Skript zur Vorlesung Mathematik für Informatiker I

Logik und Diskrete Strukturen

Autoren: Frank Hoffmann und Klaus Kriegel

Wintersemester 2015/16
Institut für Informatik, Freie Universität Berlin

Dozent: F. Hoffmann

(Stand 01.10.2015)

Hinweise an die Hörer/Leser:

Dieses Skript ist nur für den internen Gebrauch am Fachbereich Mathematik und Informatik der Freien Universität Berlin entstanden und es ist speziell zugeschnitten auf die Erstsemestervorlesung "Logik und Diskrete Mathematik". An einigen Stellen werden Themen angeschnitten, die über diese Vorlesung hinausgehen und Bezüge etwa zum Vorlesungszyklus "Algorithmen und Programmierun" herstellen. Diese sind im Inhaltsverzeichnis mit einem Stern * markiert. Für das Skript wurde unter anderem auch Material aus den folgenden Quellen benutzt:

- Meinel, Mundhenk; Mathematische Grundlagen der Informatik, Teubner
- Rosen; Disrete Mathematics and its applications, McGraw-Hill
- Aigner; Diskrete Mathematik, Vieweg
- Lehman, Leighton; Discrete Math for CS, MIT-Skript

Das Korrekturlesen der Vorgängerversion des Skripts hatte dankenswerterweise Romain Grunert übernommen. Hinweise auf die sicherlich noch vorhandenen weiteren Fehler bitte an
`hoffmann@inf.fu-berlin.de`.

Auf Grund meiner langjährigen Erfahrung mit Mathematikveranstaltungen für Informatik-Studenten an dieser Stelle noch ein eindringlicher Rat:
Lassen Sie sich durch die Existenz dieses weitestgehend vollständigen Skripts, was aber weit davon entfernt ist ein Lehrbuch zu sein, nicht verführen!
Der Besuch der Vorlesungen/Tutorien ist für fast alle Teilnehmer die effektivste Art, sich mit dem Stoff vertraut zu machen, das Nacharbeiten zu Hause kommt natürlich hinzu. Tun Sie es in Ihrem eigenen Interesse, besuchen Sie regelmäßig die Veranstaltungen, betreiben Sie das Nacharbeiten der Vorlesungen und das Lösen der Übungsaufgaben kontinuierlich und mit dem für Sie individuell notwendigen zeitlichen Aufwand! Dann kann ich garantieren, dass es sogar Spaß machen wird und auch der Erfolg wird sich letztlich einstellen.

Oktober 2015, Frank Hoffmann.

Inhaltsverzeichnis

1 Boolesche Aussagenlogik

1.1 Grundbegriffe; Vom Booleschen Term zur Booleschen Funktion

Die klassische Boolesche Aussagenlogik (George Boole, engl. Mathematiker 1815 - 1864) beruht auf zwei Grundprinzipien, dem *Zweiwertigkeitsprinzip*, welches fordert, dass jede Aussage einen eindeutig bestimmten Wahrheitswert hat, der nur *wahr* oder *falsch* sein kann, und dem *Extensionalitätsprinzip*, nach dem der Wahrheitswert einer zusammengesetzten Aussage nur von den Wahrheitswerten ihrer Bestandteile abhängt.
Wir werden im folgenden eine 1 für den Wahrheitswert *wahr* und eine 0 für *falsch* verwenden.
Zunächst sind folgende Fragen zu klären:

- Was sind Aussagen?
- Nach welchen Regeln kann man Aussagen zusammensetzen? Das ist die Syntax der Aussagenlogik.
- Nach welchen Regeln werden zusammengesetzte Aussagen interpretiert, das ist die Frage nach der Semantik.

Auf die alten Griechen, genauer Aristoteles (384 - 322 v.Chr.), geht die folgende Definition zurück:

Definition: Eine Aussage ist ein Satz (ein formalsprachliches Gebilde), von dem es sinnvoll ist zu sagen, er sei entweder wahr oder falsch.

Beispiele: Hier einige Beispiele aus der Mathematik.

1. Der Satz *"7 ist eine Primzahl."* und der Satz *"7 ist eine ungerade Zahl."* sind wahre Aussagen. Dagegen ist der Satz *"7 ist eine gerade Zahl."* eine falsche Aussage. Genauer gesehen ist der letzte Satz die Negation des zweiten Satzes, denn *nicht ungerade* zu sein, ist (zumindest für ganze Zahlen) das gleiche, wie *gerade* zu sein.
2. *" Jede natürliche Zahl* > 1 *ist das Produkt von Primzahlen."* ist eine wahre Aussage. Das ist die bekannte Primzahlzerlegung von natürlichen Zahlen, und wenn die Zahl selbst Primzahl ist, so besteht das Produkt nur aus einem Faktor, nämlich der Zahl selbst.
3. Der Satz *"*$\sqrt{2}$ *ist eine rationale Zahl."* ist – wie man aus der Schulmathematik weiß – eine falsche Aussage, aber es bedarf schon einiger Überlegungen, um das zu zeigen.
4. Der Satz *"Jede gerade natürliche Zahl größer als* 2 *ist die Summe zweier Primzahlen"* ist eine Aussage, denn entweder es gibt eine gerade Zahl, die sich nicht als Summe zweier Primzahlen darstellen lässt (dann ist die Aussage falsch), oder es gibt keine solche Zahl (dann ist die Aussage wahr). Man nimmt an, dass die Aussage wahr ist (Goldbach–Vermutung), konnte das aber bisher noch nicht beweisen.

5. Der Satz *"Dieser Satz ist falsch."* ist als Russells Paradoxon bekannt. Durch die spezielle Art des Bezugs auf sich selbst ist er weder wahr noch falsch (beides führt zum Widerspruch) und ist deshalb **keine** Aussage. Genauso hat ein Satz wie *"Sei ruhig!"* keinen zugeordneten Wahrheitswert.

Die Zuordnung des Wahrheitswertes zu Einzelaussagen ist Sache der Einzelwissenschaft (Mathematik, Biologie, Physik etc.). Die Logik interessiert sich für den Wahrheitswert zusammengesetzter Aussagen.

Wie kann man Aussagen zu neuen Aussagen zusammensetzen?:
Das Zusammensetzen von Aussagen erfolgt durch die Verwendung von logischen Verknüpfungswörtern wie:

und, oder, nicht, wenn ... dann, genau dann wenn, entweder ... oder

Dafür gibt es Operationssymbole und -namen, hier die zunächst wichtigen:

- die **Negation** einer Aussage A, Notation: $\neg A$, gesprochen: "nicht A"
- die **Konjunktion** von A und B, Notation: $A \wedge B$,gesprochen: " A und B ",
- die **Disjunktion** von A und B, Notation $A \vee B$, gesprochen: " A oder B ",
- die **Implikation**, Notation: $A \Rightarrow B$, gesprochen: " A impliziert B" ,
- die **Äquivalenz**, Notation: $A \Leftrightarrow B$, gesprochen: " A genau dann, wenn B",
- die **Antivalenz**, Notation: $A \oplus B$, gesprochen: " entweder A oder B "

Es folgt die induktive Definition der Syntax der Aussagenlogik, also die Antwort auf die Frage, welche Konstrukte wir zulassen wollen. Sei dazu A eine Menge von Einzelaussagen, deren Wahrheitswert schon feststeht. Dazu soll immer wenigstens die wahre Aussage *"true"* und die falsche Aussage *"false"* gehören. Weiterhin sei V eine abzählbare Menge von Variablen, also einfach eine Menge von Platzhaltern für Aussagen.

Definition:(Syntax der Aussagenlogik)
Die Menge der *Booleschen Formeln (Booleschen Terme)* der Aussagenlogik über der Variablenmenge V und der Aussagenmenge A ist induktiv definiert:

1. Jedes $a \in A$ und jedes $v \in V$ sind Boolescher Term.
2. Wenn t Boolescher Term ist, so ist auch $(\neg t)$ Boolescher Term.
3. Wenn t_1 und t_2 Boolesche Terme sind, so sind es auch die Konjunktion $(t_1 \wedge t_2)$ und die Disjunktion $(t_1 \vee t_2)$.
4. (Minimalitätsprinzip) Nur Konstrukte, die sich durch endlich oftes Anwenden der Regeln (1), (2) bzw. (3) erzeugen lassen, sind Boolesche Terme.

Bemerkungen:

1. Man beachte, dass diesen Konstrukten bisher keinerlei Interpretation bezüglich des Wahrheitswertes zugewiesen wurde.

2. Die runden Klammern sind wichtig! Nur durch sie kann ggf. nachvollzogen werden, wie ein Term entstanden ist.

3. Wir haben nicht alle oben eingeführten Operatoren benutzt! Der Grund wird später klar, tatsächlich könnte man es ohne weiteres tun und wir werden die anderen auch verwenden. Die Menge $\{\wedge, \vee, \neg\}$ von Operatoren heißt *Standardsignatur* für Boolesche Formeln.

Boolesche Algebra, Rechnen mit Wahrheitswerten:

Sei $\mathbb{B} = \{0,1\}$ die Menge der beiden Booleschen Wahrheitswerte. Mit $\mathbb{B} \times \mathbb{B}$ bezeichnen wir die Menge der geordneten Paare von Wahrheitswerten (das sogenannte kartesische Produkt von $\mathbb{B}$ und $\mathbb{B}$), das heißt $\mathbb{B} \times \mathbb{B} = \{(0,0),(0,1),(1,0),(1,1)\}$.
Wir definieren die Negation $\neg p$ für einen Wahrheitswert $p \in \mathbb{B}$ durch: $\neg 0 = 1$ und $\neg 1 = 0$.
Die Negation kann also aufgefasst werden als Funktion $\neg : \mathbb{B} \to \mathbb{B}$.
Des weiteren betrachten wir die folgenden zweistelligen Booleschen Funktionen von $\mathbb{B} \times \mathbb{B}$ nach $\mathbb{B}$ gegeben durch ihren Werteverlauf, die Konjunktion, Disjunktion, Implikation, Äquivalenz und Antivalenz von zwei Wahrheitswerten festlegen.

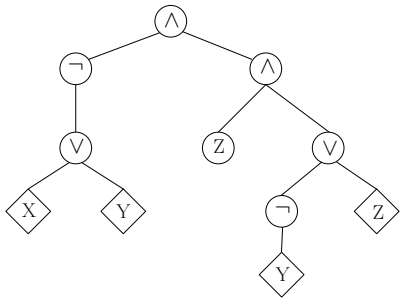
p	q	$p \wedge q$	$p \vee q$	$p \Rightarrow q$	$p \Leftrightarrow q$	$p \oplus q$
0	0	0	0	1	1	0
0	1	0	1	1	0	1
1	0	0	1	0	0	1
1	1	1	1	1	1	0

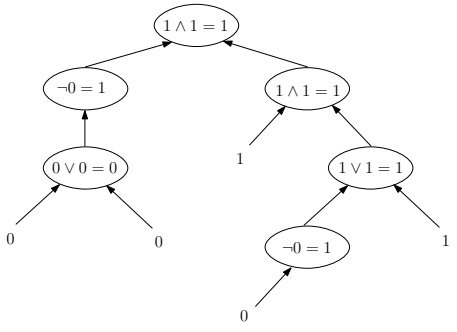
Man beachte, dass wir dieselben Operatorzeichen benutzen: einmal bei der Verknüpfung von Booleschen Termen (Syntax) und zum anderen beim Rechnen mit Wahrheitswerten. Dies ist eigentlich unzulässig, aber aus dem jeweiligem Kontext wird klar, auf welcher Ebene wir uns befinden, und bei der Definition der Semantik eines Booleschen Terms werden wir die inhaltliche Rechtfertigung finden.

An folgendem Beispiel wollen wir die einfache Konstruktion der sogenannten *Baumdarstellung eines Booleschen Terms* zeigen. Wir betrachten einen vollständig geklammerten Booleschen Term, zum Beispiel

$$t = ((\neg(x \vee y)) \wedge (z \wedge ((\neg y) \vee z)))$$

Die Klammerung widerspiegelt eine Hierarchie. Als letzte Operation wurde eine Konjunktion ausgeführt. Diese greift links zu auf die Negation einer Disjunktion, rechts auf eine Konjunktion, usw. Auf der untersten Ebene stehen die Variablen.





Es sollte klar sein, dass ein vollständig geklammerter Term die Baumdarstellung eindeutig bestimmt und umgekehrt. Das rechte Bild zeigt die bottom-up-Auswertung für eine konkrete Belegung, s.u.

Ein vollständig geklammerter Term t hat einen Rang $rg(t)$. Dies ist eine natürliche Zahl, die die maximale Schachtelungstiefe von Klammern angibt, bzw. in der zugehörigen Baumdarstellung entspricht dies der maximalen Anzahl von Schritten, die man vom obersten Niveau nach unten laufen kann, bis man bei einer Variablen angekommen ist. In obigem Beispiel ist der Rang 4. Formal folgt die Definition des Ranges dem induktiven Aufbau der Terme über einer Menge A von Einzelaussagen und V von Variablen.

Definition:(Rang eines Booleschen Terms)

1. Für jedes $a \in A$ und jedes $v \in V$ sei $rg(a) = rg(v) = 0$.
2. Falls $t = (\neg t_1)$, so sei $rg(t) = rg(t_1) + 1$.
3. Falls $t = (t_1 \vee t_2)$ oder falls $t = (t_1 \wedge t_2)$, so sei $rg(t) = \max\{rg(t_1), rg(t_2)\} + 1$

Wir werden später die Rangfunktion benutzen, um Aussagen über Boolesche Terme mittels vollständiger Induktion über den Formelrang zu beweisen.

Die Interpretation eines Booleschen Terms:
Zunächst hängt die Interpretation eines Terms t von der konkreten Belegung der Variablen V mit Aussagen, genauer mit dem Wahrheitswert dieser Aussagen zusammen.
Sei $\beta : V \to \mathbb{B}$ eine solche konkrete Zuordnung. Ziel ist es einen Wahrheitswert $I_\beta(t)$ zu finden, dies ist die Interpretation von t bezüglich β.
Wieder folgt die Bestimmung dieses Wertes dem induktiven Aufbau der Formeln.

Definition: Die Interpretation von t über V bezüglich der konkreten Belegung β ist gegeben durch:

1. Falls $t = a$ mit $a \in A$ so ist $I_\beta(a)$ nach Annahme bekannt, das ist der Wahrheitswert der Einzelaussage a. Insbesondere ist $I_\beta(true) = 1, I_\beta(false) = 0$
   Falls $t = v$ für $v \in V$ so setzen wir $I_\beta(v) = \beta(v)$.
2. Falls $t = (\neg t_1)$, so sei $I_\beta(t) = \neg I_\beta(t_1)$.
3. Falls $t = (t_1 \vee t_2)$, so sei $I_\beta(t) = I_\beta(t_1) \vee I_\beta(t_2)$.
   Analog, falls $t = (t_1 \wedge t_2)$, so sei $I_\beta(t) = I_\beta(t_1) \wedge I_\beta(t_2)$.

Anmerkungen:

1. Schaut man sich die Baumdarstellung des Termes an, so entspricht die Interpretation des Termes bezüglich β einer bottom–up–Auswertung des Baumes. An die Stelle der Variablen treten die konkreten Wahrheitswerte und die Operatoren sind dann jene aus der Booleschen Algebra.
2. Nochmal zu Verdeutlichung: In der Gleichung $I_\beta(t_1 \vee t_2) = I_\beta(t_1) \vee I_\beta(t_2)$ steht auf der linken Seite das $\vee$ aus der Syntaxdefinition der Terme, das $\vee$ auf der rechten Seite operiert auf der Ebene der Wahrheitswerte!

Wenn $V = \{x_1, \ldots, x_n\}$, so kann man eine konkrete Belegung β identifizieren mit einem sogenannten n-Tupel $(b_1, \ldots, b_n)$ von Wahrheitswerten. Das ist einfach die geordnete Aufzählung der Werte von β, genauer $b_i = \beta(x_i)$. Die Menge aller 2^n verschiedenen solcher n–Tupel bezeichnet man mit $\underbrace{\mathbb{B} \times \mathbb{B} \times \ldots\ldots \times \mathbb{B}}_{n \text{ mal}}$.

Definition: Eine n-stellige Boolesche Funktion f ist eine Funktion

$$f : \underbrace{\mathbb{B} \times \mathbb{B} \times \ldots\ldots \times \mathbb{B}}_{n \text{ mal}} \to \mathbb{B}$$

.

Damit definiert jeder Boolesche Term t über einer n–elementigen Variablenmenge eine n–stellige Boolesche Funktion

$$f_t : \mathbb{B} \times \mathbb{B} \times \ldots\ldots \times \mathbb{B} \to \mathbb{B}$$

$$f_t(b_1, \ldots, b_n) = I_\beta(t)$$

Merke: Jeder Boolesche Term definiert eine Boolesche Funktion, das ist seine Interpretation, seine abstrakte Bedeutung, seine **Semantik**.

Definition:

1. Zwei Boolesche Terme t_1, t_2 heißen *semantisch äquivalent*, geschrieben $t_1 \equiv t_2$, falls ihre Interpretationen gleich sind, das heißt, der Werteverlauf (die Wertetabellen) der Funktionen f_{t_1} und f_{t_2} sind gleich.

2. Ein Boolescher Term t heißt *erfüllbar*, falls es eine konkrete Belegung $(b_1, \ldots, b_n)$ der Variablen gibt, so dass $f_t(b_1, \ldots, b_n) = 1$. Diese Belegung heißt dann *erfüllend*.

3. Ein Boolescher Term heißt *Tautologie* oder auch *logisch gültig*, falls jede Belegung erfüllend ist.

4. Ein Boolescher Term heißt *Kontradiktion*, falls keine Belegung erfüllend ist.

Wie überprüft man algorithmisch, ob Terme diese Eigenschaften haben? Eine brute–force–Lösung besteht in der Konstruktion der Wertetabellen. Man beachte, dass bei $n = 64$ die Tabelle 2^{64} Zeilen hat! Dauert das Auswerten einer Zeile $10^{-6}s$ auf einem Rechner, so dauert das Auswerten der gesamten Tabelle ca. $5 \cdot 10^5$ Jahre!!!
Für kleine n kann man das natürlich noch machen und so ist etwa der folgende Term $t' = (((\neg x) \wedge (\neg y)) \wedge z)$ zu obigen Term t in der Baumdarstellung semantisch äquivalent.

Konventionen I:
Zur vereinfachten Darstellung von Booleschen Termen werden folgende Vereinbarungen getroffen.
1. Das äußere Klammerpaar kann weggelassen werden.
2. Die Bindungskraft der logischen Operatoren $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$ nimmt in dieser Reihung von links nach rechts ab!
Bsp.: Der Term $t = (((\neg x) \wedge (\neg y)) \wedge z)$ lässt sich somit schreiben als $t = (\neg x \wedge \neg y) \wedge z$
3. Auch die Verwendung verschiedener Klammerarten erleichtert die Lesbarkeit.

4. Im Zweifelsfall immer Klammern setzen!!!

Einige semantische Äquivalenzen sind besonders wichtig und heißen deshalb **Boolesche Gesetze**.

Satz: Für beliebige Formeln α, β, γ gelten die folgenden semantischen Äquivalenzen:

$$
\begin{array}{rcll}
(\alpha \wedge \beta) \wedge \gamma & \equiv & \alpha \wedge (\beta \wedge \gamma) & \\
(\alpha \vee \beta) \vee \gamma & \equiv & \alpha \vee (\beta \vee \gamma) & \text{(Assoziativität)} \\
\alpha \wedge \beta & \equiv & \beta \wedge \alpha & \\
\alpha \vee \beta & \equiv & \beta \vee \alpha & \text{(Kommutativität)} \\
\alpha \wedge (\beta \vee \gamma) & \equiv & (\alpha \wedge \beta) \vee (\alpha \wedge \gamma) & \\
\alpha \vee (\beta \wedge \gamma) & \equiv & (\alpha \vee \beta) \wedge (\alpha \vee \gamma) & \text{(Distributivität)} \\
\alpha \wedge \alpha & \equiv & \alpha & \\
\alpha \vee \alpha & \equiv & \alpha & \text{(Idempotenz)}
\end{array}
$$

$$
\begin{array}{rcll}
\alpha \wedge (\alpha \vee \beta) & \equiv & \alpha & \\
\alpha \vee (\alpha \wedge \beta) & \equiv & \alpha & \text{(Absorption)} \\
\neg(\alpha \wedge \beta) & \equiv & \neg\alpha \vee \neg\beta & \\
\neg(\alpha \vee \beta) & \equiv & \neg\alpha \wedge \neg\beta & \text{(deMorgansche Regel)} \\
\neg\neg\alpha & \equiv & \alpha & \text{(Doppelnegation)} \\
\alpha \Rightarrow \beta & \equiv & \neg\alpha \vee \beta & \\
\alpha \Rightarrow \beta & \equiv & \neg\beta \Rightarrow \neg\alpha & \text{(Kontraposition)} \\
\alpha \Leftrightarrow \beta & \equiv & (\alpha \wedge \beta) \vee (\neg\alpha \wedge \neg\beta) & \\
\alpha \Rightarrow \beta \wedge \gamma & \equiv & (\alpha \Rightarrow \beta) \wedge (\alpha \Rightarrow \gamma) & \\
\alpha \Rightarrow \beta \vee \gamma & \equiv & (\alpha \Rightarrow \beta) \vee (\alpha \Rightarrow \gamma) & \\
\alpha \wedge \beta \Rightarrow \gamma & \equiv & (\alpha \Rightarrow \gamma) \vee (\beta \Rightarrow \gamma) & \\
\alpha \vee \beta \Rightarrow \gamma & \equiv & (\alpha \Rightarrow \gamma) \wedge (\beta \Rightarrow \gamma) &
\end{array}
$$

Die Beweise können einfach mit Hilfe von Wertetabellen geführt werden. (s. Übung)

Beispiel: Man beachte, dass im obigen Satz α, β, γ ganze Formeln sein können, nicht nur einzelne Variable. Der Beweis der folgenden Äquivalenz mit Wahrheitstafeln würde 16 Zeilen erfordern. Verwendet man dagegen die Absorption und die doppelte Negation zur Ersetzung von Subformeln, so erhält man einen einfachen und kurzen Beweis.

$$
\begin{array}{ll}
p_1 \vee ((p_2 \vee p_3) \wedge \neg(\neg p_1 \wedge (\neg p_1 \vee p_4))) & \equiv p_1 \vee ((p_2 \vee p_3) \wedge \neg\neg p_1) \\
\equiv p_1 \vee ((p_2 \vee p_3) \wedge p_1) & \equiv p_1
\end{array}
$$

Wir wollen das zugrundeliegende Substitutionsprinzip noch einmal explizit formulieren. Bezeichne $t[x/t']$ den Booleschen Term, der entsteht, wenn ich im Term t jedes Vorkommen der Variable x durch den Term t' ersetze.

Satz:(Substitution und semantische Äquivalenz)
Sei t ein Boolescher Term, x eine darin vorkommende Variable und t_1, t_2 zwei semantisch äquivalente Terme. Es gilt:

$$t[x/t_1] \equiv t[x/t_2] \quad \text{und} \quad t_1[x/t] \equiv t_2[x/t]$$

Hinweis: Man mache sich dies an den zugehörigen Baumdarstellungen klar!
Wer das Formale nachlesen will: Broy, Informatik, eine grundlegende Einführung, Band 1, Springer, 1998

Konvention II: Wegen der Assoziativität von Konjunktion bzw. Disjunktion schreiben wir $\alpha \wedge \beta \wedge \gamma$ bzw. $\alpha \vee \beta \vee \gamma$ auch völlig ohne Klammern, obwohl natürlich dies weiterhin für uns zweistellige Operatoren sind!

