7	21	35	35	21	7	1			
8	1	8	28	56	70	56	28	8	1		
9	1	9	36	84	126	126	84	36	9	1	
10	1	10	45	120	210	252	210	120	45	10	1

Abbildung 6.3: Das Pascalsche Dreieck. Die Einträge an den freigelassenen Plätzen sind 0.

Die 1-Spalte ganz links im Dreieck kommt von der Definition $\binom{n}{0} = 1$. Nach der Symmetrieformel (6.1) ist $\binom{n}{0} = \binom{n}{n-0} = \binom{n}{n}$. Die Werte $\binom{n}{n}$ stehen in der ersten Diagonalen im Dreieck und sind somit ebenfalls alle 1. In der zweiten Spalte stehen die Werte $\binom{n}{1} = n$, die dann wegen der Symmetrie auch in der zweiten Diagonalen stehen, $\binom{n}{n-1} = n$.

Es gibt eine Vielzahl von Beziehungen zwischen den Zahlen Pascalschen Dreieck von denen wir einige herleiten wollen. Wir beginnen mit einigen einfachen Gleichungen für Binomialkoeffizienten.

Fakultätendarstellung. Die *Fakultätendarstellung* der Binomialkoeffizienten ist eine Formel, in der alle Ausdrücke Fakultäten sind. Man erhält sie durch eine einfache Erweiterung:

$$\binom{n}{k} = \frac{n^{\underline{k}}}{k!} = \frac{n(n-1)\cdots(n-k+1)}{k!} \cdot \frac{(n-k)!}{(n-k)!} = \frac{n!}{k!\,(n-k)!}.$$

Die Fakultätendarstellung lautet

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}, \qquad 0 \le k \le n \text{ ganzzahlig.} \tag{6.4}$$

Wir haben oben erwähnt, dass man im Binomialkoeffizienten $\binom{n}{k}$ für n beispielsweise auch eine reelle Zahl einsatzen kann. Die Fakultätendarstellung gilt aber nur wenn n eine natürliche Zahl ist, da sonst der Ausdruck $(n-k)!$ im Nenner nicht definiert wäre.

Absorbtions- und Extraktionsregel. Eine andere nützliche Eigenschaft ist, dass man den Bruch $\frac{n}{k}$ aus dem Binomialkoeffizienten $\binom{n}{k}$ herausziehen kann. Dabei muss natürlich $k \neq 0$ sein:

$$\begin{aligned}
\binom{n}{k} &= \frac{n^{\underline{k}}}{k!} \\
&= \frac{n\,(n-1)\cdots(n-k+1)}{k!} \\
&= \frac{n\,(n-1)\cdots(n-k+1)}{k\,(k-1)!} \\
&= \frac{n}{k} \cdot \frac{(n-1)\cdots(n-k+1)}{(k-1)!} \\
&= \frac{n}{k} \cdot \frac{(n-1)^{\underline{k-1}}}{(k-1)!} \\
&= \frac{n}{k} \binom{n-1}{k-1}.
\end{aligned}$$

Es gilt also

$$\binom{n}{k} = \frac{n}{k} \binom{n-1}{k-1}, \qquad k \neq 0. \tag{6.5}$$

Diese Gleichung wird als *Absorbtions-* oder *Extraktionsregel* bezeichnet, je nachdem in welche Richtung man sie anwendet. Nach dem Herausziehen des Bruchs $\frac{n}{k}$ bleibt also ein Binomialkoeffizient als Faktor übrig. Das liegt daran, dass wir durch das Herausziehen von k den Nenner auf $(k-1)!$ verringert haben, und nach dem Herausziehen von n immer noch $k-1$ aufeinanderfolgende Zahlen im Zähler stehen. Die Zahl n ist die größte Zahl im Zähler. Wir müssten also eine analoge Gleichung erhalten, wenn wir stattdessen die kleinste Zahl im Zähler herausziehen, also $n-k+1$, da dann ebenfalls noch $k-1$ aufeinanderfolgende Zahlen im Zähler stehen:

$$\begin{aligned}
\binom{n}{k} &= \frac{n^{\underline{k}}}{k!} \\
&= \frac{n \cdot \ldots \cdot (n-k+2)\,(n-k+1)}{k\,(k-1)!} \\
&= \frac{n-k+1}{k} \cdot \frac{n \cdot \ldots \cdot (n-k+2)}{(k-1)!} \\
&= \frac{n-k+1}{k} \cdot \frac{n^{\underline{k-1}}}{(k-1)!} \\
&= \frac{n-k+1}{k} \binom{n}{k-1}.
\end{aligned}$$

Es gilt also eine zweite Absorbtionsregel

$$\binom{n}{k} = \frac{n-k+1}{k} \binom{n}{k-1}, \qquad k \neq 0. \tag{6.6}$$

Symmetrie. Sehr auffällig im Pascalschen Dreieck ist die Symmetrie jeder Zeile bzgl. ihrer Mitte. Dies ist die Symmetrieeigenschaft (6.1), die wir bereits oben entdeckt hatten:

$$\binom{n}{k} = \binom{n}{n-k}, \qquad n \geq 0 \text{ ganzzahlig}. \tag{6.7}$$

Die kombinatorische Begründung für die Symmetrie war einfach und elegant. Aber man kann die Formel natürlich auch nachrechnen. Dabei benutzen wir die Fakultätendarstellung.

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

Die Fakultätendarstellung können wir allerdings nur für $0 \le k \le n$ anwenden. Die Symmetrie gilt aber auch für $k < 0$ und $k > n$, weil dann beide Seiten 0 sind: Ist $k < 0$, dann ist nach Definition $\binom{n}{k} = 0$. In diesem Fall ist $n - k > n$ und somit auch $\binom{n}{n-k} = 0$. Das Gleiche gilt für $k > n$, da dann $n - k < 0$ ist.

Summenformel. Eine weitere fundamentale Eigenschaft im Pascalschen Dreieck ist die *Summenformel*: nimmt man zwei nebeneinanderstehende Zahlen im Dreieck dann findet man ihre Summe direkt unterhalb der rechten der beiden Zahlen. Nehmen wir beispielsweise in der 7. Zeile $\binom{7}{2} = 21$ und $\binom{7}{3} = 35$, dann findet man die Summe $21 + 35 = 56 = \binom{8}{3}$ unterhalb der 35. Die allgemeine Summenformel lautet für beliebiges n und k,

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}. \tag{6.8}$$

Mit dieser Formel lässt sich das komplette Pascalsche Dreieck auch ohne Kenntnis von Binomialkoeffizienten aufbauen. Man beginnt mit der Eins in der obersten Zeile, bei $n = 0$, (siehe Abbildung 6.3) und stelle sich links und rechts von dieser Eins jeweils ein Null vor. Die Summe der linken Null mit der Eins ergibt Eins, dies ist nach der Summenformel die linke Eins in der zweiten Zeile. Die Summe der Eins mit der rechten Null ergibt ebenfalls Eins, dies ist nach der Summenformel die rechte Eins in der zweiten Zeile. Damit haben wir also die zweite Zeile im Pascalschen Dreieck erzeugt. Wir setzen wieder jeweils eine Null an die beiden Enden der zweiten Zeile und verfahren analog. Dann ergibt sich die dritte Zeile, und so weiter. Die Korrektheit der Summenformel lässt sich einfach nachrechnen:

$$\begin{aligned}
\binom{n-1}{k-1} + \binom{n-1}{k} &= \binom{n-1}{k-1} + \frac{n-k}{k}\binom{n-1}{k-1} && \text{nach Gleichung (6.6)}\\
&= \binom{n-1}{k-1}\left(1 + \frac{n-k}{k}\right) && \text{ausklammern}\\
&= \binom{n-1}{k-1} \cdot \frac{n}{k} && \text{Bruch zusammenfassen}\\
&= \binom{n}{k} && \text{nach Gleichung (6.5).}
\end{aligned}$$