1.2 Von der Booleschen Funktion zum Booleschen Term

Wir haben gesehen, wie man einem Booleschen Term t über einer Variablenmenge $V = \{x_1, \ldots, x_n\}$ seine Interpretation zuordnet, dies ist eine n–stellige Boolesche Funktion $f_t : \mathbb{B} \times \ldots \times \mathbb{B} \to \mathbb{B}$.

Beobachtung 1: Es gibt unendlich viele syntaktisch verschiedene Boolesche Terme, die ein und dieselbe Boolesche Funktion repräsentieren.
Beweis: Wie wir wissen ist zum Beispiel $t \equiv t \vee t \equiv t \vee t \vee t \equiv \ldots$. Alle diese syntaktisch verschiedenen Booleschen Terme haben dieselbe semantische Interpretation f_t. □

Beobachtung 2: Es gibt genau 2^{2^n} viele verschiedene n–stellige Boolesche Funktionen.
Beweis: Eine Funktion ist eineindeutig bestimmt durch ihre Wertetabelle. Statt Funktionen zu zählen, zählen wir Wertetabellen. Bei einer n–stelligen Booleschen Funktion hat diese 2^n Zeilen. Das sind die n–Tupel aus $\mathbb{B} \times \ldots \times \mathbb{B}$. Jeder Zeile wird durch die Funktion einer von zwei möglichen Werten aus $\mathbb{B}$ zugeordnet. Insgesamt gibt es

$$\underbrace{2 \cdot 2 \cdot 2 \cdot \ldots \cdot 2}_{2^n \text{ Faktoren}} = 2^{2^n}$$

Möglichkeiten, die Wertetabelle zu vervollständigen.

Beobachtung 1 sagt, wenn wir einen Term gefunden haben, der eine konkrete Funktion repräsentiert, so gibt es unendlich viele, syntaktisch verschiedene Terme, die dies auch tun. Bleibt die Frage, gibt es für jede Boolesche Funktion wenigstens einen Term, der sie repräsentiert?

Aufgabe: Gegeben sei eine n–stellige Boolesche Funktion g. Finde einen Booleschen Term $t(g)$, so dass $f_{t(g)} = g$, der also genau die vorgegebene Semantik hat.

Wir betrachten zunächst folgende Teilaufgabe:
Sei g eine n–stellige Boolesche Funktion, die genau für das konkrete Argument $(b_1, b_2, \ldots, b_n) \in \mathbb{B} \times \ldots \times \mathbb{B}$ den Wert 1 annimmt.
Finde einen Term über der Variablenmenge $V = \{x_1, x_2, \ldots, x_n\}$, der diese Funktion repräsentiert!

Beispiel:
Sei $n = 3$ und $(b_1, b_2, b_3) = (1, 1, 0)$. Wir betrachten den Term $t = x_1 \wedge x_2 \wedge \neg x_3$ und überlegen uns, dass er das Gewünschte leistet. Zunächst ist $(1, 1, 0)$ erfüllende Belegung.

Weiterhin ist keine andere Belegung erfüllend, denn sie müsste sich von $(1,1,0)$ unterscheiden. Das heißt, an mindestens einer der ersten beiden Stellen steht eine 0 oder die dritte Stelle ist 1. Mithin wird die Konjunktion zu 0 ausgewertet.

Wir verallgemeinern das Beispiel zur Lösung unserer Teilaufgabe: Dazu vereinbaren wir folgende Notation: $x_i^{b_i} = x_i$ für $b_i = 1$ und $x_i^{b_i} = \neg x_i$ für $b_i = 0$.
Nun betrachten wir den Term $t = x_1^{b_1} \wedge \ldots \wedge x_n^{b_n}$. Er wird bei der Belegung mit $(b_1,\ldots,b_n)$ zu 1 ausgewertet, während jede andere Belegung die Konjunktion zu 0 macht.
Der Term $\neg t = \neg(x_1^{b_1} \wedge \ldots \wedge x_n^{b_n}) \equiv x_1^{\neg b_1} \vee \ldots \vee x_n^{\neg b_n}$ wird also dann bei Belegung mit $(b_1,\ldots,b_n)$ zu 0 ausgewertet, bei allen anderen Belegungen zu 1.

Was hilft uns dies bei der Lösung unserer eigentlichen Aufgabe?
Sei $g^{-1}(1) = \{(b_1,\ldots,b_n) | g(b_1,\ldots,b_n) = 1\}$ das Urbild der 1 und $g^{-1}(0)$ das Urbild der 0 bei der Funktion g. Wir haben zumindestens die folgenden zwei Möglichkeiten, zu einem Term t zu kommen:

- Variante I: Sei t ein Boolescher Term, der genau dann zu 1 ausgewertet wird, wenn man diese Belegung **oder** diese Belegung **oder**...**oder** diese Belegung aus $g^{-1}(1)$ wählt.
- Variante II: Sei t ein Boolescher Term, der genau dann zu 1 ausgewertet wird, wenn man **nicht** diese Belegung **und nicht** diese Belegung **und**...**und nicht** diese Belegung aus $g^{-1}(0)$ gewählt hat

Bevor wir das formalisieren, noch ein paar Begriffe:

Definition: Ein *Literal* ist eine Variable oder deren Negation.
Eine Disjunktion $\alpha_1 \vee \ldots \vee \alpha_n$ $(n \geq 1)$ wird *disjunktive Normalform* (kurz DNF) genannt, wenn jedes α_i eine Konjunktion von Literalen oder ein einzelnes Literal ist.
Eine Konjunktion $\alpha_1 \wedge \ldots \wedge \alpha_n$ $(n \geq 1)$ wird *konjunktive Normalform* (kurz KNF) genannt, wenn jedes α_i eine Disjunktion von Literalen oder ein einzelnes Literal ist.

Beispiele: $(x_1 \wedge x_2) \vee (\neg x_1 \wedge x_3) \vee x_2$ ist eine DNF, aber keine KNF.
Die Formeln $x_1 \vee x_2$ und $\neg x_1 \wedge x_4 \wedge \neg x_6$ sind sowohl DNF's als auch KNF's.

Satz: Jede n–stellige Boolesche Funktion f ist durch die sogenannte kanonische DNF $\mathrm{dnf}(f)$ und durch die kanonische KNF $\mathrm{knf}(f)$ über der Variablenmenge $V = \{x_1, x_2, \ldots, x_n\}$ repräsentierbar, wobei:

$$\begin{aligned} \mathrm{dnf}(f) &= \bigvee_{(b_1,\ldots,b_n)\in f^{-1}(1)} (x_1^{b_1} \wedge \ldots \wedge x_n^{b_n}) \\ \mathrm{knf}(f) &= \bigwedge_{(b_1,\ldots,b_n)\in f^{-1}(0)} (x_1^{\neg b_1} \vee \ldots \vee x_n^{\neg b_n}) \end{aligned}$$

Im Spezialfall $f^{-1}(1) = \emptyset$ setzen wir $\mathrm{dnf}(f) = false$ und im Spezialfall $f^{-1}(0) = \emptyset$ setzen wir $\mathrm{knf}(f) = true$.
Beweis: Zunächst stellen wir fest, dass das genau die Umsetzung der obigen Varianten I und II ist, man störe sich nicht daran, dass die Funktion jetzt f heißt...

Nochmal formal für die DNF: Wir überzeugen uns davon, dass $x_1^{b_1} \wedge \ldots \wedge x_n^{b_n}$ für die konkrete Belegung $(b_1, \ldots, b_n)$ wahr (1) und für alle anderen Belegungen falsch (0) ist. In der Tat wird eine Konjunktion von Literalen genau dann 1, wenn jedes Literal 1 ist, und diese Situation wird nur bei der Belegung $(b_1, \ldots, b_n)$ erreicht. Die Disjunktion über alle $(b_1, \ldots, b_n) \in f^{-1}(1)$ führt dazu, dass alle Belegungen, die aus dieser Menge kommen, durch "ihre" Konjunktion akzeptiert werden, während Tupel aus $f^{-1}(0)$ von allen Konjunktionen aus $\mathrm{dnf}(f)$ verworfen werden.
Der Beweis für $\mathrm{knf}(f)$ ist analog: Jede Disjunktion $x_1^{\neg b_1} \vee \ldots \vee x_n^{\neg b_n}$ wird für die konkrete Belegung $(b_1, \ldots, b_n)$ falsch (0) und wahr für alle anderen Belegungen. Durch die Konjunktion über alle $(b_1, \ldots, b_n) \in f^{-1}(0)$ erzeugt man eine Formel, welche die vorgegebene Funktion f repräsentiert. □

Korollar: Jeder Boolesche Term t ist semantisch äquivalent zu einem Term in DNF und zu einem Term in KNF.
Beweis: Wir bilden zu t die zugehörige Interpretation f_t und für diese Funktion dann die kanonische DNF bzw. kanonische KNF. □
Alternativ kann man einen Beweis mittels vollständiger Induktion über den Formelrang von t führen (später).

Ausblick: Zum Verständnis von DNF und KNF kann man auch geometrische bzw. graphentheoretische Hilfsmittel einsetzen: Die Menge der n-Tupel $\{0,1\}^n$ bildet auch die Knotenmenge des n-dimensionalen Würfelgraphen Q_n. Jede Konjunktion von k Literalen repräsentiert einen $(n-k)$-dimensionalen Unterwürfel. Eine DNF muss durch die Unterwürfel der in ihr auftretenden Konjunktionen die Knotenmenge $f^{-1}(1)$ überdecken. In der kanonischen DNF wird jeder Knoten einzeln (als 0-dimensionaler Unterwürfel) überdeckt. Man sieht, dass man diese DNF vereinfachen kann, wenn mehrere Knoten aus $f^{-1}(1)$ einen höherdimensionalen Unterwürfel bilden.
Wir kommen darauf im 2. Logikteil am Ende des Semesters zurück.

Definition: Eine Menge von logischen Junktoren, die man zur Formelbildung einsetzt, wird *logische Signatur* genannt. Die Signatur $\{\neg, \vee, \wedge\}$ heißt *Boolesche Standardsignatur*. Eine logische Signatur ist *funktional vollständig*, wenn jede Boolesche Funktion durch eine mit dieser Signatur gebildeten Formel repräsentierbar ist.

Satz: Die Boolesche Signatur sowie die Signaturen $\{\neg, \wedge\}$ und $\{\neg, \vee\}$ sind funktional vollständig.

Die Vollständigkeit der Booleschen Standardsignatur folgt aus der Existenz von DNF's bzw. KNF's. Man beachte, daß die Formeln $p \wedge \neg p$ und $p \vee \neg p$ Funktionen repräsentieren, die konstant 0 bzw. konstant 1 sind. So erhält man auch die 0–stelligen Funktionen.
Mit den deMorganschen Regeln kann man die Disjunktion (bzw. Konjunktion) durch Negation und Konjunktion (bzw. Negation und Disjunktion) eliminieren. So kann die Vollständigkeit der Signaturen $\{\neg, \wedge\}$ und $\{\neg, \vee\}$ auf die Vollständigkeit der Booleschen Standardsignatur zurückgeführt werden.

Es ist leicht zu sehen, daß die Signaturen $\{\vee\}$, $\{\wedge\}$ und $\{\vee, \wedge\}$ nicht funktional vollständig sind. Etwas schwerer ist der Beweis der Unvollständigkeit der Signatur $\{\neg, \Leftrightarrow\}$.

Dagegen ist die Signatur $\{\mid\}$, wobei $\mid$ den "nicht und"–Junktor (bekannt auch als NAND bzw. Sheffer–Strich) bezeichnet, funktional vollständig. Ebenso ist das NOR (Negation der Disjunktion) allein schon vollständig.

1.3 Der Gebrauch von Quantoren

Definition: Eine *Aussageform* $P(x_1,\ldots,x_n)$, man spricht auch von einem Prädikat, über den Universen $U_1,U_2,\ldots,U_n$ ist ein Satz mit freien Variablen $x_1,x_2,\ldots,x_n$, der zur Aussage wird (also einen Wahrheitswert hat), wenn jedes x_i durch einen konkreten Wert aus U_i ersetzt wird.

Zum Beispiel sind "$P(x):\ x>1$" oder "$Q(x):\ x+0=x$" bzw. "$R(x,y):\ x+y=x$" Aussageformen für den Bereich der ganzen Zahlen. Dies sind so noch keine Aussagen, dazu bedarf es der Belegung der Variablen mit konkreten Werten!
Also, $P(1)$ ist falsch, $Q(0)$ ist wahr, $R(1,y)$ ist noch keine Aussage und $R(1,0)$ ist wahr.

Es gibt weitere Möglichkeiten Aussagen mit Aussageformen zu verbinden.
Sei $P(x), x\in U$ eine Aussageform.

- Wir betrachten die Aussage (!): "Für jedes konkrete $x\in U$ gilt $P(x)$". Dies ist eine Aussage, denn entweder stimmt es oder es stimmt nicht. In der mathematischen Notation wird dafür der sogenannte **Allquantor** $\forall$ benutzt:

$$\forall x\in U:\ P(x)$$

- Weiterhin betrachten wir die Aussage "Es gibt (wenigstens) ein $x\in U$, für das $P(x)$ wahr ist". Zur Notation benutzt man den sogenannten **Existenzquantor** $\exists$:

$$\exists x\in U:\ P(x)$$

Wenn das zugrundeliegende Universum aus dem Kontext klar ist, schreibt man auch verkürzt $\forall x\,P(x)$ bzw. $\exists x\,P(x)$.

Beispiele: Die Aussagen "$\forall x\in\mathbb{N}:\ x+0=x$" und "$\exists x\in\mathbb{N}:\ x^2=x$" sind wahr, die Aussagen "$\exists x\in\mathbb{N}:\ x+1=x$" und "$\forall x\in\mathbb{N}:\ x^2=x$" sind falsch.

Da wir es mit Aussagen zu tun haben, können wir diese auch mittels logischer Junktoren zu neuen Aussagen zusammensetzen und wir können von semantischer Äquivalenz sprechen.

Satz: Für beliebige Prädikate $P(x)$ und $Q(x)$ gelten die folgenden semantischen Äquivalenzen:

$$\begin{array}{rcl@{\qquad}rcl}
\neg\forall x\,P(x) & \equiv & \exists x\neg P(x) & \neg\exists x\,P(x) & \equiv & \forall x\neg P(x)\\
\forall x\,P(x)\wedge\forall x\,Q(x) & \equiv & \forall x\,(P(x)\wedge Q(x)) & \exists x\,P(x)\vee\exists x\,Q(x) & \equiv & \exists x\,(P(x)\vee Q(x))\\
\forall x\forall y\,P(x,y) & \equiv & \forall y\forall x\,P(x,y) & \exists x\exists y\,P(x,y) & \equiv & \exists y\exists x\,P(x,y)
\end{array}$$

Achtung: Die folgenden Formelpaare sind im allgemeinen nicht semantisch äquivalent:

$$\begin{array}{rcl}
(\forall x\,P(x)\vee\forall x\,Q(x)) & \not\equiv & \forall x\,(P(x)\vee Q(x))\\
(\exists x\,P(x)\wedge\exists x\,Q(x)) & \not\equiv & \exists x\,(P(x)\wedge Q(x))
\end{array}$$

Konkrete Beispiele für die letzte Bemerkung erhält man für den Bereich der natürlichen Zahlen, wenn $P(x)$ (bzw. $Q(x)$) aussagt, daß x eine gerade (bzw. ungerade) Zahl ist.

Nochmal zur Schachtelung von mehreren Quantoren:
Für ein Prädikat $P(x,y)$ mit zwei freien Variablen können wir neue Prädikate bilden, zum Beispiel das Prädikat $Q(x) : \forall y\, P(x,y)$. Wenn man jetzt für x konkrete Werte einsetzt, bekommt man Aussagen. $\forall x \forall y\ P(x,y)$ beschreibt dann die Aussage, dass $Q(x)$ für jeden konkreten Wert von x wahr ist.

Quantifizierung von zwei Variablen

Aussage	Wann wahr?	Wann falsch?
$\forall x\, \forall y\, P(x,y)$ $\forall y\, \forall x\, P(x,y)$	$P(x,y)$ wahr für jedes Paar (x,y)	Es gibt wenigstens ein Paar (x,y), für das $P(x,y)$ falsch ist
$\forall x\, \exists y\, P(x,y)$	Für jedes x gibt es ein (sein) y, so dass $P(x,y)$ wahr ist	Es gibt ein x, so dass für jedes y $P(x,y)$ falsch ist
$\exists x\, \forall y\, P(x,y)$	Es gibt ein x, das für jedes y $P(x,y)$ wahr macht	Für jedes x gibt es ein y, so dass $P(x,y)$ falsch ist.
$\exists x\, \exists y\, P(x,y)$ $\exists y\, \exists x\, P(x,y)$	Es gibt ein Paar (x,y), so dass $P(x,y)$ wahr ist.	$P(x,y)$ ist falsch für jedes Paar (x,y)

Die oben angegebene Regel zur Negation von quantifizierten Aussagen gelten auch für geschachtelte Quantoren. Insbesondere gilt, vgl. letzte Spalte der Tabelle:

$$\neg\forall x\, \forall y\, P(x,y) \equiv \exists x(\neg\forall y\, P(x,y)) \equiv \exists x \exists y\, \neg P(x,y)$$
$$\neg\forall x\, \exists y\, P(x,y) \equiv \exists x\, \forall y\, \neg P(x,y)$$
$$\neg\exists x\, \forall y\, P(x,y) \equiv \forall x\, \exists y\, \neg P(x,y)$$
$$\neg\exists y\, \exists x\, P(x,y) \equiv \forall x\, \forall y\, \neg P(x,y)$$

Wir sprechen von *Negationsnormalform*, wenn die Negationsjunktoren nicht außerhalb von Quantoren stehen, sondern sich nur auf atomare Formeln beziehen und ansonsten die Standardsignatur verwendet wird.

Beispiele:

1. Wir negieren die Aussage, dass es für alle $x < y$ immer ein z gibt mit $x < z < y$.

$$\begin{aligned}
& \neg(\forall x\ (\forall y\ (x < y \Rightarrow (\exists z\ (x < z \wedge z < y))))) \\
\equiv\ & \exists x \exists y\ \neg((x < y \Rightarrow (\exists z\ (x < z \wedge z < y)))) \\
\equiv\ & \exists x \exists y\ (\neg(\neg(x < y) \vee (\exists z\ ((x < z \wedge z < y))))) \\
\equiv\ & \exists x \exists y\ ((x < y) \wedge (\forall z\ \neg((x < z) \wedge (z < y)))) \\
\equiv\ & \exists x \exists y\ ((x < y) \wedge (\forall z\ ((x \geq z) \vee (z \geq y))))
\end{aligned}$$

2. Gelegentlich wird der Quantor $\exists!$ verwendet um auszudrücken, dass es genau ein Individium aus dem Grundbereich gibt, das das Prädikat zur wahren Aussage macht. Dies kann man äquivalent schreiben als

$$\exists! x\, P(x) \equiv \exists x (P(x) \wedge \forall y\, (x \neq y \Rightarrow \neg P(y)))$$

Und damit

$$\neg\exists! x\, P(x) \equiv \forall x\, (\neg P(x) \vee \exists y\, (x \neq y) \wedge P(y)))$$

Zum Schluss dieses Abschnitts noch ein Hinweis auf algorithmische Aspekte: Allgemein ist die Frage, ob eine durch Quantoren gebildete Aussage wahr oder falsch ist, algorithmsch nicht entscheidbar. In vielen konkreten Fällen kann man die Frage aber durch genauere Überlegungen beantworten. Wie kann man in solchen Fällen sich selbst und andere von der Richtigkeit seiner Überlegungen überzeugen? Der typische Beweis dafür, dass eine quantifizierte Aussage wahr ist, erfolgt in drei Stufen. Zuerst wird die Aussage durch Anwendung von äquivalenten Umformungen aus der Aussagenlogik und aus dem obigen Satz über Äquivalenzen in eine Standardform gebracht, bei der alle auftretenden Quantoren am Anfang stehen (man nennt dies eine *Pränexform*). Danach erfolgt die Belegung der Variablen in Form eines Spiels zwischen zwei Parteien: Einem *Beweiser* und seinem *Gegenspieler*, der nachzuweisen versucht, dass die Ausage falsch ist. Dabei darf der Gegenspieler bei jedem Allquantor die entsprechende Variable x durch ein beliebiges Objekt a aus dem Individuenbereich belegen. Sollte die Aussage doch falsch sein (also nicht für alle Objekte gelten), würde der Gegenspieler gerade ein solche Objekt wählen. Ist die Aussage wahr, dann ist es (für den Beweiser) egal, welches Objekt a der Gegenspieler gewählt hat. Der Beweiser ist bei allen Existenzquantoren am Zuge und muss ein passendes Objekt (in Abhängigkeit von den vorher vom Gegenspieler gwählten Objekten) finden, für welches die nachfolgende Ausssage wahr ist. Nachdem alle Variablen belegt sind, haben wir eine (variablenfreie) Aussage. Im letzten Schritt muss diese Aussage verifiziert (als wahr bewiesen) werden.

Wir wollen die drei Stufen eines solchen Beweises an einem einfachen Beispiel demonstrieren und die folgende Aussage für den Bereich der rationalen Zahlen beweisen.

Aussage: Für beliebige $x, y \in \mathbb{Q}$ mit $x < y$ gibt es eine von x und y verschiedene Zahl $z \in \mathbb{Q}$, die zwischen x und y liegt.

Die Beweisidee liegt klar auf der Hand: man setzt für z den Mittelwert aus x und y und kann dann die Behauptung nachrechnen. Wir stellen die Aussage als Formel mit Quantoren dar. Der Bereich $\mathbb{Q}$ ist durch Verabredung festgelegt und wird nicht explizit genannt:

$$\forall x\, \forall y\, [(x < y) \Rightarrow \exists z\, (x < z \wedge z < y)]$$

In diesem Fall könnte man die erste Stufe überspringen und gleich mit dem Spiel der Festlegung der Werte beginnen, bei dem der Gegner x und y vorgibt und wir im Fall $x < y$ (als Beweiser) $z = \frac{x+y}{2}$ setzen. Um das dreistufige Verfahren formal korrekt umzusetzen, würden wir aber zuerst die Implikation hinter den beiden

Allquatoren äquivalent umformen:

$$\begin{aligned}(x<y)\Rightarrow\exists z\,(x<z\wedge z<y) &\equiv\neg(x<y)\vee\exists z\,(x<z\wedge z<y)\\ &\equiv\exists z\,(x\geq y)\vee\exists z\,(x<z\wedge z<y)\\ &\equiv\exists z\,(x\geq y\vee\ (x<z\wedge z<y))\end{aligned}$$

Im ersten Schritt wurde die Regel $p\Rightarrow q\equiv\neg p\vee q$ angewendet. Im zweiten Schritt wurde die Negation von $x<y$ in $x\geq y$ umgewandelt und der Existenzquantor $\exists z$ davorgesetzt - das kann man machen, weil $x\geq y$ in keiner Weise von z abhängt. Im letzten Schritt wurde die vierte Regel aus obigem Satz verwendet. Jetzt liegt die Aussage in Pränexform vor:

$$\forall x\,\forall y\,\exists z\,[x\geq y\vee(x<z\wedge z<y)].$$

Das Spiel beginnt mit der Vorgabe von zwei Werten $x=a$ und $y=b$ durch den Gegenspieler. Der Beweiser setzt $z=(a+b)/2$.
Nun erfolgt die Verifikation der Aussage $a\geq b\vee(a<(a+b)/2\wedge(a+b)/2<y)$ durch Betrachtung von zwei Fällen: Entweder es gilt $a\geq b$, dann wird die Aussage durch den linken Term der Disjunktion wahr oder es gilt $a<b$. In diesem Fall müssen beide Ungleichungen auf der rechten Seite der Disjunktion erfüllt sein, aber beides folgt über den Zwischenschritt $a/2<b/2$ jeweils kombiniert mit den Ungleichungen $a/2\leq a/2$ bzw. $b/2\leq b/2$.