Diese Rechnung ist aber nur für $k \neq 0$ gültig, da wir dabei mit Brüchen hantieren, bei denen k im Nenner steht. Entsprechend gelten ja auch die beiden Gleichungen (6.5) und (6.6) nur für $k \neq 0$. Die Summenformel (6.8) ist aber auch für $k = 0$ korrekt. Das können wir einfach direkt überprüfen: Auf der linken Seite steht dann $\binom{n}{0} = 1$, und auf der rechten Seite $\binom{n-1}{-1} = 0$, da wir die Binomialkoeffizienten für $k < 0$ als 0 definiert haben, und $\binom{n-1}{0} = 1$.

Einen eleganteren Beweis der Summenformel erhält man durch folgendes kombinatorisches Argument: Es gibt $\binom{n}{k}$ Möglichkeiten k aus n Bällen zu ziehen. Wir teilen die Ergebnisse in zwei disjunkte Gruppen ein. Man sagt auch: wir *partitionieren* die Ergebnisse.

- Die eine Gruppe enthält alle Ergebnisse bei denen der Ball mit der Nummer n *nicht* gezogen wurde. Das entspricht einem Experiment bei dem k-mal gezogen wird, aber nur $n-1$ Bälle in der Urne sind, da Ball Nummer n nicht dabei ist. D. h. es gibt $\binom{n-1}{k}$ Ergebnisse in dieser Gruppe.
- In der anderen Gruppe sind alle Ergebnisse *mit* Ball Nummer n. Das entspricht einem Experiment, bei dem der Ball mit Nummer n bereits vorab zum Ergebnis gelegt wird, und dann noch $(k-1)$-mal aus $n-1$ Bällen gezogen wird. Folglich gibt es $\binom{n-1}{k-1}$ Ergebnisse in dieser Gruppe.

Die Summe der Möglichkeiten aus den beiden Gruppen muss die Gesamtzahl der Möglichkeiten ergeben, also $\binom{n}{k}$. Das ist genau die Summenformel.

Die vielleicht anschaulichste Begründung für die Summenformel erhält man über die kürzesten Gitterwege. Dazu setzen wir zwei Zwischenpunkte t_0, t_1 unmittelbar vor den Zielpunkt t, so wie es in Abbildung 6.4 gezeigt ist.

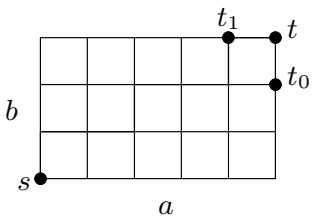


Abbildung 6.4: Jeder kürzeste Wege von s nach t geht entweder über t_0 oder über t_1. Die Zwischenpunkte t_0 und t_1 partitionieren also die Wege nach t in zwei Gruppen.

Die kürzesten Wege von s nach t lassen sich in zwei Gruppen partitionieren: Jeder kürzeste Wege von s nach t geht entweder über t_0 oder über t_1. Ein Weg der über t_0 *und* t_1 geht ist kein kürzester Weg mehr. Der weitere Weg nach t ist jeweils eindeutig. Für die Anzahl der Wege gilt also

$$w(s,t) = w(s,t_0) + w(s,t_1)\,. \tag{6.9}$$

Nach Folgerung 6.3.4 ist $w(s,t) = \binom{a+b}{a}$. Auch $w(s,t_0)$ und $w(s,t_1)$ können wir einfach bestimmen.

- Die kürzesten Wege von s nach t_0 verlaufen innerhalb des Gitters mit den Seitenlängen a und $b-1$. Es gilt also $w(s,t_0) = \binom{a+b-1}{a}$.
- Die kürzesten Wege von s nach t_1 verlaufen innerhalb des Gitters mit den Seitenlängen $a-1$ und b. Es gilt also $w(s,t_1) = \binom{a+b-1}{a-1}$.

Eingesetzt in Gleichung (6.9) erhalten wir

$$\binom{a+b}{a} = \binom{a+b-1}{a-1} + \binom{a+b-1}{a}\,.$$

Dies ist die Summenformel. Um die Form von Gleichung (6.8) zu erhalten muss man lediglich $n = a+b$ und $k = a$ setzen.

↓ **Ende der 11. Vorlesungswoche**

Parallele und obere Summe. Wir haben die Summenformel dadurch bekommen, dass wir die Wege von s nach t mittels geeignet gewählter Zwischenpunkte partitioniert haben. Mit dieser Methode lassen sich ein Vielzahl weiterer Gleichungen sehr einfach herleiten. Wir geben einige Beispiele.

Grundlagen Mathematik | 06.06: Parallele und obere Summenformel im Pascalschen Dreieck

Statt zwei Zwischenpunkten wie oben, setzen wir $b+1$ Zwischenpunkte $t_0, t_1, \ldots, t_b$ in die vorletzte Gitterspalte, siehe Abbildung 6.5.

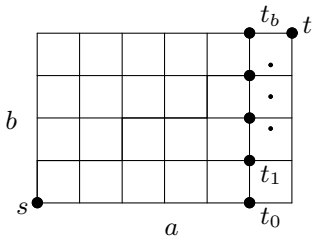


Abbildung 6.5: Jeder kürzeste Wege von s nach t geht erreicht irgendwann die rechte Spalte in der auch t liegt. Es gibt also für jeden solchen Weg genau einen Punkt t_k von dem aus ein Rechtsschritt in die rechte Spalte gemacht wird. Ab diesem Punkt ist der restliche Weg nach t eindeutig. Die Zwischenpunkte $t_0, t_1, \ldots, t_b$ partitionieren also die Wege nach t in $b+1$ Gruppen.

Ein Weg von s nach t kann jetzt über mehrere dieser Zwischenpunkte führen. Aber jeder Weg von s nach t macht von einem der Punkte t_k aus einen Rechtsschritt. Nach diesem Rechtsschritt ist der restliche Weg eindeutig, da man nur noch nach oben bis t gehen kann. Wir können also die Wege von s nach t unterteilen in die Wege, die von s bis t_k gehen und dann einen Rechtsschritt machen, für $k = 0, 1, \ldots, b$. Es gilt also

$$w(s,t) = \sum_{k=0}^{b} w(s,t_k)\,. \tag{6.10}$$

Die Punkte s und t_k definieren ein Rechteck der Größe $(a-1)\times k$. Es gilt also wieder nach Folgerung 6.3.4 und der Symmetrie (6.7):

$$w(s,t_k) = \binom{a-1+k}{k} \tag{6.11}$$

$$= \binom{a-1+k}{a-1}. \tag{6.12}$$

Wir setzen (6.11) und (6.12) in Gleichung (6.10) ein.

$$\binom{a+b}{b} = \sum_{k=0}^{b} \binom{a-1+k}{k} \tag{6.13}$$

$$= \sum_{k=0}^{b} \binom{a-1+k}{a-1}. \tag{6.14}$$

Gleichung (6.13) wird als *parallele Summe* bezeichnet, da sich im Binomialkoeffizienten bei der Summation oberer und unterer Wert gleichzeitig erhöhen. Betrachten wir ein Zahlenbeispiel mit $a = 3$ und $b = 5$ und verfolgen die Summe im Pascalschen Dreieck.

n	$\binom{n}{0}$	$\binom{n}{1}$	$\binom{n}{2}$	$\binom{n}{3}$	$\binom{n}{4}$	$\binom{n}{5}$	$\binom{n}{6}$	$\binom{n}{7}$	$\binom{n}{8}$	$\binom{n}{9}$
0	1									
1	1	1								
2	1	2	1							
3	1	3	3	1						
4	1	4	6	4	1					
5	1	5	10	10	5	1				
6	1	6	15	20	15	6	1			
7	1	7	21	35	35	21	7	1		
8	1	8	28	56	70	56	28	8	1	
9	1	9	36	84	126	126	84	36	9	1

Die Summe startet mit $\binom{a-1}{0} = \binom{2}{0} = 1$ ganz links in der Zeile für $n = 2$ im Pascalschen Dreieck. Dann wird k parallel erhöht. D. h. wir gehen im Pascalschen Dreieck vom Startpunkt $\binom{2}{0}$ die Diagonale nach rechts unten: der nächste Summand ist $\binom{3}{1} = 3$, dann geht es weiter mit $\binom{4}{2} = 6$, $\binom{5}{3} = 10$, $\binom{6}{4} = 15$. Die Summe endet bei $\binom{a-1+b}{b} = \binom{7}{5} = 21$. Die parallele Summenformel sagt, dass das Ergebnis nun direkt unterhalb von $\binom{7}{5}$ im Pascalschen Dreieck steht, nämlich $\binom{8}{5} = 56$. Wir prüfen nach: $1+3+6+10+15+21 = 56$. Wir können also zusammenfassen: wenn wir in der linken Spalte im Pascalschen Dreieck starten und die Diagonale nach rechts unten gehen, dann steht die Summe dieser Zahlen direkt unterhalb der aktuellen Zahl.

Gleichung (6.14) wird entsprechend als *obere Summe* bezeichnet, da sich im Binomialkoeffizienten bei der Summation nur der obere Wert ändert. Wir betrachten auch hier das Beispiel $a = 3$ und $b = 5$ im Pascalschen Dreieck. Startpunkt ist diesmal $\binom{a-1}{a-1} = \binom{2}{2} = 1$ ganz rechts in der Zeile für $n = 2$ im Pascalschen Dreieck. Jetzt wird k nur oben erhöht. D. h. wir bleiben in dieser Spalte und gehen senkrecht nach unten: der nächste Summand ist $\binom{3}{2} = 3$, dann geht es weiter mit $\binom{4}{2} = 6$, $\binom{5}{2} = 10$, $\binom{6}{2} = 15$. Die Summe endet bei $\binom{a-1+b}{a-1} = \binom{7}{2} = 21$. Die obere Summenformel sagt, dass das Ergebnis nun eine Zeile unterhalb und eine Zeile rechts von $\binom{7}{2}$ im Pascalschen Dreieck steht, nämlich $\binom{8}{3} = 56$. Wir bekommen also die gleichen Zahlen wie oben, $1 + 3 + 6 + 10 + 15 + 21 = 56$. Das ist natürlich kein Zufall. Parallele und oberere Summe unterscheiden sich ja nur durch die Symmetrieformel. Wir können also zusammenfassen: wenn wir in der Hauptdiagonalen im Pascalschen Dreieck starten und in dieser Spalte nach unten gehen, dann steht die Summe dieser Zahlen in der nächsten Zeile, eine Spalte rechts von der letzten Zahl.

Grundlagen Mathematik | 06.07: Vandermondesche Identität/Konvolution bei Binomialkoeffizienten

Vandermondsche Konvolution. Die Punkte t_0 und t_1 die wir zur Herleitung der Summenformel benutzt haben liegen auf einer Diagonalen. Wir verschieben die Diagonal nach links, so wie es Abbildung 6.6 zeigt.

Wir haben also wieder $b+1$ Zwischenpunkte $t_0, t_1, \ldots, t_b$. Da die Punkte auf einer Diagonalen liegen geht jeder kürzeste Weg von s nach t durch genau einen dieser Zwischenpunkte. Wenn wir bei einem t_k angekommen sind ist der weitere Weg zu t allerdings nicht mehr eindeutig. Stattdessen

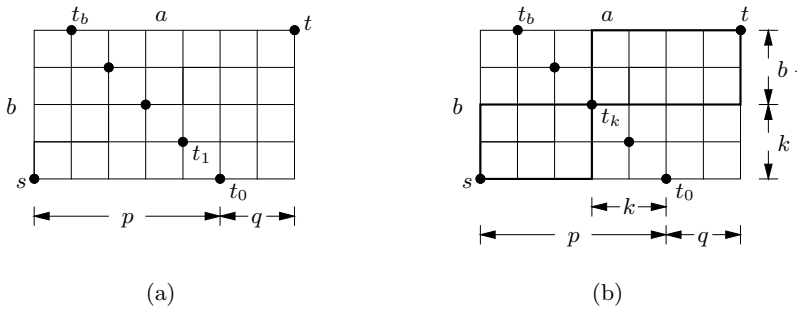


Abbildung 6.6: (a) Jeder kürzeste Wege von s nach t geht durch genau einen der Punkte $t_0, t_1, \ldots, t_b$. Die Zwischenpunkte partitionieren also die Wege nach t in $b+1$ Gruppen. (b) Ein Zwischenpunkt t_k definiert ein $(p-k) \times k$ Rechteck zwischen s und t_k, und ein $(q+k) \times (b-k)$ Rechteck zwischen t_k und b.

haben wir ein weiteres Gitter mit Startpunkt t_k und Endpunkt t. Es gibt also $w(t_k, t)$ Wege von t_k nach t. Jeder Weg von s nach t_k lässt sich mit jedem Weg von t_k nach t kombinieren, und ergibt dann einen Weg von s nach t. Die Anzahl der Wege von s nach t über den Zwischenpunkt t_k ist also $w(s, t_k)\, w(t_k, t)$. Es gilt also

$$w(s,t) \;=\; \sum_{k=0}^{b} w(s,t_k) \cdot w(t_k,t)\,. \tag{6.15}$$

Wir unterteilen die Seite a des Gitters in die Länge p von s bis t_0 und den Rest der Länge q. Es ist also $a = p + q$.