Bleibt die Frage, warum man diese Argumentation nicht für den Bereich $\mathbb{N}$ der natürlichen Zahlen wiederholen kann. Die Antwort ist wieder nicht schwer: Der Ausdruck $\frac{a+b}{2}$ führt in vielen Fällen aus dem Bereich $\mathbb{N}$ heraus. Das ist aber noch kein Beweis dafür, dass die Aussagge über diesem Bereich falsch ist. Um das zu zeigen, beweist man, dass die negierte Aussage wahr ist. Wir bilden die Negation mit Hilfe der Regeln (1) und (2) aus obigem Satz und der deMorganschen Regel:

$$\exists x\,\exists y\,\forall z\,[(x<y)\wedge(x\geq z\vee z\geq y)].$$

Da die Aussage bereits in Pränexform vorliegt, kann das Spiel der Variablenbelegung beginnen. Wir setzen als Beweiser $x=0$ und $y=1$. Der Gegenspieler belegt z mit einem Wert c. Wir müssen jetzt die Aussage $0<1\wedge(0\geq c\vee c\geq 1)$ verifizieren. Offensichtlich ist $0<1$ wahr und wir müssen nur noch zeigen, dass mindestens eine der Ungleichungen aus $(0\geq c\vee c\geq 1)$ wahr ist. Wenn aber die erste Ungleichung nicht erfüllt ist, muss c eine natürliche Zahl größer als 0, also mindestens 1 sein und damit ist die zweite Ungleichung erfüllt.

2 Einführung Mengenlehre

Moderne Mengentheorie wird in Form eines axiomatischen Kalküls betrieben. Dieser Ansatz hat aber den Nachteil, dass einfache inhaltliche Fragen oft durch einen technisch komplizierten Apparat verdeckt werden. Wir werden uns deshalb auf die Entwicklung einer "naiven" Mengenlehre beschränken, die als sprachliches Werkzeug für die nachfolgenden Teile der Vorlesung völlig ausreichend ist.

Nach G. Cantor (1895) ist eine *Menge* "eine Zusammenfassung von bestimmten wohlunterschiedenen Objekten unserer Anschauung oder unseres Denkens (welche die Elemente der Menge genannt werden) zu einem Ganzen".
Der Sachverhalt, dass ein Objekt a Element einer Menge A ist, wird durch $a \in A$ dargestellt, anderenfalls schreibt man $a \notin A$.
Zwei Mengen A und B sind *gleich* ($A = B$), wenn sie die gleichen Elemente besitzen, d.h. wenn für alle a gilt: $a \in A$ dann und nur dann, wenn $a \in B$.

Darstellungen von Mengen:

(a) Mengen können durch *Auflistung ihrer Elemente* in geschweiften Klammern dargestellt werden. Das betrifft insbesondere endliche Mengen, wie z.B. $A = \{2,3,5,7\}$ oder $B = \{rot, gelb, blau\}$. Dabei ist die Reihenfolge der Elemente in der Auflistung ohne Bedeutung. Auch die Mehrfachnennung von Elementen ist erlaubt, sie hat aber nur Einfluß auf die Darstellung der Menge und nicht auf die Menge selbst, z.B. $\{2,3,5,7\} = \{5,7,3,2,2,5,2\}$.
Wir vereinbaren, dass auch unendliche Mengen durch Auflistung dargestellt werden können, sofern dies unmissverständlich ist, wie z.B. $\{0,1,2,3,\ldots\}$ für die natürlichen Zahlen oder $\{2,4,6,8,\ldots\}$ für die positiven geraden Zahlen.

(b) Die in der Mathematik gebräuchlichste Darstellungsform von Mengen beruht auf dem sogenannten *Abstraktionsprinzip*, nach dem man Mengen – im Sinne der Cantorschen Definition – durch wohlbestimmte Eigenschaften definieren kann. Dazu werden Prädikate $P(x)$, x über einem festgelegten Individuenbereich U, benutzt. Dann wird mit $\{x \in U \mid P(x)\}$ die Menge bezeichnet, die sich aus allen Individuen aus dem Bereich zusammensetzt, für die $P(x)$ wahr ist.
Hinweis: Dass man dabei schnell bei Paradoxien landet, zeigt Russells Beispiel. Wir betrachten die Menge aller Mengen, die sich nicht selbst als Element enthalten, also $M = \{X | X \notin X\}$. Dann ist $M \in M$ genau dann, wenn $M \notin M$, ein Widerspruch!

(c) Zur Veranschaulichung können Mengen durch sogenannte *Venn–Diagramme* als Kreisscheiben oder andere Flächen in der Ebene dargestellt werden.

Definitionen:

(a) Eine Menge A ist *Teilmenge* (oder *Untermenge*) einer Menge B (Schreibweise $A \subseteq B$), wenn für jedes $a \in A$ auch $a \in B$ folgt.
B heißt dann *Obermenge* von A.

Es gilt $A = B$ genau dann, wenn $A \subseteq B$ und $B \subseteq A$. Außerdem folgt aus $A \subseteq B$ und $B \subseteq C$ auch $A \subseteq C$.

(b) Zwei Mengen A und B sind *disjunkt*, wenn sie keine gemeinsamen Elemente besitzen, d.h. wenn aus $a \in A$ folgt $a \notin B$.

(c) Die *Vereinigung* $A \cup B$ der Mengen A und B besteht aus allen Elementen, die Elemente von A oder von B sind, d.h. $A \cup B = \{x \mid x \in A \ \vee \ x \in B\}$.

(d) Der *Durchschnitt* $A \cap B$ der Mengen A und B besteht aus allen Elementen, die Elemente von A und von B sind, d.h., $A \cap B = \{x \mid x \in A \ \wedge \ x \in B\}$.

(e) Die *Differenz* $A \setminus B$ der Mengen A und B besteht aus allen Elementen, die Elemente von A aber nicht von B sind, d.h. $A \setminus B = \{x \mid x \in A \ \wedge \ x \notin B\}$

(f) Die *symmetrische Differenz* $A \oplus B$ der Mengen A und B ist definiert als $(A \setminus B) \cup (B \setminus A)$.

(g) Das *Komplement* von A bezüglich des Universums U ist definiert als $U \setminus A$. Es wird mit $\overline{A}$ bezeichnet. Ist A durch die Aussageform $P(x)$ definiert, so ist $\overline{A}$ durch $\neg P(x)$ gegeben.

(h) Die Menge, die kein Element enthält, wird *leere Menge* genannt und mit $\emptyset$ bezeichnet.

Hinweise: Sind A durch $P(x)$ und B durch $Q(x)$ bestimmt, so gilt $A = B$ genau dann, wenn $P(x) \equiv Q(x)$.
Die leere Menge wird durch eine Kontradiktion definiert, ganz U durch eine Tautologie, wie etwa $x = x$.

Satz: Folgende Identitäten gelten für alle Untermengen A, B, C eines Universums U:

Kommutativität:	$A \cup B = B \cup A$
	$A \cap B = B \cap A$
Assoziativität:	$A \cup (B \cup C) = (A \cup B) \cup C$
	$A \cap (B \cap C) = (A \cap B) \cap C$
Distributivität:	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
Idempotenz:	$A \cup A = A$
	$A \cap A = A$
Dominanz:	$A \cup U = U$
	$A \cap \emptyset = \emptyset$
Identität:	$A \cup \emptyset = A$
	$A \cap U = A$
De Morgansche Regel:	$\overline{A \cup B} = \overline{A} \cap \overline{B}$
	$\overline{A \cap B} = \bar{A} \cup \bar{B}$
Komplementierung:	$A \cup \overline{A} = U$
	$A \cap \overline{A} = \emptyset$
	$\overline{(\overline{A})} = A$
	$A \setminus B = A \cap \overline{B}$

Die Beweise benutzen einfach die entsprechenden Booleschen Gesetze, man mache sich das am Beispiel klar!

Auf Grund der Assoziativität kann man bei der Vereinigung (bzw. beim Durchschnitt) von n Mengen $A_1, A_2, \ldots, A_n$ auf Klammerungen verzichten und die folgende Schreibweise nutzen:

$$\begin{aligned} A_1 \cup A_2 \cup \ldots \cup A_n = & \quad \bigcup_{i=1}^{n} A_i \\ A_1 \cap A_2 \cap \ldots \cap A_n = & \quad \bigcap_{i=1}^{n} A_i \end{aligned}$$

Definition: Ist I eine beliebige Menge und ist für jedes $i \in I$ eine Menge A_i gegeben, dann nennen wir die Menge dieser Mengen eine *Mengenfamilie* und bezeichnen sie durch $(A_i)_{i \in I}$. Die Vereinigung (bzw. der Durchschnitt) dieser Mengenfamilie ist definiert durch

$$\bigcup_{i \in I} A_i = \quad \{x \mid \text{es gibt ein } i \in I, \text{ so daß } x \in A_i\}$$

$$\bigcap_{i \in I} A_i = \quad \{x \mid \text{für alle } i \in I, \text{ gilt } x \in A_i\}$$

Definition: Eine Familie $\{A_i \mid i \in I\}$ von nichtleeren Mengen wird *Partition* oder *Zerlegung* einer Menge A genannt, falls

1) $A = \bigcup_{i \in I} A_i$ und

2) Für beliebige, voneinander verschiedene $i, j \in I$ gilt $A_i \cap A_j = \emptyset$.

Definition: Ist A eine Menge, dann wird die Menge aller Untermengen von A die *Potenzmenge* von A genannt und mit $\mathcal{P}(A)$ bezeichnet.

Satz: Ist A eine endliche, n-elementige Menge, dann hat die Potenzmenge $\mathcal{P}(A)$ genau 2^n Elemente.
Beweis: Seien $x_1, x_2, \ldots, x_n$ die Elemente der Menge. Eine Untermenge von A ist dadurch eineindeutig bestimmt, dass für jedes Element festgelegt ist, ob es zur Untermenge dazugehört oder nicht. Insgesamt haben wir also 2^n Möglichkeiten, eine solche Festlegung zu treffen.
Im Fall, dass $A = \emptyset$, ist A die einzige Teilmenge von sich, also $\mathcal{P}(\emptyset)$ hat $2^0 = 1$ Element. $\square$

3 Relationen und Funktionen

3.1 Grundbegriffe

Definition: Ein *geordnetes Paar* (a,b) ist ein (den Objekten a und b zugeordnetes) Konstrukt mit der folgenden Eigenschaft: $(a,b) = (a',b')$ genau dann, wenn $a = a'$ und $b = b'$

Definition: Das *kartesische Produkt* $A \times B$ von zwei Mengen A und B ist definiert als die Menge aller geordneten Paare (a,b) mit $a \in A$ und $b \in B$.
Das verallgemeinert sich zur Menge aller geordneten n–Tupel:

$$A_1 \times A_2 \times \ldots \times A_n = \{(a_1, a_2, \ldots, a_n) | \forall i : \ a_i \in A_i\}$$

Sind die Mengen A_i alle gleich, so schreibt man abkürzend auch A^n für das n–fache kartesische Produkt.

Definition: Eine Untermenge R eines kartesischen Produkts $A \times B$ wird *(binäre) Relation* zwischen A und B genannt. Für $(a,b) \in R$ kann auch aRb geschrieben werden.
Eine Untermenge R eines kartesischen Produkts $A \times A$ wird (binäre) Relation in A genannt.

$\emptyset \subseteq A \times B$ wird *leere Relation* und $A \times B$ wird *Allrelation* zwischen A und B genannt. Die Menge $\{(a,a) \mid a \in A\}$ wird die *identische Relation* in A genannt und kurz mit Id_A bezeichnet.

Zur Darstellung von Relationen sind verschiedene Methoden gebräuchlich:
Darstellungen in Tabellenform (relationale Datenbanken), Diagramme in einem Rechteck und Graphen.

Operationen auf Relationen
a) Sind R und R' Relationen zwischen A und B, dann sind auch die Vereinigung $R \cup R'$, der Durchschnitt $R \cap R'$ sowie das Komplement $\overline{R} = (A \times B) \setminus R$ Relationen zwischen A und B.
b) Die zu einer Relation $R \subseteq A \times B$ *inverse Relation* $R^{-1} \subseteq B \times A$ ist definiert durch $R^{-1} = \{(b,a) \in B \times A \mid (a,b) \in R\}$.
c) Die *Verkettung* $R \circ S$ von zwei Relationen $R \subseteq A \times B$ und $S \subseteq B \times C$ ist definiert durch: $\{(a,c) \in A \times C \mid \exists b \in B : \ (a,b) \in R \wedge (b,c) \in S\}$

Beispiele:

1. Die Vergleichsrelationen $<, \leq, \geq$ und $=$ sind Relationen in den natürlichen Zahlen $\mathbb{N}$. Offensichtlich ist die Vereinigung der Relationen $<$ und $=$ die Relation $\leq$. Das Komplement der Relation $<$ ist die Relation $\geq$. Der Durchschnitt der Relationen $\leq$ und $\geq$ ist die Gleichheitsrelation $=$, die wir auch als $Id_{\mathbb{N}}$ schreiben können.
   Die zu $\leq$ inverse Relation ist $\geq$, die identische Relation ist zu sich selbst invers.

2. Sei M die Menge aller Menschen und $R \subseteq M \times M$ die "Elternrelation", also aRb, falls a Vater oder Mutter von b ist. Dann kann man die inverse Relation R^{-1} sowie die Verkettungen $R \circ R$, $R \circ R^{-1}$ und $R^{-1} \circ R$ wie folgt charakterisieren:

   $aR^{-1}b$, falls a Kind von b ist,

   $aR \circ Rb$, falls a Großvater oder Großmutter von b ist,

   $aR \circ R^{-1}b$, falls a und b ein gemeinsames Kind haben oder falls $a = b$ und a hat ein Kind ,

   $aR^{-1} \circ Rb$, falls $a = b$ oder a und b Geschwister oder Halbgeschwister sind.

Definition (Eigenschaften von Relationen): Sei R eine binäre Relation in A.

- R ist *reflexiv*, falls für jedes $a \in A$ gilt, dass aRa, d.h., $Id_A \subseteq R$.
- R ist *symmetrisch*, falls aus aRb folgt, dass bRa , d.h., $R^{-1} \subseteq R$.
- R ist *transitiv*, falls aus aRb und bRc folgt, dass aRc, d.h., $R \circ R \subseteq R$.
- R ist *antisymmetrisch*, falls aus aRb und bRa die Gleichheit $a = b$ folgt, d.h. $R \cap R^{-1} \subseteq Id_A$.
- R ist *asymmetrisch*, falls aus aRb folgt, dass $(b,a) \notin R$, d.h. $R \cap R^{-1} = \emptyset$.

3.2 Äquivalenzrelationen

Binäre Relationen in einer Grundmenge A setzen jeweils zwei Elemente aus der Grundmenge in Beziehung. Eine Äquivalenzrelation tut dies auch, aber auf eine besondere Art und Weise. Wir werden sehen, dass eine Äquivalenzrelation die Grundmenge in Klassen zerlegt. In jeder Klasse steht jeder mit jedem in Relation, zwischen Elementen aus verschiedenen Klassen gibt es aber keinerlei Beziehungen!

Definition: Eine Relation in einer Menge A wird *Äquivalenzrelation* genannt, wenn sie reflexiv, symmetrisch und transitiv ist.

Definition: Ist $R \subseteq A \times A$ eine Äquivalenzrelation und ist $a \in A$, dann nennt man die Menge $\{x \in A \mid xRa\}$ die *Äquivalenzklasse* von a (bezüglich R).
Sie wird mit $a/_R$ bezeichnet.
Ein Element einer Äquivalenzklasse wird auch *Repräsentant* dieser Klasse genannt.

Beispiele:

1. Für jedes A ist die Allrelation $R = A \times A$ eine Äquivalenzrelation. Wie sehen die Äquivalenzklassen aus? Da jedes Element mit jedem in Relation steht, gilt $a/_R = A$ für jedes $a \in A$.

2. Für jedes A ist die Identität $Id_A = \{(a,a) | a \in A\} \subseteq A \times A$ eine Äquivalenzrelation und jede Äquivalenzklasse besteht nur aus genau einem Element: $a/_{Id_A} = \{a\}$ für jedes $a \in A$.

3. $A = \{1,2,3,4,5\}$ und sei

   $R = \{(1,1),(2,2),(3,3),(4,4),(5,5),(1,3),(3,1),(1,5),(5,1),(3,5),(5,3),(2,4),(4,2)\}$

   R ist reflexiv, symmetrisch und transitiv, also eine Äquivalenzrelation.
   Die Äquivalenzklassen sind $1/_R = 3/_R = 5/_R = \{1,3,5\}, 2/_R = 4/_R = \{2,4\}$.

4. Die semantische Äquivalenz $\equiv$ zwischen Termen über einer Variablenmenge V ist eine Äquivalenzrelation und nach Definition ist für einen Term t die Äquivalenzklasse
   $t/_{\equiv} = \{t' \mid t \equiv t'\}$.

5. Sei M eine endliche Menge und $A = \mathcal{P}(M)$. Wir definieren $R = \{(N,N') | N,N' \in \mathcal{P}(M), |N| = |N'|\}$. Dies ist eine Äquivalenzrelation und die Äquivalenzklasse von N besteht aus allen mit N gleichmächtigen Teilmengen von M.

6. (Wichtig!) Sei $m \in \mathbb{Z}^+$ eine positive ganze Zahl. Jede ganze Zahl a lässt sich eindeutig als Vielfaches von m plus ein Rest aus $\{0,1,2,\ldots,m-1\}$ schreiben.
   Bsp: Sei $m = 5$. Dann ist $13 = 2 \cdot 5 + 3$, $25 = 5 \cdot 5 + 0$ und $-13 = (-3) \cdot 5 + 2$
   Wir sagen "$a \equiv b \mod m$ (a ist kongruent zu b modulo m)" falls a und b denselben Rest beim Teilen durch m lassen. Oft findet man als Notation auch $a \equiv_m b$.
   Äquivalent dazu ist die Bedingung: $m|(a-b)$, das heißt also, m teilt $a-b$ ohne Rest.
   Wir definieren für ein fixes m eine Relation in $\mathbb{Z}$ durch
   $R = \{(a,b) | \ a \equiv b \mod m\}$.
   Dies ist eine Äquivalenzrelation, denn

   - R ist reflexiv: $\forall a \in \mathbb{Z} : m|a-a$
   - R ist symmetrisch: $\forall a,b \in \mathbb{Z} : (\ m|a-b\ ) \Rightarrow (\ m|b-a\ )$
   - R ist transitiv: $\forall a,b,c \in \mathbb{Z} : (\ m|a-b\ ) \wedge (\ m|b-c\ ) \Rightarrow (\ m|a-c\ )$, denn $a-c$ lässt sich schreiben als $a-b+b-c$.

   Wie sehen für $m = 5$ die Äquivalenzklassen aus?
   $0/_R = \{\cdots,-15,-10,-5,0,5,10,15,\cdots\}$
   $1/_R = \{\cdots,-14,-9,-4,1,6,11,16,21,\cdots\}$
   $2/_R = \{\cdots,-13,-8,-3,2,7,12,17,\cdots\}$
   $3/_R = \{\cdots,-12,-7,-2,3,8,13,18,\cdots\}$
   $4/_R = \{\cdots,-11,-6,-1,4,9,14,19,\cdots\}$.
   Gibt es noch andere Äquivalenzklassen? Nein! Genauer gilt das Folgende für eine beliebige Äquivalenzrelation.

Lemma: Sei R eine beliebige Äquivalenzrelation, dann sind zwei Äquivalenzklassen $a/_R$ und $b/_R$ entweder gleich oder disjunkt.
Beweis: Sei $c \in a/_R \cap b/_R$. Wir müssen zeigen, dass dann die Klassen schon gleich sind. Zunächst zeigen wir, dass aRb und bRa gilt. Das ist klar, denn aus aRc und bRc folgt wegen Symmetrie cRb und damit wegen Transitivität aRb. Wieder wegen Symmetrie von R ist auch bRa.
Nun zeigen wir $a/_R \subseteq b/_R$: Sei $d \in a/_R$. Nach Definition ist aRd. Wegen bRa folgt aus der Transitivität bRd, also $d \in b/_R$.
Fehlt noch $b/_R \subseteq a/_R$: Das geht völlig analog. Sei $d \in b/_R$. Nach Definition ist bRd. Wegen aRb folgt aus der Transitivität aRd, also $d \in a/_R$.
Wegen $a/_R \subseteq b/_R$ und $b/_R \subseteq a/_R$ sind die beiden Mengen gleich. □

Satz: Ist $R \subseteq A \times A$ eine Äquivalenzrelation, dann bildet die Menge aller Äquivalenzklassen eine Partition von A. Umgekehrt, ist eine Partition $\{A_i \mid i \in I\}$ von A gegeben, dann ist die durch "aRb genau dann, wenn es ein $i \in I$ gibt, so daß $a \in A_i$ und $b \in A_i$" definierte Relation R eine Äquivalenzrelation.
Beweis: Wegen $a \in a/_R$ (Reflexivität!) für jedes $a \in A$ ist jede Äquivalenzklasse nicht leer. Da $\bigcup_{a \in A} a/_R \supseteq \bigcup_{a \in A} \{a\} = A$ sind zwei der drei Bedingungen einer Partition schon erfüllt. Die letzte Bedingung, dass verschiedene Teile einer Partition disjunkt sein müssen, ist wegen obigen Lemmas erfüllt.
Zweiter Teil des Beweises (Übung). □

Definition: Sei $R \subseteq A \times A$ eine Äquivalenzrelation. Eine Untermenge wird Repräsentantensystem für R genannt, wenn sie aus jeder Äquivalenzklasse genau ein Element enthält.

Für die obigen Beispiele sieht das etwa wie folgt aus: Bei Beispiel 1 der Allrelation ist jedes $\{a\}$ schon vollständiges Repräsentantensystem. In Beispiel 3 ist $\{1,2\}$ aber auch $\{5,4\}$ vollständiges Repräsentantensystem. In Beispiel 6 schließlich ist die Menge $\{0,1,\ldots,m-1\}$ das sogenannte Standard–Repräsentantensystem, $\{1,2,\ldots,m-1,2m\}$ enthält aber auch aus jeder Äquivalenzklasse genau ein Element, ist also ein vollständiges Repräsentantensystem.

Geometrische Interpretation:
Wir haben schon gesehen, dass sich einige Eigenschaften binärer Relationen in A sehr schön an der Matrixdarstellung ablesen lassen:
Reflexivität heißt, die Hauptdiagonale besteht aus Einsen. Symmetrie heißt Symmetrie der Einträge bezüglich der Hauptdiagonale. Wie kann man die Eigenschaft, Äquivalenzrelation zu sein, erkennen?

In obigem Beispiel 3 sieht die (Standard)–Matrixdarstellung wie folgt aus: Man erkennt

Reflexivität und Symmetrie.