- Die kürzesten Wege von s nach t_k verlaufen innerhalb des Gitters mit der Grundseite $p - k$ und der Höhe k. Es gilt also

$$w(s,t_k) = \binom{(p-k)+k}{k} = \binom{p}{k}.$$

- Die kürzesten Wege von t_k nach t verlaufen innerhalb des Gitters mit der Grundseite $q + k$ und der Höhe $b - k$. Es gilt also

$$w(t_k,t) = \binom{(q+k)+(b-k)}{b-k} = \binom{q+b}{b-k}.$$

Eingesetzt in Gleichung (6.15) erhalten wir

$$\binom{a+b}{b} \;=\; \sum_{k=0}^{b} \binom{p}{k}\binom{q+b}{b-k}.$$

Diese Gleichung ist wenig einprägsam. Wir machen folgende Umbenennung: setze $r = p$ und $s = q + b$. Dann ist $r + s = p + q + b = a + b$ und obige Gleichung lautet dann

$$\binom{r+s}{b} \;=\; \sum_{k=0}^{b} \binom{r}{k}\binom{s}{b-k}. \tag{6.16}$$

Gleichung (6.16) geht auf Vandermonde zurück und wird als *Vandermondsche Konvolution* bezeichnet. Ein anderer Weg diese Formel zu zeigen geht über das Ziehen von Bällen. Wir ziehen b-mal aus n Bällen, die Anzahl der Ergebnisse ist also $\binom{n}{b}$. Jeder Ball sei entweder rot oder schwarz. Sagen wir, es gibt r rote Bälle und s schwarze. Es ist also $n = r + s$.

Die Ergebnisse können wir nun gemäß der Anzahl der roten Bälle auch in Gruppen partitionieren. In der Gruppe mit k roten Bällen sind $b - k$

schwarze Bälle.

- Die Anzahl der Möglichkeiten k rote Bälle zu ziehen ist $\binom{r}{k}$.
- Die Anzahl der Möglichkeiten $b-k$ schwarze Bälle zu ziehen ist $\binom{s}{b-k}$.

Zusammen sind dies $\binom{r}{k}\binom{s}{b-k}$ Möglichkeiten. Der Bereich von k ist dabei $0 \leq k \leq b$. Die Summe über die Gruppengrößen muss die Gesamtzahl $\binom{n}{b}$ ergeben. Dies führt zur Vandermondschen Konvolution (6.16).

Grundlagen Mathematik | 06.08: Binomialtheorem hergeleitet

6.6 Das Binomialtheorem

Die Binomialkoeffizienten haben ihren Namen vom *Binomialtheorem*, das Potenzen des Binoms $x+y$ ausmultipliziert, d. h. wir betrachten $(x+y)^n$, für $n \geq 0$. Für $n = 2$ erhält man beispielsweise

$$(x+y)^2 = x^2 + 2xy + y^2 \,. \tag{6.17}$$

Wir wollen untersuchen, wie man die rechte Seite der Gleichung berechnet. Das Ausmultiplizieren von $(x+y)^2 = (x+y)(x+y)$ erfolgt nach dem Distributivgesetz. Das bedeutet letztendlich, dass man jeden Summand der ersten Klammer mit jedem Summand der zweiten Klammer multipliziert.

- Nimmt man also x aus der ersten Klammer und multipliziert dies mit den Summanden der zweiten Klammer, so erhält man die Terme x^2 und xy.
- Nimmt man y aus der ersten Klammer und multipliziert dies mit den Summanden der zweiten Klammer, so erhält man die Terme yx und y^2.

Wir nehmen an, dass wir über einem Zahlenbereich arbeiten, in dem das Kommutativgesetz gilt. Dann ist $yx = xy$. Dieser Term tritt also zweimal auf. Daher kommt der *Koeffizient* 2 von xy in Gleichung (6.17).

Wir betrachten das Beispiel $n = 3$:

$$(x+y)^3 = x^3 + 3x^2y + 3xy^2 + y^3 \,. \tag{6.18}$$

Beim Ausmultiplizieren wählt man wieder aus jeder der drei Klammern im Ausdruck $(x+y)^3 = (x+y)(x+y)(x+y)$ einen der Summanden x oder y.

- Wählt man in allen drei Klammern x aus, bekommt man den Term x^3. Dies ist die einzige Möglichkeit zu x^3 zu kommen. Deshalb ist der Koeffizient 1. Wählt man stattdessen immer y, so erhält man entsprechend y^3.
- Wählen wir aus den drei Klammern genau einmal x aus, so gibt es dafür drei Möglichkeiten. Aus den restlichen zwei Klammern wählen wir dann jeweils y. Das ergibt den Term xy^2, der also auf drei Arten zustande kommt. Deswegen bekommen wir hier den Koeffizienten 3.
- Wenn wir zweimal x auswählen, und entsprechend einmal y, so ist dies symmetrisch zum vorherigen Fall, mit vertauschten Rollen von x und y. Der Term xy^2 hat also ebenfalls den Koeffizient 3.

Da wir aus jeder Klammer einen Summanden wählen muss die Summer der Hochzahlen immer gleich 3 ergeben. D.h. alle Terme haben die Form $x^k y^{3-k}$, da $k + (3 - k) = 3$. Dies gilt auch für x^3, da man $x^3 = x^3 y^0$ schreiben kann.

Im allgemeinen Fall

$$(x + y)^n = \underbrace{(x + y)(x + y) \cdots (x + y)}_{n-\text{mal}}$$

haben wir n Klammern und wählen aus jeder entweder x oder y aus. Wenn wir k-mal x auswählen und aus den restlichen $n - k$ Klammern y, erhalten wir den Term $x^k\, y^{n-k}$. Wie oft kommt dieser Term zustande? Wir nummerieren die Klammern von 1 bis n und repräsentieren sie durch Bälle. Dann ziehen wir k mal. Aus den Klammern, deren Nummer wir gezogenen haben, wählen wir x. Aus den restlichen Klammern y. Die Ziehung erfolgt ohne Zurücklegen, da wir aus jeder Klammer nur einen Summanden wählen, und ohne Reihenfolge. Es gibt also $\binom{n}{k}$ Möglichkeiten den Term $x^k\, y^{n-k}$ zu erzeugen. Damit erhalten wir das *Binomialtheorem*,

$$(x + y)^n \;=\; \sum_{k=0}^{n} \binom{n}{k} x^k\, y^{n-k}\,, \qquad n \geq 0 \text{ ganzzahlig.} \tag{6.19}$$

Wir haben oben mit dem Beispiel $n = 2$ begonnen und die Randfälle $n = 0, 1$ ausgelassen. Oft sind es gerade die Randfälle, für die ein Argument nicht mehr zutrifft. Deshalb prüfen wir die beiden Fälle nach: Für $n = 1$ steht auf der linken Seite $(x + y)^1 = x + y$. Auf der rechten Seite erhalten wir

$$\binom{1}{0} x^0\, y^{1-0} + \binom{1}{1} x^1\, y^{1-1} \;=\; 1 \cdot 1 \cdot y + 1 \cdot 1 \cdot x \;=\; x + y\,.$$

Für $n = 0$ steht auf der linken Seite $(x + y)^0 = 1$. Auf der rechten Seite erhalten wir

$$\binom{0}{0} x^0\, y^0 \;=\; 1 \cdot 1 \cdot 1 \;=\; 1\,.$$

Das Binomialtheorem gilt also in der Tat für alle $n \geq 0$.

Für x und y können wir dabei beliebige Werte einsetzen. Setzen wir zum Beispiel $x = 1$ und $y = 1$, dann ist $(x + y)^n = (1 + 1)^n = 2^n$. Damit bekommen wir die uns bereits bekannte Gleichung (6.2):

$$2^n \;=\; \sum_{k=0}^{n} \binom{n}{k} \;=\; \binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n}\,, \qquad n \geq 0 \text{ ganzzahlig.}$$

Auf der rechten Seite wir eine komplette Zeile im Pascalschen Dreieck aufsummiert. Die n-te Zeilensumme ist also 2^n.

Setzen wir $x = -1$ und $y = 1$, dann ist $(x + y)^n = (-1 + 1)^n = 0^n = 0$ für $n \geq 1$. Diesmal erhalten wir eine neue Gleichung,

$$0 \;=\; \sum_{k=0}^{n} (-1)^k \binom{n}{k} \;=\; \binom{n}{0} - \binom{n}{1} + \cdots + (-1)^n \binom{n}{n}\,, \qquad n \geq 1 \text{ ganzzahlig.}$$

Wir summieren hier ebenfalls über eine ganze Zeile im Pascalschen Dreieck, aber mit alternierendem Vorzeichen. Die n-te alternierende Zeilensumme ist also 0. Man beachte, dass wir hier $n \geq 1$ brauchen, da für $n = 0$ ist $0^0 = 1$. Die alternierende Zeilensumme ist ebenfalls 1.

Eine Gleichung die in der Wahrscheinlichkeitstheorie eine wichige Rolle spielt erhalten wir für $x = p$ und $y = 1 - p$. In der Wahrscheinlichkeitstheorie steht p für eine Wahrscheinlichkeit (engl. *probability*), also eine Zahl $0 \leq p \leq 1$. Diese Einschränkung ist für uns aber nicht wichtig. Es ist $(x + y)^n = \big(p + (1 - p)\big)^n = 1$ und damit

$$1 \;=\; \sum_{k=0}^{n} \binom{n}{k} p^k \, (1-p)^{n-k} \,, \qquad n \geq 0 \text{ ganzzahlig.} \tag{6.20}$$

Das Binomialtheorem kann man leicht auf längere Ausdrücke erweitern. Wir betrachten *Trinome*, also Ausdrücke der Form $x + y + z$. Analog zu oben nummerieren wir die n Klammern in $(x + y + z)^n$. Jetzt haben wir drei Variablen mit jeweils einer Potenz. Die Summe der Hochzahlen muss wieder n ergeben. Die Terme haben also die Form $x^k \, y^l \, z^{n-k-l}$, für $k, l \geq 0$ und $k + l \leq n$. Den Koeffizienten eines solchen Terms können wir wieder durch eine Ziehung beschreiben:

- Ziehe k-mal aus n Bällen. Aus den Klammern, deren Nummer wir gezogenen haben, wählen wir x. Dafür gibt es $\binom{n}{k}$ Möglichkeiten. Da wir nicht zurücklegen, sind jetzt noch $n - k$ Bälle übrig.
- Dann ziehen wir l-mal aus den verbleibenden $n - k$ Bällen. Aus den Klammern, deren Nummer wir gezogenen haben, wählen wir y. Dafür gibt es $\binom{n-k}{l}$ Möglichkeiten.

Aus den restlichen Klammern wählen wir z. Insgesamt ergibt dies $\binom{n}{k}\binom{n-k}{l}$ Möglichkeiten den Term $x^k \, y^l \, z^{n-k-l}$ zu erhalten. Wir formen den Ausdruck noch etwas um:

$$\begin{aligned} \binom{n}{k}\binom{n-k}{l} \;&=\; \frac{n!}{k!\,(n-k)!} \; \frac{(n-k)!}{l!\,(n-k-l)!} \\ &=\; \frac{n!}{k!\,l!\,(n-k-l)!}. \end{aligned}$$

Bei letzten Ausdruck sieht man die Analogie zur Fakultätendarstellung der Binomialkoeffizienten. Entsprechend nennt man den Ausdruck hier *Trinomialkoeffizient* und schreibt dafür

$$\binom{n}{k,l} \;=\; \frac{n!}{k!\,l!\,(n-k-l)!}.$$

Damit können wir das *Trinomialtheorem* formulieren,

$$(x + y + z)^n \;=\; \sum_{\substack{k,l \geq 0 \\ k+l \leq n}} \binom{n}{k,l} \, x^k \, y^l \, z^{n-k-l} \,, \qquad n \geq 0 \text{ ganzzahlig.} \tag{6.21}$$

Die Summe ist hierbei so zu verstehen, dass wir über alle Werte von k und l aufsummieren, die die Bedingungen unter dem Summenzeichen erfüllen. Es handelt sich also eigentlich um eine Doppelsumme, die wir

explizit auch so schreiben können:

$$(x+y+z)^n = \sum_{k=0}^{n} \sum_{l=0}^{n-k} \binom{n}{k,l} x^k y^l z^{n-k-l}.$$

Die Schreibweise in Gleichung (6.21) ist aber wohl leichter lesbar.

Übersicht. Die folgende Tabelle stellt noch einmal alle gezeigten Gleichungen zusammen.

Gleichung			Bereich	Name
$\binom{n}{k}$	$=$	$\frac{n!}{k!\,(n-k)!}$	$n \geq k \geq 0$	*Fakultäten-Darstellung*
$\binom{n}{k}$	$=$	$\binom{n}{n-k}$	$n \geq 0$	*Symmetrie*
$\binom{n}{k}$	$=$	$\frac{n}{k}\binom{n-1}{k-1}$	$k \neq 0$	*Absorbtion/Extraktion 1*
$\binom{n}{k}$	$=$	$\frac{n-k+1}{k}\binom{n}{k-1}$	$k \neq 0$	*Absorbtion/Extraktion 2*
$\binom{n}{k}$	$=$	$\binom{n-1}{k}+\binom{n-1}{k-1}$		*Summenformel*
$\binom{m+n+1}{n}$	$=$	$\sum_{k=0}^{n}\binom{m+k}{k}$		*parallele Summe*
$\binom{n+1}{m+1}$	$=$	$\sum_{k=0}^{n}\binom{k}{m}$	$m,n \geq 0$	*obere Summe*
$\binom{r+s}{n}$	$=$	$\sum_{k=0}^{n}\binom{r}{k}\binom{s}{n-k}$		*Vandermonde Konvolution*
$(x+y)^n$	$=$	$\sum_{k=0}^{n}\binom{n}{k}x^k y^{n-k}$	$n \geq 0$	*Binomial Theorem*
$(x+y+z)^n$	$=$	$\sum_{\substack{k,l\geq 0\\ k+l\leq n}}\binom{n}{k,l}x^k y^l z^{n-k-l}$	$n \geq 0$	*Trinomial Theorem*

Freiwilliges Übungsblatt 06

Bearbeiten Sie in dieser Woche das sechste freiwillige Übungsblatt, auch als Vorbereitung für die Klausur.

↓ **Ende der 12. Vorlesungswoche**

Besprechung Blatt 06

Für das Freiwillige Blatt 06 erhalten Sie wieder eine handschriftliche Musterlösung (unter den Modulen in Canvas).

Grundlagen Mathematik | 06.09: Algorithmische Berechnung Binomialkoeffizienten mit Absorptionsregel

6.7 Berechnung der Binomialkoeffizienten

Um $\binom{n}{k}$ für gegebene Werte n und k zu berechnen bietet sich die Definition an:

$$\binom{n}{k} = \frac{n^{\underline{k}}}{k!}.$$

Die Frage ist, wie man diesen Ausdruck am Besten auswertet.