	1	2	3	4	5
1	1	0	1	0	1
2	0	1	0	1	0
3	1	0	1	0	1
4	0	1	0	1	0
5	1	0	1	0	1

Ändert man jedoch die Anordnung der Zeilen/Spalten derart, dass Elemente aus ein und derselben Äquivalenzklasse nebeneinanderstehen, so ergibt sich folgendes Bild

	1	3	5	2	4
1	1	1	1	0	0
3	1	1	1	0	0
5	1	1	1	0	0
2	0	0	0	1	1
4	0	0	0	1	1

Jetzt erkennt man den Fakt, dass eine Äquivalenzrelation eingeschränkt auf eine Äquivalenzklasse wie eine Allrelation wirkt, in einer Äquivalenzklasse steht jeder mit jedem in Beziehung, was sich als quadratischer Einserblock widerspiegelt! Äquivalenzrelationen sind Mengen von Paaren, auch hier kann man sich fragen, welche mengentheoretischen Operationen die Eigenschaft Äquivalenzrelation zu sein erhalten. Es gilt der folgende Satz.

Satz: Seien R und R' Äquivalenzrelationen in A.

1. $R \cap R'$ ist eine Äquivalenzrelation in A.
2. $R \circ R'$ ist eine Äquivalenzrelation genau dann, wenn $R \circ R' = R' \circ R$

Beweis:
Wir beweisen nur die erste Aussage, der Beweis der zweiten findet sich z.B. im Meinel/Mundhenk–Buch.
$R \cap R'$ ist reflexiv: Wegen der Reflexivität von R und R' gilt, dass $\forall a \in A : (a,a) \in R \wedge (a,a) \in R'$ also auch $(a,a) \in R \cap R'$ für jedes $a \in A$.
$R \cap R'$ ist symmetrisch: Sei $(a,b) \in R \cap R'$. Das heißt $aRb \wedge aR'b$. Wegen der Symmetrie von R und R' folgt $bRa \wedge bR'a$, also auch $b(R \cap R')a$.
$R \cap R'$ ist transitiv: Aus $(a,b) \in R \cap R'$ und $(b,c) \in R \cap R'$ folgt $(a,b),(b,c) \in R$ also wegen der Transitivität von R auch $(a,c) \in R$ und analog $(a,c) \in R'$, zusammen ergibt dies $(a,c) \in R \cap R'$. □

Anmerkungen:

1. Wie sieht die zu $R \cap R'$ gehörende Partition von A aus? Das ist die Menge aller paarweisen nichtleeren Schnitte von Äquivalenzklasseen der Relation R mit Äquivalenzklassen von R'.
2. Die Vereinigung zweier Äquivalenzrelationen R und R' ist im Allgemeinen keine Äquivalenzrelation. Dazu ist es ausreichend, dass es zwei Äquivalenzklassen

$a/_R$ und $a'/_{R'}$ gibt, die einen nichtleeren Durchschnitt haben, der verschieden von beiden Äquivalenzklasseen ist. Denn dann gibt es Elemente $b \in (a/_R \setminus a'/_{R'})$, $c \in (a/_R \cap a'/_{R'})$ und $d \in (a'/_{R'} \setminus a/_R)$. Für diese gilt: $(b,c),(c,d) \in R \cup R'$ aber $(b,d) \notin R \cup R'$, da sowohl $(b,d) \notin R$ als auch $(b,d) \notin R'$.
Im Fall, dass jede Äquivalenzklasse von R sich darstellen lässt als Vereinigung von Äquivalenzklassen von R' (man sagt, R' ist eine *Verfeinerung* von R), ist dann $R \cup R' = R$ und damit wieder Äquivalenzrelation.

Als den *Abschluss* einer Relation $R \subseteq A \times A$ bezüglich einer Eigenschaft P von Relationen bezeichnet man die kleinste (bzgl. Mengeninklusion) Obermenge von R, die diese Eigenschaft hat.
Hinweis: Man mache sich klar, dass diese Konstruktion nicht für alle Eigenschaften sinnvoll ist. Wenn R etwa selbst nicht antisymmetrisch ist, so sind es auch die Obermengen von R nicht.
Bezüglich der Eigenschaften Reflexivität, Symmetrie und Transitivität sowie der Eigenschaft Äquivalenzrelation kann man jeweils den Abschluss bilden und der ist dann auch eindeutig bestimmt (den tieferen Grund dafür lernen wir später kennen). Es gilt der folgende Satz, der Beweis findet sich im Meinel/Mundhenk–Buch.

Satz: Jede Relation $R \subseteq A \times A$ kann durch die folgenden 3 Schritte zu einer Äquivalenzrelation erweitert werden und dies ist die kleinste Äquivalenzrelation , die R enthält:

1) reflexiver Abschluss: $R_r = R \cup Id_A$
2) symmetr. Abschluss: $R_{rs} = R_r \cup R_r^{-1} = R \cup R^{-1} \cup Id_A$
3) transitiver Abschluss: $R_{rst} = R_{rs} \cup R_{rs} \circ R_{rs} \cup R_{rs} \circ R_{rs} \circ R_{rs} \cup \ldots = \bigcup_{i=1}^{\infty} R_{rs}^i$

wobei R_{rs}^i die i-fache Verkettung von R_{rs} ist.

Achtung: In obigem Satz ist die Reihenfolge der Schritte von Bedeutung! Erst den transitiven, dann den reflexiven und symmetrischen Abschluss zu bilden, führt nicht zum Ziel. Bsp.: $A = \{1,2,3\}, R = \{(1,2),(3,2)\}$.

3.3 Halbordnungsrelationen und totale Ordnungen

Definition: Eine Relation R in einer Menge A, die reflexiv, transitiv und antisymmetrisch ist, wird *Halbordnungsrelation* oder auch *partielle Ordnungsrelation* genannt. Das Paar (A,R) nennt man eine *halb– (partiell) geordnete Menge* oder kurz *poset* (partially ordered set).

Im Zusammenhang mit Ordnungsrelationen benutzen wir häufig die Notation $(A,\leq)$ anstelle von (A,R), um eine beliebige Menge mit einer Halbordnung zu bezeichnen.
Endliche, halbgeordnete Mengen werden oft durch sogenannte *Hasse–Diagramme* dargestellt. Dabei werden die Elemente der Menge als Punkte in der Ebene gezeichnet, wobei direkte Nachfolger jeweils höher als ihre Vorgänger liegen und mit ihnen durch ein Liniensegment verbunden sind. b ist direkter Nachfolger von a falls $a \neq b$, $a \leq b$ und $\forall c : (a \leq c \leq b) \Rightarrow (a = c \vee c = b)$. Für $a \neq b \wedge a \leq b$ schreiben wir kurz $a < b$.
Formal betrachtet beschreibt das Hasse–Diagramm eines Posets $(A,\leq)$ die kleinste Unterrelation von $\leq$, deren reflexiver und transitiver Abschluss wieder $\leq$ ergibt.

Definition: Zwei Elemente a und b einer halbgeordneten Menge $(A, \leq)$ nennt man vergleichbar, falls $a \leq b$ oder $b \leq a$ gilt. Anderenfalls nennt man sie unvergleichbar.
Definition: Eine Halbordnungsrelation $\leq$ in einer Menge A wird totale (oder auch lineare) Ordnungsrelation genannt, wenn jedes Paar von Elementen vergleichbar ist.

Beispiele:

1. Für jede beliebige Menge M ist $(\mathcal{P}(M), \subseteq)$ eine halbgeordnete Menge. Es ist keine totale Ordnung für $|M| > 1$, denn zwei verschiedene einelementige Teilmengen z.B. sind nicht vergleichbar.

2. Die Teilbarkeitsrelation $|$ ist eine Halbordnungsrelation in der Menge der natürlichen Zahlen $\mathbb{N}$. Dies ist keine totale Ordnung, denn z.B. 4 und 18 sind nicht vergleichbar.

3. In der Menge der reellen Zahlen $\mathbb{R}$ ist die Relation $\leq$ eine Halbordnungsrelation, die auch eine totale Ordnung ist.

Bemerkung: Taucht in der Literatur der Begriff "Ordnungsrelation" auf, so ist darunter in der Regel eine "Halbordnungsrelation" zu verstehen.

Zum Schluss die in der Praxis sehr häufig vorkommende *lexikographische Ordnung*.

Definition: Seien $(A_1, \leq_1), \ldots, (A_n, \leq_n)$ Posets. Dann sei $(A_1 \times A_2 \times \ldots \times A_n, \leq)$ das kartesische Produkt mit der Relation $\leq$, die wie folgt definiert ist:
$(a_1, a_2, \ldots, a_n) \leq (b_1, b_2, \ldots, b_n)$ falls $\forall 1 \leq i \leq n : a_i = b_i$ oder
$\exists\ 1 \leq i < n\ \forall\ 1 \leq j \leq i :\ a_j = b_j \wedge a_{i+1} <_{i+1} b_{i+1}$.

Fakt: $(A_1 \times A_2 \times \ldots \times A_n, \leq)$ ist ein Poset (eine totale Ordnung) falls alle $(A_i, \leq_i)$ Posets (totale Ordnungen) sind.

Um Tupel verschiedener endlicher Länge miteinander zu vergleichen (s. z.B. Lexikon), bedient man sich eines einfachen Tricks. Man nimmt das Leerzeichen als Sonderzeichen zu allen Mengen A_i hinzu, wo es jeweils echt kleiner als alle anderen Elemente sein soll. Jetzt kann man annehmen, dass zu vergleichende Tupel gleiche Länge haben, indem man das kürzere am Ende durch Leerzeichen auffüllt.

Schranken in Halbordnungen
Sei $(A, \leq)$ eine Halbordnung und M eine nichtleere Teilmenge von A.

Definitionen:

- $m \in M$ heißt *maximales* Element in M, falls es kein $m' \in M$ gibt mit $m < m'$.
- $m \in M$ heißt *minimales* Element in M, falls es kein $m' \in M$ gibt mit $m' < m$.
- $a \in A$ heißt *obere Schranke* von M in A, falls $m \leq a$ für jedes $m \in M$.
- $a \in A$ heißt *untere Schranke* von M in A, falls $a \leq m$ für jedes $m \in M$.

- $a \in A$ ist *Supremum* von M in A, falls $a \leq a'$ für alle oberen Schranken a' von M.
- $m \in M$ heißt *Maximum* von M, falls für alle $m' \in M$ gilt: $m' \leq m$. Falls das Maximum existiert, ist es auch Supremum von M.
- $a \in A$ ist *Infimum* von M in A, falls $a' \leq a$ für alle unteren Schranken a' von M.
- $m \in M$ heißt *Minimum* von M, falls falls für alle $m' \in M$ gilt: $m \leq m'$. Falls das Minimum existiert, ist es auch Infimum von M.

Beachte, ein M kann mehrere maximale bzw. minimale Elemente haben, jedoch höchstens ein Minimum/Infimum bzw. höchstens ein Maximum/Supremum.
Ein *Kette* in einer Halbordnung $(A, \leq)$ ist eine Teilmenge von verschiedenen Elementen aus A, die bzgl. $\leq$ eine lineare Ordnung bildet. Ein *Antikette* ist eine Menge paarweise unvergleichbarer Elemente.

Satz: Jede endliche Halbordnung hat ein minimales und ein maximales Element.
Beweis: Sei $a_1 < a_2 < \ldots < a_n$ eine Kette maximaler Länge. Dann ist a_1 minimales und a_n maximales Element in der Halbordnung. Wenn das nicht so wäre, könnte man eine noch längere Kette finden! □

Ebenso einfach sieht man wegen der Antisymmetrie einer Halbordnung, dass es in einem Hasse–Diagramm keine gerichteten Zyklen der Form $a_1 < a_2 < \ldots < a_n < a_1$ mit $n > 1$ geben kann.

Definition: Eine totale Ordnung $(A, \preceq)$ heißt *lineare Erweiterung* einer Halbordnung $(A, \leq)$, falls aus $a \leq b$ folgt, dass $a \preceq b$ in der totalen Ordnung gilt.

Satz: Jede endliche Halbordnung $(A, \leq)$ hat eine lineare Erweiterung.
Beweis: (algorithmisch)
Finde ein minimales Element $m \in A$. Setze m als Minimum in die lineare Erweiterung. $(A \setminus \{m\})$ ist wieder eine endliches Poset. Iteriere diesen Prozess bis das Poset leer ist. □

Hinweise:

1. Die lineare Erweiterung ist nur dann eindeutig bestimmt, wenn $(A, \leq)$ selbst schon linear ist und damit seine eigene Erweiterung ist.
2. Das Finden einer linearen Erweiterung (man spricht auch vom *Topologischen Sortieren*) ist ein typisches Scheduling-Problem. Das Poset spiegelt eine Menge von Jobs mit einigen Nebenbedingungen wider: $a < b$ heißt dabei, a muss zeitlich vor b ausgeführt werden. Zu finden ist eine sequentielle Anordnung der Jobs, die all diese Nebenbedingungen berücksichtigt. Die Aufgabe ist genau dann lösbar, wenn es keine gerichteten Zyklen $a < b < a$ gibt, was zu einem deadlock führen würde. Bei Posets kommt dies nicht vor, s. oben.

3.4 Funktionen

Definition: Unter einer *Funktion* (oder *Abbildung*) f von einer Menge A in eine Menge B versteht man eine Zuordnung, bei der jedem Element aus A ein eindeutig bestimmtes Element aus B entspricht. Formal kann f als eine Relation zwischen A und B charakterisiert werden, so daß für jedes $a \in A$ genau ein $b \in B$ existiert mit $a\,f\,b$. Als übliche Schreibweise dafür verwenden wir $f : A \longrightarrow B$ und $f(a) = b$.
Man spricht von einer *partiellen* Funktion, wenn jedem $a \in A$ höchstens ein $b \in B$ zugeordnet ist. Das heißt, es kann auch Argumente geben, bei denen die Funktion nicht definiert ist.

Definition: Ist $f : A \longrightarrow B$ eine Funktion, $M \subseteq A$ und $N \subseteq B$, dann nennt man die Menge
$f(M) = \{y \in B \mid \text{es gibt ein } x \in M \text{ mit } f(x) = y\}$
das *Bild* von M unter f und die Menge
$f^{-1}(N) = \{x \in A \mid f(x) \in N\}$
das *vollständige Urbild* von N unter f.

Definition: Eine Funktion $f : A \longrightarrow B$ heißt *surjektiv (auf B)*, falls jedes Element von B im Bild von A auftritt, d.h. $f(A) = B$.
Eine Funktion $f : A \longrightarrow B$ heißt *injektiv*, falls je zwei verschiedene Elemente aus A auch verschiedene Bilder haben, d.h. wenn aus $f(a) = f(a')$ folgt: $a = a'$.
Eine Funktion wird *bijektiv* genannt, wenn sie injektiv und surjektiv ist.

Beispiel: Wir betrachten die bekannte Sinusfunktion. Als Funktion von den reellen Zahlen in die reellen Zahlen ist $sin : \mathbb{R} \longrightarrow \mathbb{R}$ weder injektiv noch surjektiv. Dagegen ist
$sin : \mathbb{R} \longrightarrow [-1,1]$ eine surjektive Funktion,
$sin : [-\frac{\pi}{2}, \frac{\pi}{2}] \longrightarrow \mathbb{R}$ eine injektive Funktion und
$sin : [-\frac{\pi}{2}, \frac{\pi}{2}] \longrightarrow [-1,1]$ eine bijektive Funktion.

Betrachtet man eine Funktion $f : A \longrightarrow B$ als Relation, dann ist die zu f inverse Relation f^{-1} genau dann eine Funktion, wenn f bijektiv ist. In diesem Fall wird f^{-1} die zu f *inverse Funktion* genannt.

Satz: Sei $f : A \to B$ eine beliebige Funktion und $M_1, M_2 \subseteq A$ sowie $N_1, N_2 \subseteq B$. Dann gilt:

1. $f(M_1 \cup M_2) = f(M_1) \cup f(M_2)$
2. $f(M_1 \cap M_2) \subseteq f(M_1) \cap f(M_2)$
3. $f^{-1}(N_1 \cup N_2) = f^{-1}(N_1) \cup f^{-1}(N_2)$
4. $f^{-1}(N_1 \cap N_2) = f^{-1}(N_1) \cap f^{-1}(N_2)$

Beweis: Wir beweisen nur die zweite Einschaft.
Wir müssen für beliebige $b \in B$ zeigen: $b \in f(M_1 \cap M_2) \Rightarrow b \in f(M_1) \cap f(M_2)$.
Das ist klar, denn: $b \in f(M_1 \cap M_2) \Rightarrow (\exists a \in M_1 \cap M_2 :\ f(a) = b) \Rightarrow b \in f(M_1) \cap f(M_2)$.
In der anderen Richtung gilt nur:
$b \in f(M_1) \cap f(M_2) \Rightarrow (\exists a_1 \in M_1 :\ f(a_1) = b) \wedge (\exists a_2 \in M_2 :\ f(a_2) = b)$.

Wir können nicht schlussfolgern, dass a_1 und a_2 Element von $M_1 \cap M_2$ sind!
Das liefert auch sofort ein Beispiel, dass eine echte Inklusion auftreten kann: Wir wählen $A = \{a_1, a_2\}, M_1 = \{a_1\}, M_2 = \{a_2\}, B = \{b\}$ und $f(a_1) = f(a_2) = b$.
Dann ist $f(M_1 \cap M_2) = f(\emptyset) = \emptyset \subset f(M_1) \cap f(M_2) = \{b\}$ □

Definition: Sind $f : A \longrightarrow B$ und $g : B \longrightarrow C$ Funktionen, dann ist die Relationsverkettung $f \circ g$ eine Funktion von A in C. Sie wird *Verknüpfung* von f mit g genannt und durch $g \circ f : A \longrightarrow C$ bezeichnet, wobei $g \circ f(a) = g(f(a))$ gilt für alle $a \in A$.
Man beachte, daß Relationsverkettungen von links nach rechts und Funktionsverknüpfungen von rechts nach links geschrieben werden. Oft schreiben wir einfach gf statt $g \circ f$, wenn aus dem Kontext klar ist, dass es sich um eine Verknüpfung handelt.

Satz: Die folgenden Fakten ergeben sich als einfache Schlußfolgerungen aus den Definitionen. Seien $f : A \longrightarrow B$ und $g : B \longrightarrow C$ zwei Funktionen, dann gilt:

1. Ist f bijektiv, dann ist $f^{-1}f = Id_A$ und $ff^{-1} = Id_B$
2. f ist genau dann injektiv, wenn eine Funktion $h : B \longrightarrow A$ existiert mit $hf = Id_A$.
3. f ist genau dann surjektiv, wenn eine Funktion $h : B \longrightarrow A$ existiert mit $fh = Id_B$.
4. Sind f und g injektiv, dann ist auch gf injektiv.
5. Sind f und g surjektiv, dann ist auch gf surjektiv.
6. Sind f und g bijektiv, dann ist auch gf bijektiv und es gilt $(gf)^{-1} = f^{-1}g^{-1}$.

Wir beweisen die folgende Charakterisierung surjektiver Funktionen.

Satz: Folgende Bedingungen sind äquivalent.

1. $f : A \to B$ ist surjektiv.
2. $\forall b \in B : \ f^{-1}(b) \neq \emptyset$
3. $\exists g : B \to A : \ f \circ g = Id_B$
4. $\forall C \ \forall r, s : B \to C : \ (r \circ f = s \circ f) \Rightarrow r = s$

Beweis:
Um diese Äquivalenzen zu zeigen reicht es aus, die folgenden Implikationen zu beweisen:

$$(1) \Rightarrow (2), (2) \Rightarrow (3), (3) \Rightarrow (4), (4) \Rightarrow (1)$$

(1) ⇒ (2): $f(A) = B \Rightarrow \forall b \in B \exists a \in A : \ f(a) = b \Rightarrow \forall b \in B : \ f^{-1}(b) \neq \emptyset$

(2) ⇒ (3): Wir wählen für jedes $b \in B$ ein konkretes $a_b \in f^{-1}(b)$. Nun definieren wir $g : B \to A$ durch $g(b) = a_b$. Dafür gilt: $fg(b) = f(g(b)) = f(a_b) = b$ für jedes $b \in B$.

(3) ⇒ (4): Unter Benutzung der Voraussetzung und der Assoziativität der Funktionskomposition gelten auf Funktionsebene folgende Gleichungen:
$r = r \circ Id_B = r \circ (f \circ g) = (r \circ f) \circ g = (s \circ f) \circ g = s \circ (f \circ g) = s \circ Id_B = s$

$\underline{(4) \Rightarrow (1)}$: Wir zeigen die Implikation in der semantisch äquivalenten Form der Kontraposition
$\neg(1) \Rightarrow \neg(4)$.
$\neg(1)$: $\exists b_0 \in B:\ b_0 \notin f(A)$
$\neg(4)$ lässt sich schreiben als: $\exists C \exists r,s:\ (r \circ f = r \circ f) \wedge r \neq s$.
Wir wählen $C = \{0,1\}$. Wir setzen $\forall b \in B:\ r(b) = 0$ und weiterhin $s(b) = 0$ für $b \neq b_0$ und $s(b_0) = 1$.
Für diese Wahl gilt $rf(a) = 0 = sf(a)$ für $a \in A$ aber $r \neq s$ weil $r(b_0) \neq s(b_0)$. □

Analog zeig man für injektive Funktionen:
$f : A \to B$ injektiv $\Leftrightarrow \forall b \in B:\ |f^{-1}(b)| \leq 1 \Leftrightarrow \exists g : B \to A\ \ gf = Id_A$
$\Leftrightarrow \forall D\ \forall r,s : D \to A\ (f \circ r = f \circ s) \Rightarrow r = s$.

Satz: Jede Funktion $f : A \longrightarrow B$ induziert eine Äquivalenzrelation $\sim_f$ durch
$a_1 \sim_f a_2$ genau dann, wenn $f(a_1) = f(a_2)$.

Diese Äquivalenzrelation wird auch *Faserung* von A durch f genannt. Ihre Äquivalenzklassen sind die Urbilder der $b \in B$. Wir werden später sehen, wie man diesen Satz benutzt, um die Anzahle der surjektiven Funktionen zwischen zwei endlichen Mengen zu bestimmen.

3.5 Abzählbarkeit

Als Anwendung von Bijektionen beschäftigen wir uns kurz mit der Formalisierung des Mächtigkeitsbegriffs für Mengen.
Für eine Menge M haben wir mit $|M|$ die Mächtigkeit (die *Kardinalität*) von M bezeichnet, also die Anzahl ihrer Elemente.
Ziel ist es, diesen für den endlichen Fall sehr intuitiven Begriff der Anzahl auch für Mengen mit unendlich vielen Elementen zu fassen. *Unendlich* heißt zunächst einfach nur *nicht endlich*. Bleibt die Frage, ob es auch unendliche Mengen mit unterschiedlicher Mächtigkeit geben kann. Die Antwort wird JA sein.

Definition:(G. Cantor)
Zwei Mengen A und B heißen *gleichmächtig*, wenn es eine bijektive Abbildung f von A auf B gibt.

Definition: Eine unendliche Menge A heißt *abzählbar unendlich* falls sie gleichmächtig mit der Menge $\mathbb{N}$ der natürlichen Zahlen ist. A heißt *abzählbar*, wenn A endlich oder abzählbar unendlich ist.

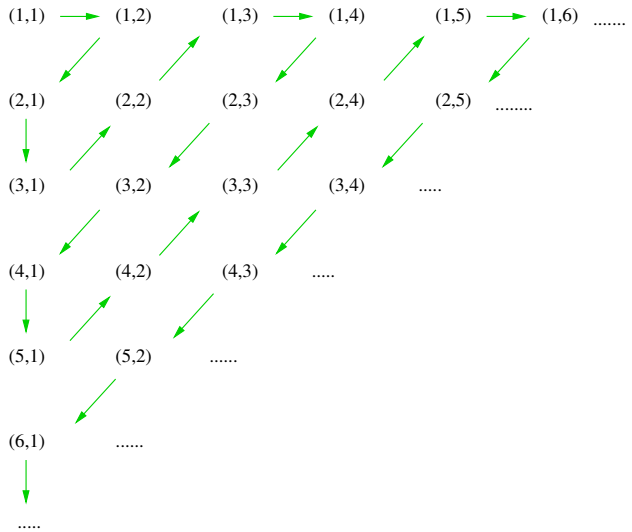
Man stelle sich dazu vor, dass die Bijektion $f : A \to B$ jedem einzelnen Element aus B einen eindeutigen "Namen" eines Elementes aus A zuordnet und verschiedene Elemente aus B verschiedene Namen bekommen. Bei einer abzählbar unendlichen Menge ist diese Zuordnung einfach eine Durchnummerierung der Elemente. Es gibt das 0–te, das erste, das zweite usw. Element.

Beobachtung 1: Ist $f : \mathbb{N} \to A$ eine Surjektion, so ist A abzählbar.
Beweis: Wir definieren eine neue (möglicherweise partielle) Funktion $g : \mathbb{N} \to A$ wie folgt.
$g(0) = f(0)$ und
für $n > 0$ sei $g(n) = f(m)$ wobei $m = \min\{i |\ f(i) \notin \{g(0), g(1), \ldots, g(n-1)\}\}$.
Wenn g partiell ist, so ist A endlich, ansonsten ist g eine Bijektion. □
Beobachtung 2: Ist A abzählbar und $B \subseteq A$, so ist B abzählbar.
Beweis: Das folgt sofort aus Beobachtung 1, wir können es aber auch direkt zeigen. Wenn B unendlich ist und f die Bijektion zwischen $\mathbb{N}$ und A, so betrachten wir die unendliche Folge $n_0, n_1, n_2 \ldots$ von Argumenten von f für die $f(n_i) \in B$. Die Funktion $g : \mathbb{N} \to B$ definiert durch $g(i) = f(n_i)$ ist eine Bijektion. □

Beispiel 1: Die Menge $\mathbb{Z}$ der ganzen Zahlen ist abzählbar.
Die Funktion $f : \mathbb{N} \to \mathbb{Z}$ definiert durch $f(n) = -n/2$ für gerades n und $f(n) = (n+1)/2$ für ungerades n ist eine Bijektion. (Übung)

Man mache sich klar, was diese Funktion leistet. $\mathbb{Z}$ ist disjunkte Vereinigung der natürlichen Zahlen und der negativen ganzen Zahlen. Beide Teile sind offensichtlich abzählbar unendlich. Wir erhalten eine Nummerierung der Elemente aus $\mathbb{Z}$, indem wir die Elemente aus beiden Teilen abwechselnd aufzählen. Es funktioniert zum Beispiel nicht, zunächst alle negativen ganzen Zahlen aufzählen zu wollen und dann den Rest!

Beispiel 2: Die Menge $\mathbb{N}_+ \times \mathbb{N}_+$ ist abzählbar, dabei ist $\mathbb{N}_+$ die Menge der positiven ganzen Zahlen.
Beweis: Die Idee geht wiederum auf Cantor zurück. Sie lässt sich am besten durch ein Bild illustrieren. Wir schreiben die geordneten Paare in Form einer unendlichen Tabelle. Zeile i enthält alle Paare der Form (i,n), Spalte j die Paare (n,j). Die Pfeile geben die Reihenfolge der Nummerierung an. Es ist klar, dass jeder Eintrag in der Tabelle genau einmal und das nach endlich vielen Schritten erreicht wird. □



Beispiel 3: Die Menge der positiven rationalen Zahlen $\mathbb{Q}_+$ ist abzählbar.

Beweis: Wir identifizieren eine positive rationale Zahl p/q, p und q teilerfremd, mit dem geordneten Paar $(p,q) \in \mathbb{N}_+ \times \mathbb{N}_+$. Damit folgt die Behauptung aus Beispiel 2 und obiger Beobachtung 2. □

Korollar: Die Menge $\mathbb{Q}$ aller rationalen Zahlen ist abzählbar.
Beweis: Wir wenden nochmal den Trick aus Beispiel 1 an und zählen die Elemente von $\mathbb{Q}_+$ und $\mathbb{Q}_-$ abwechselnd auf. □

Beispiel 4: Die Menge aller endlichen (!) Teilmengen von $\mathbb{N}$ ist abzählbar.
Beweisidee: Wir ordnen jeder endlichen Teilmenge die Summe der Elemente zu. Das ist eine natürliche Zahl. Umgekehrt, gibt es zu jeder natürlichen Zahl n nur endlich viele Mengen von natürlichen Zahlen mit Summe n.
Wir zählen nun für wachsendes n jeweils alle Mengen mit Summe n auf. □

Beispiel 5: Die Vereinigung von abzählbar vielen abzählbaren Mengen ist abzählbar.
Beweis: Das ist nochmal der Cantor–Trick. Wir erstellen eine unendliche Tabelle. Jede einzelne Zeile enthält die Elemente einer der Mengen und zwar in der Reihenfolge ihrer Aufzählung. Ist die Menge endlich, so fügen wir unendlich viele Leerzeichen an. Alle Elemente der Tabelle werden nun wie in Beispiel 2 nummeriert. □

Fast scheint es so, als könne man jede Menge abzählen. Dem ist nicht so!

Satz: Die Menge aller (!) Untermengen von $\mathbb{N}$ ist nicht abzählbar.
Beweis: (Diagonalisierung nach Cantor)
Wir führen einen Widerspruchsbeweis, das heißt, aus der Annahme, dass $\mathcal{P}(\mathbb{N})$ abzählbar ist, leiten wir einen Widerspruch her.
Angenommen $f : \mathbb{N} \to \mathcal{P}(\mathbb{N})$ ist eine Bijektion. Wir ordnen dieser Bijektion eine konkrete Menge S_f von natürlichen Zahlen zu und werden zeigen, dass diese Menge S_f nicht im Bild von f liegt.

$$S_f = \{n \in \mathbb{N} |\ n \notin f(n)\}$$

Man beachte, jedes $f(n)$ ist eine Menge von natürlichen Zahlen. Und für eine solche Menge kann man fragen, ob eine konkrete Zahl n drin liegt oder nicht. Also ist S_f wohldefiniert.
Da f Bijektion ist nach Annahme und S_f eine Menge von natürlichen Zahlen ist, folgt: $\exists n_0 \in \mathbb{N} :\ f(n_0) = S_f$.
Nochmal, für jede Teilmenge $A \subseteq \mathbb{N}$ und jedes $m \in \mathbb{N}$ ist "$m \in A$" eine Aussage, also entweder wahr oder falsch.
Wir fragen, ob n_0 Element von S_f ist. Die Definition von S_f liefert als Antwort:

$$n_0 \in S_f \Leftrightarrow n_0 \notin S_f$$

Das ist der gesuchte Widerspruch. □

Hinweis: Schaut man sich den Beweis nochmal an, so sieht man, dass es nicht von Bedeutung ist, dass wir von der Potenzmenge der $\mathbb{N}$ sprechen. In der Tat gilt ganz allgemein, dass es für eine beliebige Menge X keine Bijektion zwischen X und der Potenzmenge von X gibt.

Insbesondere gilt also, dass die Potenzmenge der Potenzmenge wiederum mächtiger ist als die einfache Potenzmenge usw.

Definition: Eine Menge, die nicht abzählbar ist, heißt *überabzählbar*.

Gibt es neben der Potenzmenge der natürlichen Zahlen noch andere überabzählbare Mengen?

Satz: Die Menge $\mathbb{R}$ aller reellen Zahlen ist überabzählbar.
Beweis: Wir zeigen, dass das offene Intervall $(0,1)$ schon überabzählbar ist und führen wieder einen Widerspruchsbeweis mit der Cantorschen Diagonalisierungsmethode.
Angenommen $f : \mathbb{N} \to (0,1)$ ist eine Bijektion. Wir können f auch in Form einer unendlichen Wertetabelle darstellen, dabei wird eine reelle Zahl durch einen unendlichen Dezimalbruch dargestellt. Ein solches konkrete f könnte etwa wie folgt aussehen:

f(0)	0	. 1	4	5	8	8	6	4	0	1	.....
f(1)	0	. 4	4	8	2	3	9	4	1	2	.....
f(2)	0	. 8	8	8	2	4	6	6	5	5	.....
f(3)	0	. 8	1	1	0	0	3	3	3	3	.....
f(4)	0	. 4	3	5	5	5	5	5	3	8	.....
f(5)	0	. 3	1	4	4	3	0	3	6	4	.....
f(6)	0	. 9	9	1	6	4	0	2	2	2	.....
	.....						r_f				

Wir ordnen dieser Tabelle eine reelle Zahl $r_f \in (0,1)$ zu, das ist die Zahl auf der Diagonale. Im Beispiel ist $r_f = 0.481530\ldots$.
Nun bilden wir nach den folgenden Regeln aus r_f eine neue Zahl $t_f \in (0,1)$.

1. In jeder Nachkommastelle wird die Ziffer durch die nächstgrößere Ziffer $(mod 10)$ ersetzt, nur die 8 wird durch 7 ersetzt.

2. Ist die erste Nachkommastelle der neuen Zahl jetzt gleich der ersten Nachkommastelle von $f(0)$, so ersetzen wir sie nochmals durch die nächstgrößere Ziffer.

Die Regeln sind eigentlich sehr flexibel, sie müssen nur sicherstellen, dass die entstehende Zahl t_f in jeder Nachkommastelle sich von r_f unterscheidet, nicht auf Periode 9 endet und dass $t_f \neq f(0)$.
Die von uns gewählten Regeln erfüllen diese Forderung. Im Beispiel wäre $t_f = 0.572641\ldots$
Nach Annahme existiert ein n_0 mit $f(n_0) = t_f$. Wir wissen schon $n_0 \neq 0$. Wir schauen uns die n_0–te Nachkommastelle in der n_0–ten Zeile an. Dort steht die n_0–te Nachkommastelle von r_f !!! Und t_f wurde so definiert, dass sich t_f und r_f zumindestens in genau

dieser Nachkommastelle unterscheiden. Widerspruch. □

Anmerkungen:

1. Wir wissen jetzt, dass die Menge $\mathbb{R} \setminus \mathbb{Q}$ der irrationalen Zahlen überabzählbar ist.

2. Eine reelle Zahl heißt *algebraisch*, wenn sie Lösung einer Gleichung der Form $a_0 + a_1 x + \ldots + a_n x^n = 0$ mit allen $a_i \in \mathbb{Z}$ ist. Zum Beispiel ist jede rationale Zahl p/q algebraisch, denn sie ist Lösung von $-p + qx = 0$. Auch die irrationale Zahl $\sqrt{7}$ ist algebraisch, denn sie ist Lösung von $-7 + x^2 = 0$. Man kann (analog zu Beispiel 4 oben) zeigen, dass es nur abzählbar viele solche Gleichungen und damit nur abzählbar viele algebraische irrationale Zahlen gibt.
   Daraus folgt, es gibt überabzählbar viele nichtalgebraische irrationale Zahlen. Diese heißen *transzendent*. Der Nachweis, dass eine konkrete reelle Zahl transzendent ist, ist recht schwer. Prominente Vertreter sind π und die Euler–Konstante e.

3. Die Anzahl $|\mathbb{N}|$ wird oft mit $\aleph_0$ (gesprochen: aleph-null) bezeichnet. $\aleph$ ist der erste Buchstabe des hebräischen Alphabets. Die *Continuumshypothese* sagt, dass es keine Menge X gibt mit $\aleph_0 < |X| < |\mathbb{R}|$. Heute weiß man, dass diese Hypothese aus den Standardaxiomen der Mengenlehre weder beweisbar noch widerlegbar ist.

4 Mathematische Beweise; Vollständige Induktion

4.1 Das Schubfachprinzip von Dirichlet

Das Schubfachprinzip ist ein sehr einfaches kombinatorisches Argument, das es erlaubt, die Existenz von Objekten zu beweisen, ohne diese explizit zu konstruieren.

Satz (Schubfachprinzip): Seien A und B endliche Mengen und $f : A \to B$ eine Funktion.

$$\exists b_0 \; : \; |f^{-1}(b_0)| \geq \lceil |A|/|B| \rceil$$

Beweis: Die Menge A wird durch die nichtleeren Urbilder der Elemente aus B partitioniert, also $\sum_{b \in B} \; |f^{-1}(b)| = |A|$. Sei b_0 ein Element aus B für das $|f^{-1}(b_0)|$ maximal ist.

$$|A| = \sum_{b \in B} |f^{-1}(b)| \leq |B| \cdot |f^{-1}(b_0)|$$

Das liefert $|f^{-1}(b_0)| \geq |A|/|B|$. Da $|f^{-1}(b_0)|$ eine natürliche Zahl ist und $|A|/|B|$ eine i.A. nicht ganze Zahl, kann man $|A|/|B|$ bis zur nächsten ganzen Zahl aufrunden, also $|f^{-1}(b_0)| \geq \lceil |A|/|B| \rceil$ □

Anmerkungen:

1. Das Schubfachprinzip ist nichts weiter als ein Durchschnittsargument. Bildet man das arithmetische Mittel der Größe der Urbilder, so gibt es wenigstens ein Element, dessen Urbild mindestens so groß ist wie der Durchschnitt.
   Merke: Sind $k+1$ verschiedene Sachen in insgesamt k Schubfächern, so gibt es ein Schubfach mit mindestens $\lceil \frac{k+1}{k} \rceil = 2$ Sachen.
   Man beachte, es wird nicht gesagt (und das geht auch gar nicht), um welches Schubfach es sich handelt, und es könnten auch alle $k+1$ in einem Fach liegen.

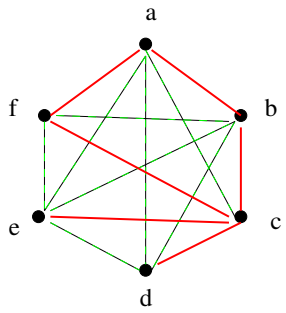
2. Natürlich ist das Schubfachprinzip nur dann nicht trivial anzuwenden, wenn gilt $|A| > |B|$.

3. Oft nennt man das Schubfachprinzip auch *Taubenschlag–Prinzip*, englisch: pigeon hole principle.

Wir werden eine Reihe von Beispielen sehen, in denen das Schubfachprinzip direkt oder auch etwas versteckt zur Anwendung kommt.

Beipiel 1: In einer Gruppe A von 36 Personen haben mindestens 6 Personen am gleichen Wochentag Geburtstag.
Beweis: Sei B die Menge der 7 Wochentage und f die Funktion, die jeder Person denjenigen Wochentag zuordnet, an dem sie Geburtstag hat. Nach dem Schubfachprinzip:
$\exists b_0 : |f^{-1}(b_0)| \geq \lceil \frac{36}{7} \rceil = 6$ □

Beipiel 2: Sei $A = \{a,b,c,d,e,f\}$ eine Menge von 6 Personen, die auf einer Party zusammentreffen. Dann gibt es darunter mindestens 3 Personen, die sich paarweise schon vor der Party kannten, oder sich paarweise nicht kannten.
Beweis: Wir illustrieren die Aussage zunächst graphisch. Wir stellen die Tatsache, dass



sich x und y schon kannten durch eine grüne Kante und das Nichtkennen durch eine rote Kante dar. Die Situation könnte also z.B. so aussehen: In dieser Situation kennen sich b,d und e. Die Aussage lässt sich auch so formulieren: Egal, wie man die 15 Kanten mit den Farben rot und grün einfärbt, es entsteht immer ein einfarbiges Dreieck!
Betrachten wir Person a:
Nach Schubfachprinzip (!) ist sie mit $\lceil \frac{5}{2} \rceil = 3$ der anderen Personen bekannt oder nicht bekannt. Seien dies $x,y,z \in A$.
Fall 1: a ist mit x,y,z bekannt.
Fall 1.1: x,y,z kennen sich paarweise nicht. Fertig, denn sie bilden ein rotes Dreieck.
Fall 1.2: Zwei Personen aus $\{x,y,z\}$ kennen sich. Dann bilden a und diese beiden ein grünes Dreieck.
Fall 2: a ist mit x,y,z nicht bekannt.
In völliger Analogie zu Fall 1, man muss nur "bekannt" und "nicht bekannt" miteinander vertauschen. □

Beispiel 3: (Satz von Erdös–Szekeres) Jede Folge von n^2+1 verschiedenen natürlichen Zahlen enthält eine monotone (d.h. monoton steigend oder monoton fallend) Teilfolge der Länge $n+1$.

Zunächst bemerken wir, dass das Resultat bestmöglich ist, denn wenn wir nur n^2 viele Zahlen nehmen, so gibt es Beipiele, in denen nur eine monotone Teilfolge der Länge n existiert: Die Folge

$$n, n-1, \ldots, 1,\ \ 2n-1, 2n-2, \ldots, n+1,\ \ldots\ , n^2, n^2-1, \ldots (n-1)n+1$$

hat diese Eigenschaft. Sie besteht aus den Zahlen von 1 bis n^2 angeordnet in n Blöcken der Länge n. Alle Elemente eines Blocks sind kleiner als alle Elemente der folgenden Blöcke und jeder Block ist monoton fallend. Damit ist n die maximale Länge einer monotonen Teilfolge.

Beweis: (Widerspruchsbeweis)
Annahme: Es gibt keine monotone Teilfolge der Länge $n+1$, d.h., alle monotonen Teilfolgen haben Länge $\leq n$.
Sei $a_1, a_2, \ldots, a_{n^2+1}$ die Folge. Wir definieren eine Funktion

$$f : \{1,2,\ldots,n^2+1\} \to \{1,2,\ldots,n\} \times \{1,2,\ldots,n\}$$

durch $f(k) = (s_k, f_k)$. Dabei ist s_k die maximale Länge einer monoton steigenden Teilfolge, die bei a_k beginnt, und f_k die maximale Länge einer monoton fallenden Teilfolge, die bei a_k beginnt. Die Funktion ist wohldefiniert, denn nach Annahme sind s_k und f_k beide $\leq n$, und beide sind mindestens 1, denn die einelementige Folge a_k ist sowohl monoton steigend wie fallend.
Nach dem Schubfachprinzip gibt es zwei Indizes $l < r$ mit $f(l) = f(r)$, also $(s_l, f_l) = (s_r, f_r)$. Wir wissen $a_l \neq a_r$.
Fall 1: $a_l < a_r$
Wir setzen vor die in a_r beginnende monoton steigende Folge der Länge s_r das Glied a_l und erhalten $s_l > s_r$, Widerspruch!
Fall 2: $a_l > a_r$
Wir setzen vor die in a_r beginnende monoton fallende Folge der Länge f_r das Glied a_l und erhalten $f_l > f_r$, Widerspruch!
Damit ist unsere Annahme falsch, die Aussage des Satzes also richtig. □

Schaut man sich den Beweis an, so ist es nicht wichtig, dass wir es mit natürlichen Zahlen zu tun haben. Tatsächlich ist der Satz wahr für beliebige linear geordnete Mengen.

Beispiel 4: (Erdös) Sei X eine $(n+1)$–elementige Teilmenge der Menge $\{1, 2, \ldots, 2n-1\}$. Dann gibt es in X zwei verschiedene Elemente a, b mit $a|b$.
Beweis: Zunächst wieder der Hinweis, dass das Resultat bestmöglich ist, denn wählt man nur n Zahlen, so könnte man auch die Menge $X = \{n, n+1, \ldots, 2n-1\}$ nehmen und für diese ist die Aussage falsch!
Wir wenden das Schubfachprinzip an und definieren eine Funktion

$$f : X \to \{1, 3, 5, \ldots, 2n-1\}$$

durch die Bedingung: $f(x)$ sei der größte ungerade Teiler von x. $f(x)$ ist offenbar eine ungerade Zahl zwischen 1 und $2n-1$. Bsp.: $f(6) = 3, f(7) = 7, f(84) = 21$
Nach dem Schubfachprinzip gibt es zwei Zahlen $a, b \in X$ mit $f(a) = f(b)$. Sei $a < b$. Wir wissen $a = 2^i \cdot f(a)$ und $b = 2^j \cdot f(a)$ für irgendwelche $0 \leq i < j$. Und damit ist b Vielfaches von a. □

Beispiel 5
Gegeben sei eine Menge X von 90 höchstens 25–stelligen Dezimalzahlen.
Behauptung: $\exists A, B \subseteq X : A \neq B \wedge \sum_{a \in A} a = \sum_{b \in B} b$
Beweis: Jedes Element $a \in X$ ist kleiner als 10^{25}. Damit ist $\sum_{a \in A} a < 90 \cdot 10^{25} = 0.9 \cdot 10^{27}$.
Wir betrachten die Abbildung $f : \mathcal{P}(X) \to \{0, 1, 2, \ldots, 0.9 \cdot 10^{27}\}$, die jeder Teilmenge von X die Summe ihrer Elemente zuordnet.
Da aber $|\mathcal{P}(X)| = 2^{90} > 10^{27} > 0.9 \cdot 10^{27}$ gibt es nach dem Schubfachprinzip zwei Mengen A, B mit $f(A) = f(B)$. □

Anmerkungen:

1. Natürlich gibt es beliebig große Mengen von ganzen Zahlen, bei denen alle Teilmengen verschiedene Summen haben. $\{2^0, 2^1, 2^2, \ldots 2^{n-1}\}$ ist ein solches Beispiel. Man kann jede Teilmengen davon interpretieren als Binärdarstellung einer Zahl $< 2^n$.

2. Was das Beispiel so interessant macht, ist die Tatsache, dass man trotz des so einfachen mathematischen Beweises kein algorithmisches Hilfsmittel kennt, so ein Paar (A,B) tatsächlich zu **finden**. Natürlich kann man die 90 Zahlen zuerst sortieren, aber das hilft nicht viel und ein brute-force Ansatz des Durchprobierens verbietet sich von selbst: Bei optimistischen 10^9 also rund 2^{30} Paaren pro Sekunde sind das 2^{150} Sekunden und das rechnet man besser nicht in Jahre um...

Beispiel 6: Zum Schluss was Praktisches.
Satz: Eine reelle Zahl r ist rational $\Leftrightarrow$ r hat eine periodische Dezimalbruchdarstellung (Periode $\overline{0}$ ist möglich)
Beweis: Wir können $r \in (0,1)$ annehmen.
($\Rightarrow$): Wir schauen uns das Schulverfahren zur schriftlichen Division an. Sei $r = j/k$. Ab der ersten Nachkommastelle werden Reste aus $\{0,1,\ldots k-1\}$ produziert. Nach höchstens k Schritten kommt als Rest die 0 oder ein Rest wiederholt sich! (Schubfachprinzip)
($\Leftarrow$): Der Beweis dieser Implikation benutzt nicht das Schubfachprinzip!
Sei $r = 0.a_1a_2...a_n\overline{p_1p_2\ldots p_m}$ der periodische Dezimalbruch. Dann ist

$$d = (10^n \cdot r - a_1a_2\ldots a_n)\cdot(10^m-1) \in \mathbb{Z} \quad \text{also} \quad r = (d/(10^m-1) + a_1a_2\ldots a_n)/10^n \in \mathbb{Q}$$

□

4.2 Prinzipielles zu mathematischen Beweisen

In diesem Abschnitt geht es darum, einige grundlegende Beweisstrategien kennenzulernen bzw. zu wiederholen und nochmal die Verbindung zur Logik herzustellen. Da das prinzipielle Verständnis im Mittelpunkt stehen soll, sind die in den Beispielen bewiesenen Aussagen sehr einfach gewählt. Gerade bei diesen scheinbaren Selbstverständlichkeiten wird ein weiteres Problem deutlich: Bevor man an einen Beweis geht, muss man sich klarmachen, was man schon als Basiswissen voraussetzen kann, und was man noch beweisen muss. Oft ist als erster hilfreicher Schritt eine geeignete Formalisierung der Aussage notwendig.