Der naheliegendste Weg ist wohl, zuerst Zähler $n^{\underline{k}}$ und Nenner $k!$ zu berechnen und diese anschließend zu dividieren. Der Nachteil dieser Vorgehensweise ist, dass die Zwischenergebnisse die man für Zähler und Nenner erhält riesig sein können im Vergleich zum Endergebnis. Im Extremfall ist $k = n$. Dann würde man $n!$ berechnen und am Ende das Ergebnis $\binom{n}{n} = 1$ bekommen. Da die darstellbaren Zahlen in einem Computer begrenzt sind, könnte es also passieren, dass die Rechnung an den zu hohen Zwischenergebnissen scheitert, obwohl das Ergebnis eigentlich im darstellbaren Bereich liegt.

Um die großen Zahlen zu vermeiden, kann man $\binom{n}{k}$ als Produkt von Brüchen berechnen:

$$\binom{n}{k} = \frac{n}{k}\,\frac{n-1}{k-1}\,\cdots\,\frac{n-k+2}{2}\,\frac{n-k+1}{1}. \tag{6.22}$$

Dann sind alle Zwischenergebnisse kleiner als das Endergebnis. Allerdings wartet bereits das nächste Problem: wenn wir die Brüche $\frac{n-i}{k-i}$ für $i = 0, 1 \ldots, k-1$ berechnen, und dann multiplizieren erhalten wir als Zwischenergebnisse gebrochene Zahlen, was bei endlicher Zahlendarstellung zu Rundungsfehlern führen kann. Da $\binom{n}{k}$ eine ganze Zahl ist, wäre es natürlich gut, wenn unsere Rechnung ebenfalls ganzzahlig bleibt.

Das gelingt dadurch, dass man die Brüche in Gleichung (6.22) von rechts her aufmultipliziert. Das liegt an der Extraktionsregel (6.5):

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

Hat man $\binom{n-1}{k-1}$ bereits berechnet und multipliziert dies mit n, so kann man anschließend durch k dividieren und erhält als Ergebnis wieder eine ganze Zahl, nämlich $\binom{n}{k}$.

Folgender Algorithmus Binom(n, k) berechnet nach dieser Methode $\binom{n}{k}$. Als Optimierung wird dabei die Symmetrie ausgenützt: ist $k > n/2$ dann wird k durch $n - k$ ersetzt.

Binom(n, k)

```
1  if 2k > n then k ← n − k
2  b ← 1
3  for i ← 1 to k do
4      b ← b · (n − k + i)
5      b ← b/i
6  return b
```

Algorithmus Binom(n, k) berechnet $\binom{n}{k}$ mit k Multiplikationen und k Divisionen. Zwar sind nun alle Zwischenergebnisse ganzzahlig, aber nicht

mehr kleiner als das Endergebnis: zum Beispiel ist die letzte Operation in Binom(n,k) die Division durch k. Folglich ist der Wert von b vor der Division um den Faktor k größer als das Endergebnis. Im Vergleich zum ersten Ansatz ist das aber natürlich eine drastische Verbesserung.

Die Summenformel (6.8) liefert eine Methode um $\binom{n}{k}$ ganzzahlig berechnen und alle Zwischenergebnisse klein zu halten. Die Idee ist einfach, das Pascalsche Dreieck bis zur Position $\binom{n}{k}$ aufzubauen. Dazu muss man das Dreieck nicht vollständig bis zur n-ten Zeile aufbauen. Es genügt den rautenförmigen Ausschnitt mit der Summenformel zu berechnen, so dass die untere Ecke bei $\binom{n}{k}$ liegt (siehe Abbildung 6.7).

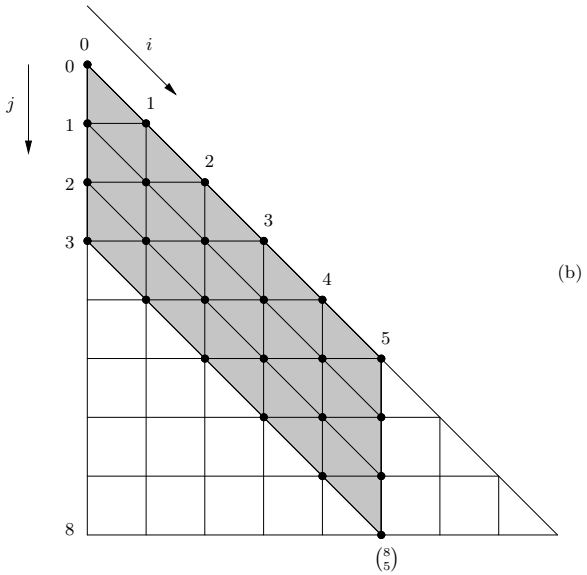


Abbildung 6.7: Um $\binom{n}{k} = \binom{8}{5}$ zu berechnen, werden alle Binomialkoeffizienten im grau gefärbten Bereich mit der Summenformel berechnet.

Grundlagen Mathematik | 06.10: Algorithmische Berechnung Binomialkoeffizienten mit Summenformel

Algorithmus Binom-Additiv(n,k) beschreibt die Methode detailliert.

Binom-Additiv(n,k)

```
1  if 2k > n then k ← n − k
2  for i ← 0 to k do b[i] ← 1
3  for j ← 1 to n − k do
4      for i ← 1 to k do b[i] ← b[i − 1] + b[i]
5  return b[k]
```

In Zeile 1 wird $b[i] = 1 = \binom{i}{i}$ mit den Diagonalwerten des Pascalschen Dreiecks initialisiert. Die for-Schleife in Zeile 2 berechnet die Binomialkoeffizienten auf den Parallelen zur Diagonalen im Abstand $j = 1, 2, \ldots, n-k$. In der for-Schleife in Zeile 3 wird $b[i]$ der Wert $\binom{j+i}{i} = \binom{j+i-1}{i-1} + \binom{j+i-1}{i}$ zugewiesen. Bei der letzten Zuweisung ist $j = n-k$ und folglich ist am Ende $b[k] = \binom{j+k}{k} = \binom{n}{k}$. Insgesamt haben wir dabei $k(n-k)$ Binomialkoeffizienten berechnet und ein array der Länge $k \leq n/2$ benutzt.

Grundlagen Mathematik | 06.11: Abschätzungen von Binomialkoeffizienten - Allgemeiner Teil

6.8 Abschätzungen der Binomialkoeffizienten

Maximalität der Binomialkoeffizienten. Im Pascalschen Dreieck kann man beobachten, dass die Binomialkoeffizienten bis zur Mitte immer größer werden, und dann symmetrisch in gleicher Weise wieder kleiner. Diese Eigenschaft lässt sich einfach nachrechnen. Dazu untersuchen wir für welche k die Binomialkoeffizienten wachsen, also wann $\binom{n}{k-1} < \binom{n}{k}$ gilt. Die zweite Absorptionsregel (6.6) sagt $\binom{n}{k} = \frac{n-k+1}{k}\binom{n}{k-1}$. Folglich gilt

$$\begin{aligned}\binom{n}{k-1} < \binom{n}{k} &\iff \binom{n}{k-1} < \frac{n-k+1}{k}\binom{n}{k-1}\\ &\iff 1 < \frac{n-k+1}{k}\\ &\iff k < \frac{n+1}{2}.\end{aligned}$$

Die letzte Ungleichung bestätigt das monotone Wachstum bis zur Mitte. Folglich sind die Koeffizienten in der Mitte am größten, also $\binom{n}{n/2}$ für gerades n und $\binom{n}{(n-1)/2}$ für ungerades n.