Zunächst einige *goldene Regeln* für schöne Beweise:

- Erkläre, was Du machen willst! Zum Beispiel: Wir führen eine Beweis mit vollständiger Induktion ... Damit ist die prinzipielle Struktur dessen, was folgt, schon klar für den Leser.
- Die verwendeten Argumente sollten linear geordnet sein. Also keine unbewiesenen Fakten benutzen!
- Ein Beweis ist eher ein Essay als reine Rechnung, vermeide auch den exzessiven Gebrauch von Symbolismus.
- Vereinfache die Darstellung so weit wie möglich!

- Wähle sinnvolle Bezeichner! Grundregel (nicht nur bei Beweisen): Ähnliche Sachen werden ähnlich bezeichnet! Das sind dann oft im Alphabet nebeneinanderstehende Buchstaben wie X,Y,Z oder man verwendet Indizes X_1,X_2,X_3 oder auch X',X'',X^* usw. Es gibt dann stillschweigende Vereinbarungen wie der Gebrauch von $f,g,h\ldots$ für Funktionen, $n,m,i,j,k\ldots$ für natürliche Zahlen und Indizes.

- Keine unerlaubten Tricks!!! Formulierungen wie "Offensichtlich gilt...", "Es ist klar, dass...","Der Beweis wird dem Leser überlassen..." erwecken nur Zweifel.

- Führe den Beweis zu Ende, was für den Autor klar ist, muss für den Leser noch lange nicht klar sein.

Viele mathematische Sätze haben die Form einer Implikation, sie sagen, dass aus einer bestimmten Voraussetzung p eine Behauptung q folgt.

Zum Beweis kann man verschiedene Techniken anwenden. Basis für die Gültigkeit solcher Beweise sind einige einfache Äquivalenzen und Implikationen, die man leicht mit der Wahrheitstafelmethode nachweisen kann.

Die naheliegendste Technik ist der **direkte Beweis**, der darauf beruht, die Implikation $p \Rightarrow q$ in mehrere elementare Teilschritte zu zerlegen, wobei man die folgende Tautologie nutzt: $((p \Rightarrow r) \wedge (r \Rightarrow q)) \Rightarrow (p \Rightarrow q)$. Grundlage dafür ist der *modus ponens*, das ist die Tautologie $(p \wedge (p \Rightarrow r)) \Rightarrow r$.
Wie das folgende Beispiel zeigt, bewegt man sich bei der Begründung der Elementarschritte in einem System, das sich auf einigen Axiomen (Grundannahmen) aufbaut und in dem man auf bereits bewiesene Tatsachen zurückgreifen kann.
(Hinweis: Axiomensystem von Zermelo–Fraenkel, Gödels Unvollständigkeitssatz)

Satz: Ist eine natürliche Zahl n durch 6 teilbar, so ist ihr Quadrat durch 9 teilbar.
Beweis: Die Idee ist offensichtlich – ist n durch 6 teilbar, so kann man den Faktor 6 und damit auch den Faktor 3 von n abspalten. Folglich kann man den Faktor 3 mindestens zwei mal von n^2 abspalten. Wenn wir diese Idee etwas formaler umsetzen wollen, müssen wir mit der Definition von Teilbarkeit beginnen:
$n \in \mathbb{N}$ ist durch $k \in \mathbb{N}$ teilbar, falls ein $l \in \mathbb{N}$ existiert, so dass $n = k \cdot l$.
Damit kann man die Voraussetzung des Satzes durch eine einfache Formel ausdrücken und die folgende Beweiskette bilden:

n ist durch 6 teilbar		Hypothese
$\exists l \in \mathbb{N}$	$n = 6 \cdot l$	Teilbarkeitsdefinition
$\exists l \in \mathbb{N}$	$n = (3 \cdot 2) \cdot l$	$6 = 3 \cdot 2$
$\exists l \in \mathbb{N}$	$n^2 = ((3 \cdot 2) \cdot l)((3 \cdot 2) \cdot l)$	Quadrieren
$\exists l \in \mathbb{N}$	$n^2 = (3 \cdot 3)((2 \cdot 2) \cdot (l \cdot l))$	Muliplikation ist assoziativ und kommutativ
$\exists l \in \mathbb{N}$	$n^2 = 9 \cdot (4 \cdot l^2)$	$3 \cdot 3 = 9$ und $2 \cdot 2 = 4$
$\exists l' \in \mathbb{N}$	$n^2 = 9 \cdot l'$	$l' = 4l^2$
n^2 ist durch 9 teilbar		Teilbarkeitsdefinition $\square$

Genau betrachtet haben wir beim Schritt von der vierten zur fünften Zeile sogar mehrere Elementarschritte zu einem Schritt zusammengefasst.

Manchmal ist es schwierig, den Beweis direkt zu führen. Als Alternativen bieten sich indirekte Beweise durch **Kontraposition** oder in der Form von **Widerspruchs-Beweisen** an. Beim *Beweis durch Kontraposition* wird anstelle von $p \Rightarrow q$ die logisch äquivalente Aussage $\neg q \Rightarrow \neg p$ bewiesen. Beim Widerspruchs-Beweis wird anstelle von $p \Rightarrow q$ die logisch äquivalente Aussage $(p \wedge \neg q) \Rightarrow 0$ bewiesen.
Man beachte, dass eine Aussage p äquivalent ist zu $1 \Rightarrow p$ und damit auch zu $\neg p \Rightarrow 0$.

Beispiel: Wir beweisen durch Kontraposition die folgende Aussage über ganze Zahlen: *"Ist a^2 ungerade, so ist auch a ungerade"*.
Beweis: Da die Negation von *"ungerade sein"* die Eigenschaft *"gerade sein"* ist, lautet die Kontraposition *"Ist a gerade, so ist auch a^2 gerade"*. Und dafür gibt es einen einfachen direkten Beweis:
Ist a gerade, so gibt es eine ganze Zahl b mit $a = 2b$. Folglich ist $a^2 = (2b)^2 = 2 \cdot (2b^2)$ und somit ist a^2 gerade. □

Häufig ist es notwendig, verschiedene Fälle zu analysieren. Das dabei verwendete logische Prinzip ist Äquivalenz der Aussagen $p \Rightarrow q$ und $(p \wedge r \Rightarrow q) \wedge (p \wedge \neg r \Rightarrow q)$ für ein beliebig gewähltes r, wir unterscheiden also die Fälle r und $\neg r$. Man kann diese Fallunterscheidung auch noch feiner machen und den Fall r etwa aufspalten in $r \wedge t$ und $r \wedge \neg t$.

Beispiel: Wir beweisen durch Fallunterscheidung, dass für jede Primzahl $p \geq 5$ die Zahl $p^2 - 1$ durch 24 teilbar ist.
Beweis: Zuerst formen wir $p^2 - 1$ in $(p+1)(p-1)$ um und beobachten, dass von drei aufeinanderfolgenden ganzen Zahlen genau eine durch 3 teilbar ist. Da $p > 3$ und Primzahl ist, muss $p-1$ oder $p+1$ und damit auch $p^2 - 1$ durch 3 teilbar sein. Bleibt zu zeigen, dass $p^2 - 1$ durch 8 teilbar ist. Da p ungerade ist, sind sowohl $p-1$ als auch $p+1$ gerade und damit ist $p^2 - 1$ durch 4 teilbar. Den noch fehlenden Faktor 2 kann man durch Fallunterscheidung nachweisen:
1. Fall: Ist $p-1$ durch 4 teilbar, so ist $p - 1 = 4k$ und $p + 1 = 4k + 2 = 2(2k+1)$ und damit $p^2 - 1 = 8k(2k+1)$ für eine natürliche Zahl k.
2. Fall: Ist $p-1$ nicht durch 4 teilbar, so hat es die Form $4m + 2 = 2(2m+1)$ für eine natürliche Zahl m und folglich ist $p + 1 = 4m + 4 = 4(m+1)$. Damit erhalten wir $p^2 - 1 = 8(2m+1)(m+1)$. □

4.3 Natürliche Zahlen und das Prinzip der vollständige Induktion

Alle aus der Schulmathematik bekannten Aussagen über natürliche Zahlen können aus einigen wenigen Grundannahmen, den Peano'schen Axiomen, abgeleitet werden:

1. Axiom: 0 ist eine natürliche Zahl.
2. Axiom: Jede natürliche Zahl n hat einen eindeutigen Nachfolger $S(n)$, der auch eine natürliche Zahl ist.
3. Axiom: Aus $S(n) = S(m)$ folgt $n = m$.
4. Axiom: 0 ist kein Nachfolger einer natürlichen Zahl.
5. Axiom: Jede Menge X, die 0 enthält und für die gilt, dass aus $n \in X$ auch $S(n) \in X$ folgt, enthält alle natürlichen Zahlen.

Achtung: Wir schreiben für den Nachfolger $S(n)$ auch $n+1$, aber das ist als symbolische

Schreibweise und nicht als Anwendung der Operation Addition zu verstehen. Im Gegenteil, wie die folgenden Betrachtungen zeigen, kann die Addition durch Anwendung der Nachfolgerfunktion rekursiv definiert werden. (vgl. ALP-I-Vorlesung)

Konsequenz 1: Man kann Funktionen $f : \mathbb{N} \to A$ definieren, indem man $f(0)$ festlegt und $f(S(n))$ auf $f(n)$ zurückführt. Dieses Prinzip der Definition von Funktionen nennt man *Rekursion*.

Beispiel: Um die Addition von natürlichen Zahlen einzuführen, definieren wir für jede fest gewählte Zahl m die Funktion $m+ : \mathbb{N} \to \mathbb{N}$, die jedem n aus dem Definitionsbereich die Summe $m+n$ zuordnen soll. Diese Funktion hat die folgende rekursive Definition: $m+(0) := m$ und $m+(S(n)) := S(m+n)$. Das entspricht den Regeln $m+0 := m$ und $m+(n+1) := (m+n)+1$.
Analog kann man die Multiplikation durch $m\cdot : \mathbb{N} \longrightarrow \mathbb{N}$ mit $m\cdot(0) := 0$ und $m\cdot(S(n)) := (m\cdot(n))+m$ definieren, was den Regeln $m\cdot 0 := 0$ und $m\cdot(n+1) := (m\cdot n)+m$ entspricht.

Konsequenz 2: Man kann allgemeine Aussagen über natürliche Zahlen nach dem folgenden Schema beweisen. Eine Aussageform $P(x)$ über dem Bereich der natürlichen Zahlen ist wahr für alle natürlichen Zahlen, wenn sie die folgenden zwei Bedingungen erfüllt:

1. $P(0)$ ist wahr.
2. Für beliebige $n \in \mathbb{N}$ gilt: Ist $P(n)$ wahr, dann ist auch $P(n+1)$ wahr.

Dieses Beweisprinzip nennt man **vollständige Induktion**.
Die erste Bedingung wird *Induktionsanfang* oder *Induktionsbasis*, die zweite Bedingung *Induktionsschluss* genannt. Dabei heißt $P(n)$ *Induktionsvoraussetzung* oder *Induktionsannahme* und $P(n+1)$ *Induktionsbehauptung*.
Beweis: Sei $W \subseteq \mathbb{N}$ die Menge der natürlichen Zahlen, für die $P(n)$ wahr ist. Wegen des Induktionsanfangs ist $0 \in W$. Der Induktionsschritt zeigt, dass falls $n \in W$ gilt, auch $n+1 \in W$. Nach dem 5. Peanoschen Axiom ist $\mathbb{N} \subseteq W$, also $W = \mathbb{N}$. □

Es folgen Beispiele für Aussagen, die man mit vollständiger Induktion beweisen kann:

Beispiel 1: Für jede natürliche Zahl $n \geq 0$ ist die Summe der ungeraden Zahlen von 0 bis $2n+1$ gleich $(n+1)^2$. Es gilt also:

$$\forall n \in \mathbb{N} : \sum_{i=0}^{n} (2i+1) = (n+1)^2$$

Beweis:
Wir führen einen Beweis mit vollständiger Induktion.
Sei $P(n)$ die Aussage $\sum_{i=0}^{n}(2i+1) = (n+1)^2$.
Induktionsanfang: $P(0)$ gilt, denn $\sum_{i=0}^{0}(2i+1) = 1 = (0+1)^2$
Induktionsschritt: Sei n eine beliebige natürliche Zahl und nehmen wir an, dass $P(n)$ gilt. Wir zeigen, dass auch $P(n+1)$ gilt.
$\sum_{i=0}^{n+1}(2i+1) = \sum_{i=0}^{n}(2i+1) + (2(n+1)+1)$

Wir wenden auf den ersten Teil der Summe die Induktionsvoraussetzung an und erhalten durch Vereinfachen: $\sum_{i=0}^{n}(2i+1)+\ (2(n+1)+1) = (n+1)^2+2n+3 =$
$= n^2+2n+1+2n+3 = n^2+4n+4 = (n+2)^2$
Dies zeigt die Richtigkeit von $P(n+1)$ unter der Annahme der Richtigkeit von $P(n)$ und nach dem Prinzip der vollständigen Induktion haben wir die Aussage für jedes $n \in \mathbb{N}$ bewiesen. □

Beispiel 2: Für beliebige reelle Zahlen a und $r \neq 1$ und für jede natürliche Zahl n gilt

$$\sum_{i=0}^{n} ar^i = \frac{ar^{n+1}-a}{r-1}$$

Beweis: Übung □

Zwei Varianten des Induktionsprinzips werden häufig verwendet:

1. Wird die Induktionsbasis nicht für $n=0$ sondern für einen anderen festen Anfangswert $k>0$ bewiesen und zeigt man außerdem $\forall n \geq k:\ P(n) \Rightarrow P(n+1)$, so gilt die Aussage für alle natürlichen Zahlen $n \geq k$.

2. Beim Induktionsschritt ist es erlaubt, nicht nur auf $P(n)$, sondern auf beliebige kleinere Zahlen zurückzugreifen, d.h. an Stelle von $P(n) \Rightarrow P(n+1)$ zeigt man $P(k) \wedge P(k+1) \wedge \ldots \wedge P(n) \Rightarrow P(n+1)$, wobei k der Anfangswert aus der Induktionsbasis ist. Dieses Prinzip wird *verallgemeinerte vollständige Induktion* genannt.

Beispiel 3: Jede natürliche Zahl $n \geq 2$ kann man als Produkt von Primzahlen darstellen.
Beweis: Wir führen einen Beweis mittels verallgemeinerter vollständiger Induktion. Sei $P(n)$ die Aussage, dass sich n als Produkt von Primzahlen schreiben lässt. (Achtung: Das Produkt kann auch nur aus einem Faktor bestehen.)
Induktionsanfang: $P(2)$ gilt, denn $2=2$ ist in der geforderten Produktform.
Induktionsschritt: Sei n eine beliebige natürliche Zahl und nehmen wir an, dass $P(2) \wedge P(3) \wedge \ldots \wedge P(n)$ gilt. Wir zeigen, dass dann die Aussage $P(n+1)$ gilt.
Wir führen eine Fallunterscheidung durch.
Fall 1: $n+1$ ist Primzahl. Dann ist die Zahl selbst die gesuchte Faktorisierung.
Fall 2: $n+1$ ist keine Primzahl. Das heißt:$\exists k,l \in \mathbb{N}: 1<k,l<n+1 \wedge n+1 = k \cdot l$.
Nach Annahme gibt es für k und für l Primzahlfaktorisierungen:

$$k = p_1 \cdot \ldots \cdot p_{m_k} \ \text{ und } \ l = q_1 \cdot \ldots \cdot q_{m_l}$$

wobei alle p_i und q_j Primzahlen sind. Das liefert aber sofort eine Faktorisierung für $n+1$:

$$n+1 = k \cdot l = p_1 \cdot \ldots \cdot p_{m_k} \cdot q_1 \cdot \ldots \cdot q_{m_l}$$

Nach dem Prinzip der vollständigen Induktion ist damit die Aussage für alle natürlichen Zahlen ≥ 2 bewiesen. □

Manchmal ist es hilfreich, im Induktionsanfang die Aussage für mehrere Werte zu beweisen.

Beispiel 4: Jede natürliche Zahl ≥ 12 lässt sich als Summe schreiben, in der alle Summanden 4 oder 5 sind. Formal:

$$\forall n \in \mathbb{N}, n \geq 12 \exists k, l \in \mathbb{N} : n = k \cdot 4 + l \cdot 5$$

Beweis: Wir führen einen Beweis mittels verallgemeinerter vollständiger Induktion.
Induktionsanfang: Die Aussagen $P(12), P(13), P(14), P(15)$ gelten.
Induktionsschritt: Wir zeigen für ein beliebiges $n+1 \geq 16$, dass die Aussage $P(n+1)$ aus $P(12) \wedge P(13) \wedge \ldots \wedge P(n)$ folgt.
Das ist sofort klar, wenn man sich die Aussage $P(n+1-4)$ anschaut. Denn $n-3 \geq 12$ und damit gilt nach Annahme $n-3 = k \cdot 4 + l \cdot 5$ für irgendwelche $k, l \in \mathbb{N}$. Also ist dann $n+1 = (k+1) \cdot 4 + l \cdot 5$ und nach dem Prinzip der vollständigen Induktion ist damit die Aussage für alle natürlichen Zahlen ≥ 12 bewiesen. □

Zum Schluss das Beispiel eines falschen(!) Beweises, das illustriert, dass man im Induktionsschritt die Implikation $P(n) \Rightarrow P(n+1)$ tatsächlich für alle n zeigen muss.

Beispiel 5: In jeder Menge von $n > 0$ Menschen haben alle das gleiche Geschlecht.
Beweis:Wir führen einen Beweis mittels vollständiger Induktion.
Induktionsanfang: Für $n = 1$ ist die Aussage tatsächlich richtig!
Induktionsschritt: Nehmen wir an, für ein beliebiges n gilt $P(n)$, und betrachten wir eine $(n+1)$–elementige Menschenmenge $M = \{m_1, m_2, \ldots, m_{n+1}\}$.
Wir bilden zwei n–elementige Menschenmengen $M_1 = \{m_2, m_3, \ldots, m_{n+1}\}$ und $M_2 = \{m_1, m_2, \ldots, m_n\}$. Nach Induktionsannahme haben jeweils in M_1 und in M_2 alle dasselbe Geschlecht. Aber die Menschen in $M_1 \cap M_2$ gehören zu beiden, also haben in der Tat alle $n+1$ Menschen dasselbe Geschlecht!!

Was ist falsch?: Man mache sich klar, dass die Argumentation im Induktionsschritt nicht für $n = 1$ funktioniert, denn dann ist $M_1 \cap M_2$ leer. Für größere n funktioniert es, aber dann findet man keinen passenden Induktionsanker! Also haben nicht in jeder Menschenmenge alle das gleiche Geschlecht und das ist gut so...

Man kann mit vollständiger Induktion nicht nur Aussagen über natürliche Zahlen beweisen sondern auch allgemein über induktiv definierte Objekte. Diesen kann man nämlich eine natürliche Zahl als Parameter zuordnen, der sozusagen misst, wie komplex ein gewisses Objekt ist. Prominente Beispiele sind Boolesche Terme mit dem Rang als Parameter oder aber Listen in Haskell. Hier ist die Länge der Parameter. Es gibt die leere Liste `[]` mit Länge 0 oder aber die Liste hat die Form `x:xs` und Länge $n+1$, wenn n die Länge von `xs` ist.
Definiert man Funktionen für solche Objekte, dann sind diese in der Regel rekursiv und beweist man Eigenschaften für solche Objekte bzw. Funktionen darauf, so ist vollständige Induktion das Mittel der Wahl! Man spricht dann von **struktureller Induktion**. Als Beispiel folgender einfacher Sachverhalt.

Satz: Jeder Boolesche Term über der Variablenmenge X lässt sich semantisch äquivalent mit den Junktoren $\{\neg, \vee\}$ ausdrücken.

Beweis: (Strukturelle Induktion über den Formelrang)
Sei Φ eine Formel und $rg(\Phi)$ ihr Rang.
Induktionsanfang: $rg(\Phi) = 0$
Das heißt $\Phi = x$ für $x \in X$ oder $\Phi = 0$ oder $\Phi = 1$. Da $1 \equiv x \vee \neg x$ und $0 \equiv \neg(x \vee \neg x)$ ist die Aussage für Formeln vom Rang 0 wahr.
Induktionsschritt: Sei die Behauptung richtig für alle Formeln vom Rang $\leq n$. Sei Φ eine Formel vom Rang $n+1$. Wir machen eine Fallunterscheidung nach der möglichen Gestalt von Φ.
Fall 1: $\Phi = \neg\Phi_1$ und $rg(\Phi_1) = n$
Dann gibt es nach Voraussetzung ein Φ_1' über $\{\neg, \vee\}$ mit $\Phi_1 \equiv \Phi_1'$. Also ist $\Phi \equiv \neg\Phi_1'$.
Fall 2: $\Phi = \Phi_1 \vee \Phi_2$ und $rg(\Phi_1) \leq n$, $rg(\Phi_2) \leq n$.
Damit gibt es nach Annahme Terme $\Phi_1' \equiv \Phi_1$ und $\Phi_2' \equiv \Phi_2$ jeweils über den Junktoren $\{\neg, \vee\}$. Damit ist auch $\Phi \equiv \Phi_1' \vee \Phi_2'$ von der gewünschten Gestalt.
Fall 3: $\Phi = \Phi_1 \wedge \Phi_2$ und $rg(\Phi_1) \leq n$, $rg(\Phi_2) \leq n$.
Damit gibt es nach Annahme Terme $\Phi_1' \equiv \Phi_1$ und $\Phi_2' \equiv \Phi_2$ jeweils über den Junktoren $\{\neg, \vee\}$.
Für Φ können wir mit der deMorganschen Regel schreiben:
$\Phi \equiv \neg(\neg\Phi_1' \vee \neg\Phi_2')$.
Nach dem Prinzip der vollständigen Induktion gilt somit die Aussage für jeden Booleschen Term. $\square$

5 Kombinatorik

5.1 Abzählen I

Die Abzählung von endlichen Mengen ist das klassische Thema der Kombinatorik. Dabei wird eine unendliche Familie $\{A_n \mid n \in I\}$ von endlichen Mengen A_n betrachtet, wobei n eine Indexmenge I durchläuft (in der Regel die natürlichen Zahlen). Zu bestimmen ist die *Zählfunktion* $f : I \longrightarrow \mathbb{N}$ mit $f(n) = |A_n|$ für alle $n \in I$. Hier und im Folgenden bezeichnet $|M|$ die Anzahl der Elemente (Mächtigkeit) einer Menge M, falls diese endlich ist, und ∞ anderenfalls.
Oft werden die Mengen A_n aus einer gegebenen n–elementige Menge M (auch kurz n–*Menge* genannt) durch einfache strukturelle oder kombinatorische Bedingungen abgeleitet, z.B. die Menge $S(M)$ aller Permutationen von M (das sind bijektive Abbildungen von M nach M), die Menge $\mathcal{P}(M)$ aller Untermengen oder die Menge $\binom{M}{k}$ aller k–Untermengen von M. Im letzten Fall ist die Zählfunktion von den zwei Größen n und k abhängig. Ziel ist es, eine möglichst kurze, geschlossene Formel für $f(n)$ zu finden.