Lemma 6.8.1. *Der Binomialkoeffizient $\binom{n}{k}$ ist maximal für $k = \lfloor n/2 \rfloor$.*

Eine erste untere und obere Schranke. Arithmetische Ausdrücke möchte man meistens so einfach wie möglich darstellen. Wenn Binomialkoeffizienten in einem Ausdruck vorkommen ist das in der Regel eher hinderlich. Eine Möglichkeit, Binomialkoeffizienten loszuwerden, besteht darin, sie möglichst genau nach oben und unten durch einfache Ausdrücke abzuschätzen und dann mit einer Abschätzung anstatt dem genauen Wert weiter zu rechnen. Für viele praktischen Anwendungen ist das völlig ausreichend. Hier ist ein Beispiel für eine einfache Abschätzung:

$$\left(\frac{n}{k}\right)^k \le \binom{n}{k} \le \left(\frac{n}{\sqrt{k}}\right)^k, \qquad 1 \le k \le n. \tag{6.23}$$

Untere und obere Schranke in (6.23) sind Brüche mit Potenzen. Mit diesen Ausdrücken rechnet es sich natürlich einfacher als mit den fallenden Faktoriellen im Binomialkoeffizienten.

Wir zeigen als erstes die untere Schranke in (6.23). Dazu schreiben wir $\binom{n}{k}$ als Produkt von k Brüchen:

$$\begin{aligned}\binom{n}{k} &= \frac{n(n-1)\cdots(n-k+1)}{k!}\\ &= \frac{n}{k}\cdot\frac{n-1}{k-1}\cdot\ldots\cdot\frac{n-k+2}{2}\cdot\frac{n-k+1}{1}\\ &= \prod_{j=0}^{k-1}\frac{n-j}{k-j}.\end{aligned}$$

D. h. ausgehend von $\frac{n}{k}$ werden Zähler und Nenner gleichermaßen verringert. Der Ausdruck $(\frac{n}{k})^k$ auf der linken Seite in Ungleichung (6.23) ist dagegen das k-fache Produkt von immer dem gleichen Bruch, $\frac{n}{k}$. Wir

zeigern, dass

$$\frac{n}{k} \leq \frac{n-j}{k-j} \text{ für alle } j = 0, 1, \ldots, k-1.$$

Daraus folgt dann die erste Ungleichung.

Für $j = 0$ ist die Behauptung trivial. Für $j \geq 1$ gilt

$$\begin{aligned} \frac{n}{k} \;\leq\; \frac{n-j}{k-j} \quad &\Longleftrightarrow \quad (k-j)\,n \;\leq\; k\,(n-j) \\ &\Longleftrightarrow \quad j\,k \;\leq\; j\,n \\ &\Longleftrightarrow \quad k \;\leq\; n. \end{aligned}$$

Die letzte Ungleichung ist nach Voraussetzung richtig. Damit ist die untere Schranke gezeigt.

Betrachten wir die obere Schranke $\frac{n^k}{k!} \leq (\frac{n}{\sqrt{k}})^k$. Der Zähler auf der linken Seite ist kleiner gleich dem rechten Zähler, $n^{\underline{k}} \leq n^k$ Wir zeigen, dass auch $\frac{1}{k!} \leq (\frac{1}{\sqrt{k}})^k$ gilt. Daraus folgt dann die Ungleichung. Es gilt:

$$\begin{aligned} \frac{1}{k!} \;\leq\; \left(\frac{1}{\sqrt{k}}\right)^k \quad &\Longleftrightarrow \quad \sqrt{k}^k \;\leq\; k! && (6.24) \\ &\Longleftrightarrow \quad k^k \;\leq\; (k!)^2. && (6.25) \end{aligned}$$

Im letzten Schritt haben wir beide Seiten quadriert. Wir schreiben $(k!)^2$ etwas um:

$$\begin{aligned} (k!)^2 \;&=\; \begin{array}{ccccccccc} 1 & \cdot & 2 & \cdots & j & \cdots & (k-1) & \cdot & k \\ k & \cdot & (k-1) & \cdots & (k-j+1) & \cdots & 2 & \cdot & 1 \end{array} \\ &=\; \prod_{j=1}^{k} j\,(k-j+1) \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad . \end{aligned}$$

In der letzten Formel haben wir jeweils die zwei übereinander stehenden Faktoren in den zwei Zeilen davor zusammengefasst. Um Ungleichung (6.25) zu zeigen, genügt es, wenn jeder der k Faktoren $j\,(k-j+1)$ größer oder gleich k ist, für $j = 1, \ldots, k$. Es ist

$$j\,(k-j+1) \geq k \quad \Longleftrightarrow \quad j\,(k-j+1) - k \geq 0.$$

Wir schreiben den letzten Ausdruck um:

$$j\,(k-j+1) - k \;=\; -j^2 + (k+1)\,j - k \;=\; -(j-1)(j-k).$$

Der Ausdruck $-(j-1)(j-k)$ ist ein quadratisches Polynom in j mit den Nullstellen $j = 1$ und $j = k$. Für $1 < j < k$ ist $-(j-1)(j-k) > 0$. Das war zu zeigen.

Eine bessere obere Schranke mit analytischen Methoden. Mit Methoden aus der Analysis erhalten wir eine bessere obere Schranke. Dort wird die Reihenentwicklung der Exponentialfunktion gezeigt. Sei e die *eulersche Zahl*, $\mathrm{e} = \lim_{n\to\infty} \left(1 + \frac{1}{n}\right)^n \approx 2{,}718\cdots$. Für alle $x \in \mathbb{R}$ gilt

dann

$$e^x = \sum_{j=1}^{\infty} \frac{x^j}{j!}.$$

Für $x \geq 0$ sind alle Summanden in der Summe nicht-negativ. Die Summe ist also sicher größer als jeder ihrer Summanden. Für $x = k$ ergibt sich

$$e^k = \sum_{j=1}^{\infty} \frac{k^j}{j!} \geq \frac{k^k}{k!}.$$

In der letzten Abschätzungen haben wir den Summanden für $j = k$ aus der Summe herausgepickt. Folglich gilt

$$\frac{1}{k!} \leq \frac{e^k}{k^k},$$

und damit auch

$$\frac{n^{\underline{k}}}{k!} \leq \frac{n^k e^k}{k^k}.$$

Damit erhalten wir die oberer Schranke

$$\binom{n}{k} \leq \left(\frac{en}{k}\right)^k. \tag{6.26}$$

Diese Schranke ist für $k \geq 8$ besser als die obere Schranke in (6.23):

$$\begin{aligned} \left(\frac{en}{k}\right)^k \leq \left(\frac{n}{\sqrt{k}}\right)^k &\iff \frac{en}{k} \leq \frac{n}{\sqrt{k}} \\ &\iff \frac{e}{k} \leq \frac{1}{\sqrt{k}} \\ &\iff e^2 \leq k \\ &\iff 8 \leq k \end{aligned}$$

Die letzte Ungleichung folgt, da $e^2 \approx 7,389\cdots$ und k ganzzahlig ist.