Die meisten Aufgabenstellungen dieser Art lassen sich (auch wenn die Lösung teilweise enormen technischen Aufwand erfordert) auf Grundregeln zurückführen. Dazu gehören:

1. **Gleichheitsregel:** Existiert eine bijektive Abbildung zwischen zwei Mengen S und T, so gilt $|S| = |T|$

2. **Summenregel:** Ist S die Vereinigung von paarweise disjunkten, endlichen Mengen $S_1, S_2, \ldots, S_t$, dann gilt $|S| = \sum_{i=1}^{t} |S_i|$.

3. **Produktregel:** Ist S das Kartesische Produkt der endlichen Mengen $S_1, S_2, \ldots, S_t$, dann gilt $|S| = \prod_{i=1}^{t} |S_i|$.

Dabei ist die Gleichheitsregel nicht anderes als die Definition von Gleichmächtigkeit, die Summen– und Produktregel lassen sich sehr einfach mit vollständiger Induktion beweisen.
Als Beispiel der Anwendung der Regeln ist hier nochmal ein kompletter Beweis mit vollständiger Induktion nach n für den

Satz Die Potenzmenge einer n–elementigen Menge hat 2^n Elemente.
Beweis:
Induktionsanfang: Für $n = 0$ ist $M = \emptyset$, $\mathcal{P}(M) = \{\emptyset\}$ und folglich $|\mathcal{P}(M)| = 1 = 2^0$
Induktionsschritt von n nach $n+1$: Sei M eine $(n+1)$–Menge und a ein fest gewähltes Element aus M. Wir zerlegen $\mathcal{P}(M)$ in zwei disjunkte Teilmengen $S_1 = \{A \subseteq M \mid a \in A\}$ und $S_2 = \{A \subseteq M \mid a \notin A\}$. Offensichtlich ist $S_2 = \mathcal{P}(M \setminus \{a\})$ die Potenzmenge einer n–Menge und nach Induktionsvorausetzung gilt $|S_2| = 2^n$. Wir konstruieren eine Bijektion zwischen S_1 und S_2, indem wir jeder Menge $A \in S_1$ die Menge $B = A \setminus \{a\} \in S_2$ zuordnen. Man erkennt die Bijektivität dieser Abbildung daran, dass eine Umkehrabbildung existiert, die jedem $B \in S_2$ die Menge $A = B \cup \{a\} \in S_1$ zuordnet. Aus der Gleichheitsregel folgt $|S_1| = |S_2|$ und aus der Summenregel $|\mathcal{P}(M)| = |S_1| + |S_2| = 2|S_2| = 2 \cdot 2^n = 2^{n+1}$.□

Da die Summenregel sich auf disjunkte Mengen bezieht bleibt die Frage, was passiert im allgemeinen Fall.

Satz: (Inklusions–Exklusionsprinzip)
Für beliebige endliche Mengen A und B gilt

$$|A \cup B| = |A| + |B| - |A \cap B|$$

Beweis: Wir schreiben $A \cup B$ als Vereinigung von disjunkten Mengen.

$$A \cup B = (A \setminus B) \cup (B \setminus A) \cup (A \cap B) \quad (*)$$

Nach der Summenregel gilt:

$$|A \cup B| = |A \setminus B| + |B \setminus A| + |A \cap B|$$

Außerdem haben wir wieder mit der Summenregel:

$$|A| = |A \setminus B| + |A \cap B| \text{ und } |B| = |B \setminus A| + |A \cap B|$$

Nach Umstellen und Einsetzen in $(*)$ ergibt sich die Behauptung. □

Beispiel: Was ist die Anzahl n der Binärwörter der Länge 11, die 001 als Präfix oder 01 als Suffix haben?
Antwort: Es gibt 2^8 Wörter mit Präfix 001, es gibt 2^9 Wörter mit Suffix 11 und 2^6 Wörter die 001 als Präfix und 11 als Suffix haben. Nach dem Inklusions-Exklusionsprinzip ist:

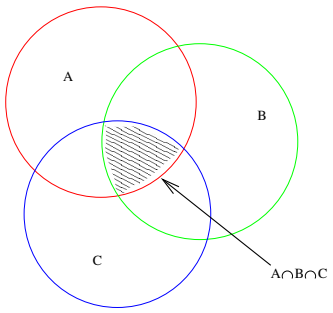
$$n = 2^8 + 2^9 - 2^6 = 704$$

Was ist die Mächtigkeit der Vereinigung dreier Mengen A, B, C?

Satz: Es gilt für endliche Mengen A, B, C:

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

Beweis: Intuitiv sollte die Formel klar sein, vgl. Abbildung. Der gemeinsame Durchschnitt $|A \cap B \cap C|$ wird zunächst bei $|A| + |B| + |C|$ dreimal addiert, bei den paarweisen Durchschnitten dann insgesamt dreimal subtrahiert, also muss er noch einmal addiert werden.



Formal kann man es wie folgt zeigen:
$|A \cup B \cup C| = |A \cup B| + |C| - |(A \cup B) \cap C| = |A \cup B| + |C| - |(A \cap C) \cup (B \cap C)| =$

$= |A \cup B| + |C| - |A \cap C| - |B \cap C| + |(A \cap C) \cap (B \cap C)| =$
$= |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$
Wir haben dabei mehrmals das Inklusions–Exklusionsprinzip für zwei Mengen angewendet und die Distributivität von $\cap$ und $\cup$ benutzt. $\square$

Für eine beliebige endliche Anzahl von endlichen Mengen lässt sich das Inklusions–Exklusions–Prinzip schreiben als :

$$|\bigcup_{i=1}^{k} A_i| = \sum_{1 \le i \le k} |A_i| - (\sum_{1 \le i_1 < i_2 \le k} |A_{i_1} \cap A_{i_2}|) + (\sum_{1 \le i_1 < i_2 < i_3 \le k} |A_{i_1} \cap A_{i_2} \cap A_{i_3}|) - \ldots$$
$$+ \ldots\ldots + (-1)^{k+1} |\bigcap_{i=1}^{k} A_i|$$

Beispiel: Wieviele positive ganze Zahlen ≤ 3000 sind durch 2,3 oder 5 teilbar?
Antwort: Sei $D_i = \{n \in \mathbb{N} | 1 \le n \le 3000, i|n\}$. Gesucht ist also $|D_2 \cup D_3 \cup D_5|$.
Es gilt $|D_2| = 1500, |D_3| = 1000, |D_5| = 600$.
Weiterhin $|D_2 \cap D_3| = |D_6| = 500, |D_2 \cap D_5| = 300, |D_3 \cap D_5| = 200$ und schließlich $|D_2 \cap D_3 \cap D_5| = 100$.
Nach dem Inklusions–Exklusions–Prinzip gilt: $|D_2 \cup D_3 \cup D_5| = 1500 + 1000 + 600 - 500 - 300 - 200 + 100 = 2200$.

Definition: Eine Permutation einer n–Menge A ist eine Bijektion $\pi : \{1, 2, \ldots, n\} \to A$. Das ist nichts anderes als eine geordnete Aufzählung der n Elemente $a_1, a_2, \ldots, a_n$ von A in der Reihenfolge $a_{\pi(1)} a_{\pi(2)} \ldots a_{\pi(n)}$.

Beispiel: Für $A = \{a, b, c\}$ ist $\{abc, acb, bac, bca, cab, cba\}$ die Menge aller Permutationen.

Satz: Es gibt $n!$ viele Permutationen einer n–elementigen Menge A.
Beweis: Vollständige Induktion nach n.
Induktionsanfang: $n = 1$ Klar, weil $1! = 1$.
Induktionsschritt: Angenommen, die Aussage ist richtig für n–elementige Mengen. Sei $A = \{a_1, \ldots, a_{n+1}\}$ eine $(n+1)$–elementige Menge.
Betrachten $A' = A \setminus \{a_{n+1}\}$. Es gibt nach Annahme $n!$ Permutationen von A'.
Wir bilden $n+1$ viele paarweise disjunkte Mengen $\Pi(A)_i$ von Permutationen von A. $\Pi(A)_i$ entsteht dadurch, dass wir in jede Permutation von A' das Element a_{n+1} genau an die i-te Stelle einfügen. Dies ist eine Bijektion zwischen allen Permutationen von A' und $\Pi(A)_i$, also $|\Pi(A)_i| = n!$ für jedes i.
Da jede Permutation von A zu genau einer der Mengen $\Pi(A)_i$ gehört, gilt nach der Summenregel, dass es $(n+1) \cdot n! = (n+1)!$ viele Permutationen von A gibt. $\square$

5.1.1 Binomialkoeffizienten

Bezeichne der *Binomialkoeffizient* $\binom{n}{k}$ die Anzahl der k–elementigen Untermengen einer n–elementigen Menge M. Auch ohne zu wissen, wie groß diese Zahl ist, können wir sagen: $\sum_{k=0}^{n} \binom{n}{k} = 2^n$, denn die Anzahl aller Untermengen von M ist die Mächtigkeit der Potenzmenge von M und das ist 2^n.
Außerdem ist $\binom{n}{0} = \binom{n}{n} = 1$ und $\binom{n}{k} = 0$ für $k > n$.

Satz: Für $n \geq k$ gilt:

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

Beweis: Zunächst zählen wir eine größere Menge ab. Wieviele Möglichkeiten gibt es, eine Folge der Länge k aus verschiedenen Elementen zu bilden? Das ist einfach:
Es gibt n Möglichkeiten, das erste Element der Folge auszuwählen. Hat man das getan, gibt es für jede Auswahl des ersten Elements noch $n-1$ Möglichkeiten das zweite Element festzulegen usw., schließlich $n-k+1$ Möglichkeiten für das k-te Element der Folge. Zusammen sind dies

$$n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot (n-k+1) = \frac{n!}{(n-k)!}$$

Möglichkeiten. Wie oft wird dabei eine konkrete k–elementige Menge aufgelistet? Das passiert $k!$ mal. Wir fassen jeweils diese $k!$ Auflistungen zusammen, indem wir die Anzahl aller Auflistungen durch $k!$ teilen. □

Hinweis: Die Auflistung von k der n Elemente von M heißt *k–Permutation* von M. Deren Anzahl $n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot (n-k+1)$ wird auch mit $n^{\underline{k}}$ bezeichnet und heißt *fallende Faktorielle*.

Beispiel 1:
Der Ziehungsvorgang beim Lotto "6 aus 49" liefert zunächst eine 6–Permutation der Menge $\{1,2,\ldots,49\}$ (wir vernachlässigen hier die Zusatzzahl). Die Anzahl dieser 6–Permutationen ist $49^{\underline{6}}$. Für das Ergebnis der Ziehung ist nur die Menge der gezogenen Zahlen entscheidend und nicht die Reihenfolge, in der sie gezogen wurden. Deshalb hieß es auch immer am Schluss (die Übertragung der Ziehungen im deutschen Fernsehen wurden 2013 eingestellt): " ... und hier noch einmal die gezogenen Zahlen in geordneter Reihenfolge ...". Da jedes Ziehungsergebnis in 6! verschiedenen Reihenfolgen gezogen werden kann, ist die Anzahl der möglichen Ziehungsergebnisse (und damit auch die Anzahl der möglichen Tips) gleich

$$\frac{49^{\underline{6}}}{6!} = \binom{49}{6} = 13983816$$

.

Beispiel 2:

1. Es gibt $\binom{n}{k}$ 0–1–Strings der Länge n mit genau k Einsen. Das ist klar, denn wir können bestimmen, an welchen k der n Stellen die Einsen stehen sollen.

2. Es gibt $\binom{n}{k} \cdot 2^{n-k}$ Strings der Länge n über dem Alphabet $\{0,1,2\}$ mit genau k Einsen.
   Warum? $\binom{n}{k}$ zählt die Möglichkeiten, die k Stellen auszuwählen. Was steht an den anderen $n-k$ Stellen? Jeweils 0 oder 2 und die Produktregel liefert das Ergebnis.

3. Es gibt $\binom{n}{k} \cdot \binom{n-k}{l}$ Strings der Länge n über dem Alphabet $\{0,1,2\}$ mit genau k Einsen und l Zweien.

Satz (Rekursiver Zugang zu Binomialkoeffizienten)
Es gilt $\binom{n}{0} = 1$, $\binom{n}{k} = 0$ für $k > n$ und

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1} \text{ für } k \leq n.$$

Beweis Wir fixieren ein Element $a \in M$. Wir unterteilen die Menge aller k–elementigen Teilmengen S von M in zwei disjunkte (!) Teilmengen.
Zum ersten sind dies alle S, die a nicht enthalten. Wieviele gibt es? Um eine solche Menge zu bestimmen, müssen wir k Elemente aus $M \setminus \{a\}$wählen. Damit sind es $\binom{n-1}{k}$ viele.
Wir zählen jetzt die k-Mengen S, die a enthalten. Wenn a drin ist, müssen wir noch $k-1$ Elemente aus den restlichen $n-1$ Elementen ziehen, dafür gibt es $\binom{n-1}{k-1}$ Möglichkeiten.
Die Summenformel liefert das Ergebnis. □

Es gibt eine Vielzahl von Identitäten für Binomialkoeffizienten. Einige werden wir jetzt genauer untersuchen. Als prinzipielle Beweismöglichkeiten gibt es

- direktes Nachrechnen mit Formel
- vollständige Induktion
- Konstruktion von Bijektionen; insbesondere das Abzählen einer Menge auf verschiedene Arten.

1. **Symmetrie:** Es gilt: $\binom{n}{k} = \binom{n}{n-k}$.
   Dies folgt sofort aus der Formel. Oder auch mittels Bijektion: Jede k elementige Teilmenge bestimmt eindeutig ihr $(n-k)$–elementiges Komplement und umgekehrt.

2. **Pascalsches Dreieck**: Das ist der oben geschilderte rekursive Zugang.Wir haben $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$.
   In Tabellenform ergibt das:

n \ k	0	1	2	3	4	5	6…
0	1	0					
1	1	1	0				
2	1	2	1	0			
3	1	3	3	1	0		
4	1	4	6	4	1	0	
5	1	5	10	10	5	1	0
6	1	6	15	20	15	6	1
:							

   Es ist klar, daß die Summe aller Einträge in der n–ten Zeile die Anzahl aller Untermengen einer n–Menge ergibt: $\sum_{k=0}^{n} \binom{n}{k} = 2^n$.

3. **Vandermonde'sche Identität**: Es gilt

$$\binom{n+m}{k} = \sum_{i=0}^{k} \binom{n}{i} \binom{m}{k-i}$$

Dafür gibt es einen schönen kombinatorischen Beweis:
Wir betrachten eine $(n+m)$–Menge A und zwei disjunkte Teilmengen B und C mit $|B| = n$ und $|C| = m$. Jede k–elementige Teilmenge von A besteht aus i Elementen aus B und $k-i$ Elementen aus C für ein i mit $0 \leq i \leq k$. □

4. **Binomischer Satz** Der Grund dafür, dass Binomialkoeffizienten so heißen, wie sie heißen, liegt in folgendem Satz:

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

Beweis: Wir zählen, wie oft beim Ausmultiplizieren von $(x+y)\cdot(x+y)\cdot\ldots\cdot(x+y)$ der Term $x^{n-k}y^k$ entsteht.
Das ist offensichtlich die Anzahl der Möglichkeiten beim Ausmultiplizieren aus genau $n-k$ der insgesamt n Klammern das x zu wählen, aus den restlichen k Klammern muss dann das y genommen werden. Also sind das $\binom{n}{n-k} = \binom{n}{k}$ Möglichkeiten. Das k kann im Bereich $0 \leq k \leq n$ liegen. □

Bemerkung: Als Spezialfälle des Binomischen Satzes können wir 2 Gleichungen ableiten, die erste kennen wir schon.

$$2^n = (1+1)^n = \sum_{k=0}^{n} \binom{n}{k}$$

$$0 = (-1+1)^n = \sum_{k=0}^{n} (-1)^k \binom{n}{k}$$

5. **Verallgemeinerte Rekursion** Es gilt der folgende Sachverhalt für die Spaltensummen im Pascalschen Dreieck:

$$\binom{n+1}{k+1} = \sum_{i=0}^{n} \binom{i}{k}$$

Beweis: (kombinatorisch)
Wir betrachten die $(n+1)$–Menge $L = \{0,1,2,\ldots,n\}$ von "Losen". Dann beschreibt $\binom{n+1}{k+1}$ die Anzahl der möglichen Ziehungen von $k+1$ Losen. Sei Z diese Menge. Wir zerlegen Z in disjunkte Teile. Sei Z_i die Menge von Ziehungen (von $k+1$ Losen), bei denen Los i als höchstes Los gezogen wurde. Offensichtlich $Z = \bigcup_{i=0}^{n} Z_i$. Wie groß ist $|Z_i|$? Wir haben $\binom{i}{k}$ Möglichkeiten, das höchste Los i mit weiteren k Losen aus $\{0,1,\ldots,i-1\}$ zu ergänzen. Die Summenformel liefert das Ergebnis. □

6. **Newton–Identität** Für $0 \leq l \leq k \leq n$ gilt:

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

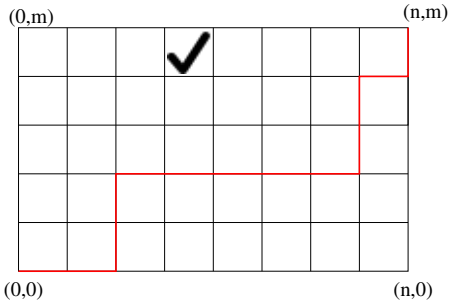
Beweis: Der kombinatorische Beweis ist recht einfach. Die linke Seite der Gleichung zählt, wie oft man in der n–Menge eine k–Untermenge markieren kann, um

dann aus der k–Menge noch eine l–Teilmenge auszuwählen. Die rechte Seite zählt dieselben Objekte, nur auf andere Art. Wir wählen zuerst die l–Teilmenge und ergänzen diese dann zur markierten k–Teilmenge. Dazu muss man $k-l$ Elemente aus den verbliebenen $n-l$ Elementen auswählen. □

5.1.2 Binomialkoeffizienten und monotone Gitterwege

Hier folgt eine schöne geometrische Interpretation der Binomialkoeffizienten, bei der sich eine Reihe der obigen Identitäten als Eigenschaften von Gitterwegen wiederfindet.

Definition: Ein monotoner Weg in einem ganzzahligen Gitter der Form $\{0,1,\ldots,n\} \times \{0,1,\ldots,m\}$ ist ein Weg von $(0,0)$ zum Knoten (n,m), der nur Schritte nach rechts oder oben macht.



Fakt: Jeder monotone Gitterweg von $(0,0)$ zum Knoten (n,m) enthält $n+m$ Kanten, davon sind n waagerecht und m senkrecht.
Ein solcher Weg ist eindeutig dadurch bestimmt, dass festgelegt wird, bei welchen Schritten nach rechts gegangen wird.
Es gibt $\binom{n+m}{n}$ verschiedene monotone Wege. □

Da es egal ist, ob wir die waagerechten Schritte oder die senkrechten festlegen, haben wir:

$$\binom{n+m}{n} = \binom{n+m}{m}$$

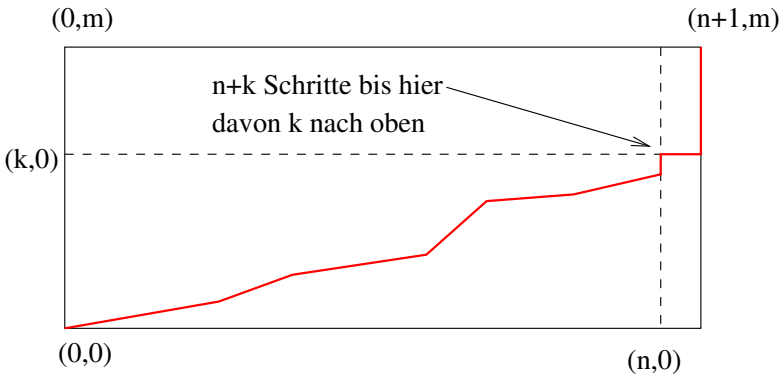
Das ist die oben angesprochene Symmetrie innerhalb der Zeilen des Pascalschen Dreiecks.

Auch die Rekursionsformel findet man hier wieder: Wir zerlegen die Menge aller monotonen Wege von $(0,0)$ zum Knoten (n,m) in zwei disjunkte Mengen: Diejenigen Wege, die mit einem Schritt nach rechts anfangen, und jene, die als erstes einen Schritt nach oben machen. Von ersterer Sorte gibt es $\binom{n+m-1}{n-1}$ viele, denn bei den verbliebenen $n+m-1$ Schritten müssen noch $n-1$ nach rechts gehen; von der zweiten Sorte gibt es $\binom{n+m-1}{m-1} = \binom{n+m-1}{n}$ viele.

Die verallgemeinerte Rekursionsformel sieht im Kontext der Gitterwege wie folgt aus:

Wir betrachten monotone Gitterwege von $(0,0)$ nach $(n+1,m)$. Jeder Weg muss die vertikale durch $(n,0)$ kreuzen, genauer wir betrachten den Zeitpunkt, bei dem der Weg den letzten Schritt nach rechts macht. Danach geht der Weg nur nach oben. Wenn das in Höhe k passiert, gab es vorher insgesamt $n+k$ Schritte, davon k nach oben. Die Summenformel liefert:

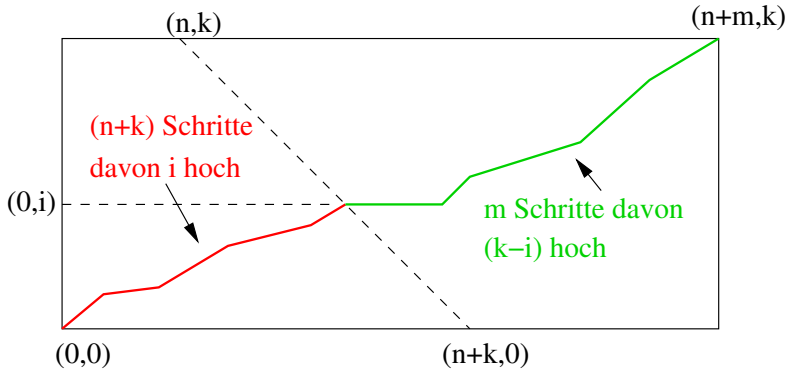
$$\sum_{k=0}^{m} \binom{n+k}{k} = \binom{m+n+1}{m}$$



Zum Schluss das Bild zur Vandermondeschen Gleichung:

$$\binom{n+k+m}{k} = \sum_{i=0}^{k} \binom{n+k}{i} \cdot \binom{m}{k-i}$$

Hierbei wird die Menge aller Wege unterteilt bezüglich der Anzahl i der vertikalen Schritte unter den ersten $n+k$ Schritten:



5.1.3 Mengenpartitionen

Definition Sei M eine n–elementige Menge. Eine Zerlegung von M in k nichtleere, paarweise disjunkte Teilmengen (*Blöcke*) heißt *k–Partition* von M.

Wir bezeichnen mit $S_{n,k}$ die Anzahl der verschiedenen k–Partitionen. Diese Zahlen heißen auch *Stirling–Zahlen 2. Art*.
Hinweis: Partitionen sind Mengen, genauer Mengen von Teilmengen. Möchte man die Reihenfolge der Teilmengen berücksichtigen, so spricht man von *geordneten k–Partitionen* und davon gibt es dann $k! \cdot S_{n,k}$ viele.

Vorüberlegungen: Wir können einige der Werte $S_{n,k}$ sofort bestimmen.

1. Es ist $S_{n,n} = 1$ für alle $n \in \mathbb{N}$. Für $n > 0$ gibt es da nichts zu zeigen, denn jeder Block muss aus genau einem Element bestehen. Dass $S_{0,0} = 1$ ist, wird so vereinbart und hilft uns dann beim Rekursionsanker.
   Weiterhin ist $S_{n,0} = 0$ für $n > 0$ ebenso wie $S_{n,k} = 0$ für $n < k$.