Grundlagen Mathematik | 06.12: Abschätzungen von Binomialkoeffizienten - Bruchteile von n

Bruchteile von n. Die Abschätzungen in (6.23) und (6.26) gelten für jedes k im Bereich $1 \leq k \leq n$. In vielen Anwendungen ist k aber nicht im Randbereich, sondern ein Anteil von n, etwa $k = n/2$ oder $k = n/3$. Betrachten wir den größten Binomialkoeffizienten, also $k = n/2$. Aus Ungleichung (6.23) und (6.26) erhalten wir folgende Schranken:

$$\left(\frac{n}{n/2}\right)^{n/2} = 2^{n/2} \leq \binom{n}{n/2} \leq \left(\frac{en}{n/2}\right)^{n/2} = (2e)^{n/2}.$$

Da $e > 2$, ist die obere Schranke $> 2^n$. Wir können also festhalten, dass die Binomialkoeffizienten zur Mitte hin exponentiell groß sind. Dies ist für viele Anwendungen bereits gut genug.

Benötigt man aber möglichst genaue Approximationen, dann sind unsere Schranken nicht besonders gut. Bereits eine (fast triviale) Überlegung liefert bessere Schranken für $k = n/2$. Dazu betrachten wir noch einmal

die Summe aller Binomialkoeffizient,

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

- Da alle Summanden positiv sind müssen folglich die einzelnen Koeffizienten $\leq 2^n$ sein.
- Die Summe besteht aus $n+1$ Summanden. Der Mittelwert der Summanden ist also $2^n/(n+1)$. Es muss folglich mindestens einen Summanden geben, der diesen Mittelwert erreicht. Da der Koeffizient in der Mitte der Größte ist, muss dies zumindest für diesen Koeffizienten zutreffen.

Damit erhalten für $\binom{n}{n/2}$ die verbesserte Abschätzung

$$\frac{2^n}{n+1} \;\leq\; \binom{n}{n/2} \;\leq\; 2^n \,. \tag{6.27}$$

Diese Schranken sind schon sehr präzise: Untere und obere Schranke unterscheiden sich nur noch um den Faktor $n+1$. Wenn man bedenkt, dass $\binom{n}{n/2}$ exponentiell groß ist, kann man einen vergleichsweise so kleinen Faktor an Ungenauigkeit in vielen Anwendungen vernachlässigen, z. B. wenn es darum geht Rechenzeiten von Algorithmen abzuschätzen.

In der Literatur findet man noch bessere Abschätzungen. Mit Hilfe der *Stirlingschen Formel* kann man zeigen, dass die Größenordnung von $\binom{n}{n/2}$ ziemlich genau durch den Ausdruck $\frac{2^n}{\sqrt{n}}$ beschrieben wird.

Abschätzung (6.27) kann man auf beliebige Bruchteile von n verallgemeinern. Sei $0 < p < 1$. Wir wollen $\binom{n}{pn}$ abschätzen. Folgendes Lemma erweitert Lemma 6.8.1.

Lemma 6.8.2. *Sei $0 < p < 1$. Dann ist $s_k := \binom{n}{k} p^k (1-p)^{n-k}$ maximal für $k = \lfloor pn \rfloor$.*

Um dies zu zeigen, untersuchen wir für welche k die Folge $(s_k)_{k\in\mathbb{N}}$ monoton wachsend ist:

$$\begin{aligned}
s_{k-1} < s_k \quad &\Longleftrightarrow \quad \binom{n}{k-1} p^{k-1} (1-p)^{n-(k-1)} \;<\; \binom{n}{k} p^k (1-p)^{n-k} \\
&\Longleftrightarrow \quad 1-p \;<\; \frac{n-k+1}{k}\, p \\
&\Longleftrightarrow \quad k(1-p) \;<\; (n-k+1)\, p \\
&\Longleftrightarrow \quad k \;<\; (n+1)p \,.
\end{aligned}$$

Folglich wächst s_k für $k = 0, 1, \ldots, \lfloor pn \rfloor$.

In unserer Abschätzung für $\binom{n}{pn}$ benützten wir die *(binäre) Entropiefunktion*. Das ist die Funktion $h : [0,1] \to [0,1]$ definiert durch $h(0) = h(1) = 0$, und für $0 < p < 1$ durch

$$h(p) = -\big(p \log p + (1-p)\log(1-p)\big).$$

Die Funktionswerte der Entropiefunktion liegen zwischen 0 und 1, d. h. es gilt $0 \leq h(p) \leq 1$ für alle $p \in [0,1]$. Der Hochpunkt liegt bei $p = \frac{1}{2}$, an dieser Stelle gilt $h(\frac{1}{2}) = 1$. Anhand der Definition sieht man sofort,

dass h symmetrisch zum Hochpunkt ist, $h(p) = h(1-p)$. Zum Beispiel ist $h(\frac{1}{4}) = h(\frac{3}{4}) \approx 0,8113\ldots$.

Wir benützen die Entropiefunktion im Exponenten. Es gilt

$$2^{h(p)} = \frac{1}{p^p\,(1-p)^{1-p}}.$$

Sei also $0 < p < 1$. Um uns die eckigen Klammern zum abrunden von pn zu ersparen nehmen wir der Einfachheit halber an, dass pn bereits ganzzahlig ist. Dann gilt folgende Verallgemeinerung von Abschätzung (6.27):

$$\frac{2^{nh(p)}}{n+1} \;\leq\; \binom{n}{np} \;\leq\; 2^{nh(p)}. \tag{6.28}$$

Die Behauptung ist trivial für $p = 0$ und $p = 1$. Sei also $0 < p < 1$.

Wir zeigen zuerst die obere Schranke. Dazu betrachten wir noch einmal Gleichung (6.20)

$$\sum_{k=0}^{n} \binom{n}{k} p^k\,(1-p)^{n-k} \;=\; 1.$$

Da alle Summanden in der Summe größer als 0 sind, müssen sie auch kleiner als 1 sein. Dies gilt auch für den Summanden mit $k = pn$, also

$$\binom{n}{np} p^{np}\,(1-p)^{n-np} \;\leq\; 1$$

Daraus erhalten wir

$$\begin{aligned}
\binom{n}{np} &\leq \frac{1}{p^{np}\,(1-p)^{n-np}} \\
&= \left(\frac{1}{p^p\,(1-p)^{1-p}}\right)^n \\
&= 2^{nh(p)}.
\end{aligned}$$

Für die untere Schranke betrachten wir ebenfalls Gleichung (6.20). Wir schätzen die Summe mit Hilfe des größten Summanden nach oben ab. Nach Lemma 6.8.2 ist dies $\binom{n}{np} p^{np}\,(1-p)^{n-np}$. Da die Summe $n+1$ Summanden hat bekommen wir

$$\begin{aligned}
1 &= \sum_{k=0}^{n} \binom{n}{k} p^k\,(1-p)^{n-k} \\
&\leq (n+1)\binom{n}{np} p^{np}\,(1-p)^{n-np}.
\end{aligned}$$

Daraus erhalten wir analog wie bei der oberen Schranke

$$\begin{aligned}
\binom{n}{np} &\geq \frac{1}{n+1}\,\frac{1}{p^{np}\,(1-p)^{n-np}} \\
&= \frac{1}{n+1}\left(\frac{1}{p^p\,(1-p)^{1-p}}\right)^n \\
&= \frac{1}{n+1}\,2^{nh(p)}.
\end{aligned}$$

6.9 Übungen

5. Zeigen Sie das Binomialtheorem (6.19) durch vollständige Induktion über n.

Eine Lösung finden Sie unter `https://de.wikibooks.org/wiki/Beweisarchiv:_Algebra:_Ringe:_Binomischer_Lehrsatz`.

↓ Ende der 13. Vorlesungswoche