2. Wir zeigen $S_{n,2} = 2^{n-1} - 1$.
   Beweis: Jede echte (d.h. nichtleere und nichtvolle) Teilmenge von M bestimmt eindeutig den zweiten Block in einer 2–Partition. Es gibt $2^n - 2$ solche echten Teilmengen und jede 2–Partition wird zweimal aufgezählt. □

3. Es gilt: $S_{n,n-1} = \binom{n}{2}$.
   Beweis: Alle Blöcke bis auf einen enthalten genau ein Element und ein Block enthält 2 Elemente. Diese 2 Elemente bestimmen somit, wie alle anderen Blöcke aussehen, und $\binom{n}{2}$ zählt die Möglichkeiten, 2 Elemente auszuwählen. □

Der folgende Satz gibt die Rekursionsformel zur Bestimmung der $S_{n,k}$ an.

Satz: Mit $S_{0,0} = 1, S_{n,0} = 0$ für $n > 0$ und $S_{n,k} = 0$ für $n < k$ gilt für alle $n \geq k \geq 1$:

$$S_{n,k} = S_{n-1,k-1} + k \cdot S_{n-1,k}$$

Beweis: (kombinatorisch)
Wir fixieren in M ein Element a. Wie kann eine k–Partition von M aussehen?
1. Möglichkeit: $\{a\}$ bildet einen eigenen Block. Die restliche Menge $M \setminus \{a\}$ mit $n-1$ Elementen zerfällt in $k-1$ Blöcke. Dafür gibt es $S_{n-1,k-1}$ viele Möglichkeiten.
2. Möglichkeit: Das Element a ist nicht allein in einem Block. Entfernt man a, so bleibt eine k–Partition von $M \setminus \{a\}$ übrig. Umgekehrt kann man jede k–Partition von $M \setminus \{a\}$ auf k Arten zu einer k–Partition von M machen, man kann nämlich a in jeden der k Blöcke stecken.
Die Summenformel liefert das Ergebnis. □

In Tabellenform ergibt das:

n	$S_{n,0}$	$S_{n,1}$	$S_{n,2}$	$S_{n,3}$	$S_{n,4}$	$S_{n,5}$	$S_{n,6}\ldots$
0	1	0					
1	0	1	0				
2	0	1	1	0			
3	0	1	3	1	0		
4	0	1	7	6	1	0	
5	0	1	15	25	10	1	0
6	0	1	31	90	65	15	1
:							

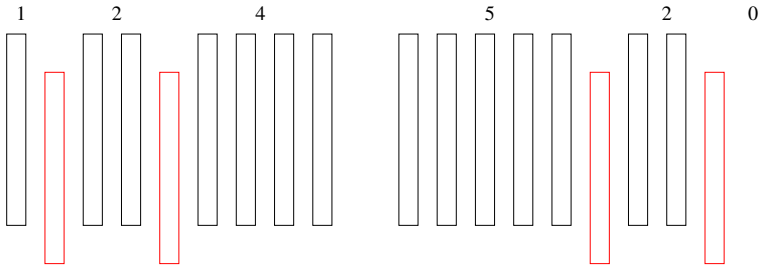
Satz: (Geschlossene Formel für die $S_{n,k}$, ohne Beweis)

$$S_{n,k} = \frac{1}{k!}\sum_{i=0}^{k}(-1)^{(k-i)}\binom{k}{i}i^n$$

Beispiel: Anzahl surjektiver Funktionen
Wieviele surjektive Funktionen $f : M \to A$ mit $|M| = n, |A| = k$ gibt es? Wir wissen, jede solche surjektive Funktion bestimmt eine k–Partition von M, das sind die Urbilder der k Elemente aus A. Eine solche k–Partition sagt aber noch nicht, welcher Block auf welches Element $i \in A$ abgebildet wird. Für eine gegebene k–Partition von M gibt es dafür $k!$ viele Möglichkeiten.
Die Anzahl der surjektiven Funktionen entspricht also der Anzahl $k!S_{n,k}$ von geordneten k–Partitionen der Menge M.

5.1.4 Zahlpartitionen

Beispiel: Wieviele Möglichkeiten gibt es, 7 Stück weiße Kreide unter 3 Dozenten zu verteilen. Zerteilen wollen wir die Stücke nicht, wie im richtigen Leben kann jemand auch gar nichts bekommen und die Stücke sind nicht unterscheidbar. Das entspricht also der Anzahl der geordneten Zerlegungen $7 = n_1 + n_2 + n_3$ mit $n_i \geq 0$.



Im Bild sind zwei zulässige Zerlegungen dargestellt und man erkennt auch den Lösungsweg. Eine Zerlegung von 7 Stück in 3 Teile entspricht genau der Auswahl von 2 Trenn–Stücken aus 9=7+2 Stücken. Davon gibt es $\binom{9}{2} = 36$ viele.

Satz: Es gibt $\binom{n+k-1}{k-1}$ viele geordnete Summendarstellungen $n = n_1 + n_2 + \ldots + n_k$ mit den $n_i \geq 0$.

Der Beweis sollte als Verallgemeinerung des Beispiels klar sein.

Anmerkungen:
(1) Es gibt $\binom{n+k}{k}$ viele geordnete Lösungen für $n \geq n_1 + n_2 + \ldots + n_k$ mit den $n_i \geq 0$. Man betrachte dazu die geordneten Summendarstellungen $n = n_1 + n_2 + \ldots + n_k + n_{k+1}$ mit allen $n_i \geq 0$.
(2) Hat man eine zusätzliche Nebenbedingung wie etwa $n_1 \geq 3$ gegeben, so ist die Anzahl der geordneten Summendarstellungen von n mit k Summanden $\binom{n+k-4}{k-1}$.

Spricht man von Zahlpartitionen, so meint man Summendarstellungen mit Summanden ≥ 1.

Definition:

1. Eine ungeordnete k–Partition einer ganzen Zahl $n > 0$ ist eine (Multi-)Menge von positiven ganzen Zahlen $\{n_1, n_2, \ldots, n_k\}$ mit $\sum_{i=1}^{k} n_i = n$. Bezeichne $P_{n,k}$ die Anzahl dieser ungeordneten k–Partitionen von n.

2. Eine geordnete k–Partition einer ganzen Zahl $n > 0$ ist eine Folge von positiven ganzen Zahlen $n_1, n_2, \ldots, n_k$ mit $\sum_{i=1}^{k} n_i = n$.

Satz: Es gibt $\binom{n-1}{k-1}$ viele geordnete k–Partitionen einer ganzen Zahl $n > 0$.
Beweis: Wir schreiben uns die Zahl n als Summe von Einsen, das ist eine Folge von n 1'en und $(n-1)$ (+)–Zeichen.

$$n = \underbrace{1+1+1+\ldots+1}_{n \text{ mal}}$$

Wählen wir von den $(+)$–Zeichen in dieser Darstellung $k-1$ aus, so erzeugen wir eine geordnete k–Partition von n, und umgekehrt. □

Man beachte den kleinen Unterschied zum Beispiel oben. Dort war die Null als Summand zugelassen, was man als Auswahl von $(+)$–Zeichen nicht oder nur schwer modellieren kann.
Wir bestimmen als nächstes die Anzahl der ungeordneten Zahlpartitionen.
Beachte: Wenn die Reihenfolge der Summanden keine Rolle spielt, dann können wir o.B.d.A. annehmen, dass $n_1 \leq n_2 \leq \ldots \leq n_k$ gilt.

Satz: Es ist $P_{n,n} = 1$ und es gilt für $n \geq 1, k \geq 1$ die folgende Rekursionsformel:

$$P_{n+k,k} = \sum_{j=1}^{k} P_{n,j}$$

Beweis: Zunächst ändern wir die Reihenfolge in der Summation und schreiben:

$$\sum_{j=1}^{k} P_{n,j} = \sum_{i=0}^{k-1} P_{n,k-i}$$

Nun konstruieren wir eine Bijektion zwischen der Menge der k–Partitionen von $n+k$ und der Vereinigung der $(k-i)$–Partitionen von n für $0 \leq i \leq k-1$.
Betrachten wir eine Partition

$$n+k = n_1 + n_2 + \ldots + n_k$$

Sei i die Anzahl der Einsen in der Partition, d.h. $n_1 = \ldots = n_i = 1$. Wegen $n > 0$ ist $i < k$.
Es ist dann

$$n+k = i + n_{i+1} + n_{i+2} + \ldots n_k$$

Also ist auch

$$n = (i-k) + n_{i+1} + n_{i+2} + \ldots n_k = (n_{i+1} - 1) + (n_{i+2} - 1) + \ldots (n_k - 1)$$

eine $(k-i)$–Partition von n.
Umgedreht, sei

$$n = n'_1 + n'_2 + \ldots + n'_{k-i}$$

eine Partition von n. Dann ist

$$n + k = \underbrace{1 + 1 + \ldots + 1}_{i \text{ Einsen}} + (n'_1 + 1) + (n'_2 + 1) + \ldots + (n'_{k-i} + 1)$$

eine k–Partition von $n+k$.
Da diese Zuordnung bijektiv ist, gilt die obige Rekursionsformel. □

5.1.5 Doppeltes Abzählen

Hier ist ein Standardtrick, der oft angewendet wird: Das ist das *doppelte Abzählen*.

Definition: Ein *Inzidenzsystem* (S,T,I) besteht aus zwei (endlichen) Mengen S und T und einer *Inzidenzrelation* $I \subseteq S \times T$.

Satz: (Regel vom zweifachen Abzählen) Sei (S,T,I) ein Inzidenzsystem und sei $r(a) = |\{b \in T \mid (a,b) \in I\}|$, $r(b) = |\{a \in S \mid (a,b) \in I\}|$ für alle $a \in S$, $b \in T$, dann gilt:

$$\sum_{a \in S} r(a) = \sum_{b \in T} r(b).$$

Beweis: In der Tat sind beide Summen gleich $|I|$, denn die Paare der Relation I wurden nur auf zwei verschiedene Weisen aufgezählt. □

Oft wird diese Regel auch in etwas modifizierter Form unter Verwendung von Inzidenzmatrizen oder bipartiten Graphen verwendet:
Sei $M = (m_{ij})$ eine $m \times n$ Matrix, d.h. ein Rechteck–Schema mit m Zeilen und n Spalten, in dem Zahlen m_{ij} (i-te Zeile, j-te Spalte) eingetragen sind.
Sind $z_i = \sum_{j=1}^{n} m_{ij}$ $(1 \leq i \leq m)$ die Zeilensummen und $s_j = \sum_{i=1}^{m} m_{ij}$ $(1 \leq j \leq n)$ die Spaltensummen, dann gilt $\sum_{i=1}^{m} z_i = \sum_{j=1}^{n} s_j$, denn beide Summen sind gleich der Summe aller Matrixelemente. Die Regel vom zweifachen Abzählen ergibt sich nun als Spezialfall für die sogenannte *Inzidenzmatrix*.
Dazu werden die Elemente aus S und T nummeriert, $S = \{a_1, \ldots, a_m\}$, $T = \{b_1, \ldots, b_n\}$ und $m_{ij} = \begin{cases} 1 & falls\ (a_i, b_j) \in I \\ 0 & sonst \end{cases}$ gesetzt.

Beispiel:
Sei $S = T = \{1, 2, \ldots, n\}$ und wir definieren die binäre Relation I als die Teilbarkeitsrelation. Das heißt, aIb falls $a|b$.

Hier ist die Inzidenzmatrix für den Fall $n = 8$:

	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1	1
2	0	1	0	1	0	1	0	1
3	0	0	1	0	0	1	0	0
4	0	0	0	1	0	0	0	1
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	1	0	0
7	0	0	0	0	0	0	1	0
8	0	0	0	0	0	0	0	1

Sei $t(j)$ die Anzahl der Teiler der Zahl j. Das ist nichts anderes als die Summe der Einträge in der j–ten Spalte.
Was kann man über die durchschnittliche Anzahl $t^*(n) = \frac{1}{n}\sum_{j=1}^{n} t(j)$ von Teilern einer Zahl $\leq n$ sagen? Hier hilft doppeltes Abzählen und wir benutzen, dass in der i–ten Zeile jeder i–te Eintrag 1 ist:

$$t^*(n) = \frac{1}{n}\sum_{j=1}^{n} t(j) = \frac{1}{n}\sum_{i=1}^{n} \lfloor \frac{n}{i} \rfloor \approx \sum_{i=1}^{n} \frac{1}{i} = H_n \approx \ln n$$

Dabei ist H_n die sogenannte n–te Harmonische Zahl, die uns noch oft begegnen wird und die den natürlichen Logarithmus $\ln n$ von n approximiert.

5.2 Die 12 Arten des Abzählens und ein Kartentrick

Welche Zählkoeffizienten haben wir bisher kennengelernt und was zählen sie?

- Anzahl aller Untermengen einer n–Menge: 2^n
- Anzahl aller k–Untermengen einer n–Menge: $\binom{n}{k}$
- Anzahl aller Permutationen einer n–Menge: $n!$
- Anzahl geordneter k–Teilmengen einer n–Menge: $n^{\underline{k}} = n \cdot (n-1) \cdot \ldots \cdot (n-k+1)$
- Anzahl aller k–Partitionen einer n–Menge: $S_{n,k}$
- Anzahl der geordneten k–Zahlpartitionen der Zahl n: $\binom{n-1}{k-1}$
- Anzahl der ungeordneten k–Zahlpartitionen der Zahl n: $P_{n,k}$

Die hier angesprochenen Zählkoeffizienten sind auch sehr gut zum Abzählen von Funktionenmengen geeignet:
Sei N eine n–Menge und R eine r–Menge.
Aus der Mengenlehre wissen wir, dass die Anzahl aller Abbildungen von N nach R gleich r^n ist, denn für jedes der n Elemente aus N muss genau eines von r möglichen Elementen aus R als Bild festgesetzt werden.
Ist $n \leq r$, dann sind injektive Abbildungen von N in R durch n–Permutationen von R charakterisiert, d.h. $|Inj(N,R)| = r^{\underline{n}}$.

Ist $n \geq r$, dann sind surjektive Abbildungen von N auf R durch geordnete r–Partitionen von N charakterisiert, d.h.
$|Surj(N,R)| = r!S_{n,r}$.
Die Anzahl der bijektiven Abbildungen ist $n!$ (falls $n = r$) oder 0 sonst.
Klassifizieren wir die Menge aller Abbildungen $f : N \longrightarrow R$ nach ihren Bildern $A = \{f(x) \mid x \in N\} \subseteq R$, so ergibt die Summenregel noch folgende Identität:

$$\begin{aligned} r^n = \quad |Abb(N,R)| \quad &= \sum_{A \subseteq R} |Surj(N,A)| \\ &= \sum_{k=0}^{r} \sum_{|A|=k} |Surj(N,A)| \\ &= \sum_{k=0}^{r} \binom{r}{k} k! S_{n,k} \\ &= \sum_{k=0}^{r} r^{\underline{k}} S_{n,k} \end{aligned}$$

Eine zusammenfassende Interpretation der Zählkoeffizienten erhalten wir durch die folgende Fragestellung:
Sei eine Menge von n Bällen gegeben, die in r Fächer verteilt werden sollen. Gesucht ist die Anzahl solcher Verteilungen allgemein und unter der zusätzlichen Bedingung, dass nur injektive, surjektive bzw. bijektive Verteilungen zu betrachten sind.
Natürlich hängt die Problemstellung auch davon ab, ob die Bälle bzw. die Fächer unterscheidbar sind oder nicht.

Offensichtlich korrespondieren die obigen Abzählungen von Abbildungen zu dem Fall, dass sowohl die Bälle als auch die Fächer unterscheidbar sind. Da die Frage der bijektiven Verteilungen relativ leicht zu beantworten sind, reduziert sich die Problemstellung auf zwölf Varianten.

1. Sind nur die Bälle unterscheidbar, aber nicht die Fächer, dann korrespondieren surjektive Verteilungen zu r–Partitionen einer n–Menge und die Menge aller Verteilungen kann mit der Summenregel über die Klassifizierung nach Anzahl der belegten Fächer abgezählt werden.

2. Sind sowohl Bälle als auch die Fächer nicht unterscheidbar, dann korrespondieren surjektive Verteilungen zu ungeordneten r–Zahlpartitionen von n und auch hier kann die Menge aller Verteilungen mit der Summenregel über die Klassifizierung nach Anzahl der belegten Fächer abgezählt werden.

3. Sind nur die Fächer unterscheidbar, aber nicht die Bälle, dann korrespondieren die surjektiven Verteilungen zu geordneten r–Zahlpartitionen von n. Die injektiven Verteilungen sind durch die Auswahl der n belegten Fächer aus der r–Menge aller Fächer vollständig charakterisiert. Eine beliebige Abbildung ist durch eine geordnete Summenzerlegung der Form $n = n_1 + \ldots + n_r$ mit $n_i \in \mathbb{N}$ charakterisiert. Um daraus eine r–Zahlpartition im Sinne der Definition zu machen (Zusatzbedingung $n_i \geq 1$) addiert man zu jedem Summanden eine 1. Damit erhalten wir eine Bijektion auf die Menge der geordneten r–Zahlpartitionen von $(n+r)$. Diese Menge hat $\binom{n+r-1}{r-1}$ Elemente.

Die folgende Tabelle gibt einen vollständigen Überblick:

	beliebig	injektiv	surjektiv	bijektiv
N unterscheidbar R unterscheidbar	r^n	$r^{\underline{n}}$	$r!S_{n,r}$	0 oder $n!$
N nicht untersch. R unterscheidbar	$\binom{n+r-1}{r-1}$	$\binom{r}{n}$	$\binom{n-1}{r-1}$	0 oder 1
N unterscheidbar R nicht untersch.	$\sum_{k=1}^{r} S_{n,k}$	0 oder 1	$S_{n,r}$	0 oder 1
N nicht untersch. R nicht untersch.	$\sum_{k=1}^{r} P_{n,k}$	0 oder 1	$P_{n,r}$	0 oder 1

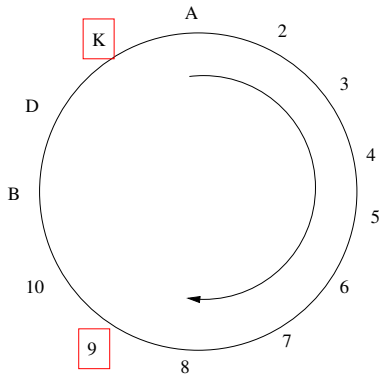
Zum Schluss dieses Abschnittes ein **Kartentrick**:

Beteiligte: Magier M, Assistent A und Publikum P

Ablauf: Das Publikum zieht 5 Karten aus einem Kartenspiel mit 52 Blatt. A sieht sich die Karten an und zeigt dann nacheinander 4 der 5 Karten dem Magier. M sagt dann die 5. noch nicht gesehene Karte richtig voraus!

Für den Informatiker: A muss mit den 4 Karten jede der 48 restlichen Karten kodieren können. Welche Möglichkeiten hat er? Er kann bestimmen, in welcher Reihenfolge er die Karten zeigt, und er kann bestimmen, welche 4 Karten er zeigt.

Satz: Es funktioniert!
Beweis: Bei 5 Karten muss wenigstens eine Farbe doppelt auftreten. Eine davon zeigt A als erste Karte k_1, die andere wird die magische 5. Karte k_5. Damit kennt M schon deren Farbe. Welche ist die erste und welche die 5. Karte? Wir ordnen die Kartenwerte zyklisch im Uhrzeigersinn an. Eine Auswahl von zwei Karten zerlegt den Kreis der 13 Karten in zwei Abschnitte, einer davon hat Länge $l \leq 6$. Die erste Karte k_1 ist der Anfang dieses kürzeren Segments (im Uhrzeigersinn). Im Beispiel die Neun. Die andere von M zu erratende Karte k_5 lässt sich berechnen als $k_5 = k_1 + l$Schritte im Uhrzeigersinn. Im gezeigten Beispiel: Neun+4=König. Was muss A also tun? Er muss M mit den restlichen



drei Karten diese Zahl l kommunizieren, dann kann M k_5 ausrechnen.
Wie geht das? Wir bringen alle 52 Karten in eine lineare Ordnung. Zum Beispiel lexiko-

graphisch: erst die Farbe, dann der Wert. Das ergibt als Ordnung:
Karo2 $<$ Karo3 $<$ ... $<$ KaroAss $<$ Herz2 $<$ ...$<$ HerzAss $<$ Pik2 $<$ ....$<$ PikAss $<$ Kreuz2 $<$ ...$<$ KreuzAss
Die drei restlichen verschiedenen Karten von A seien bezüglich dieser Ordnung $k < m < g$. Es gibt 6 Permutationen dieser Karten und das benutzt A wie folgt, um l zu kodieren:

$$(k,m,g) \leftrightarrow 1, (k,g,m) \leftrightarrow 2, (m,k,g) \leftrightarrow 3$$
$$(m,g,k) \leftrightarrow 4, (g,k,m) \leftrightarrow 5, (g,m,k) \leftrightarrow 6$$

A zeigt die 3 Karten nach k_1 in der entsprechenden Reihenfolge. □

Anmerkung: Einen analogen Trick mit 4 gezogenen und 3 gezeigten Karten kann es nicht geben! Es gibt echt mehr Mengen mit 4 Karten als Folgen mit 3 Karten:
$\binom{52}{4} > 52 \cdot 51 \cdot 50$. Laut Schubfachprinzip würden verschiedene Mengen existieren, bei denen der Assistent dieselbe Folge von 3 Karten zeigt! Widerspruch.

5.3 Abzählen II: Diskrete Wahrscheinlichkeitsrechnung; Grundlagen

Die Wahrscheinlichkeitsrechnung beschäftigt sich mit der Berechnung von Wahrscheinlichkeiten für das Eintreten von bestimmten Ergebnissen bei Zufallsexperimenten.
Wir beschäftigen uns zunächst nur mit dem *diskreten* Fall, bei dem endliche oder abzählbar unendliche Grundmengen betrachtet werden. Standardmodelle sind hier etwa Würfelspiele, Münzwürfe u.ä.
Worin liegt die Bedeutung für die Informatik?

- Analyse von Algorithmen bei zufällig gewählter Eingabe
- Entwurf von Algorithmen, die Zufallsentscheidungen ausnutzen (randomisiertes Quicksort, Routing–Algorithmen, Cache–Verwaltung...)
- generell bei der vereinfachenden Modellierung/Analyse sehr komplexer Zusammenhänge

Definition: Ein *diskreter Wahrscheinlichkeitsraum* ist ein Paar (Ω, Pr), wobei Ω eine endliche (oder abzählbar unendliche) Menge von elementaren Ereignissen und
$Pr : \Omega \to [0,1]$ eine *Wahrscheinlichkeitsverteilung* ist, die jedem $\omega \in \Omega$ seine *Wahrscheinlichkeit* $Pr(\omega)$ zuordnet.
Außerdem wird als Normierung gefordert, dass $\sum_{\omega \in \Omega} Pr(\omega) = 1$ ist.
Wenn für alle $\omega \in \Omega$ auch $Pr(\omega) = \frac{1}{|\Omega|}$ gilt, dann wird Pr eine Gleichverteilung genannt.
Eine beliebige Untermenge $A \subseteq \Omega$ wird *Ereignis* genannt, und man definiert
$Pr(A) = \sum_{\omega \in A} Pr(\omega)$.

Beispiel 1:
1) Für einen (fairen) Würfel ist $\Omega = \{1,2,3,4,5,6\}$ und $Pr(1) = Pr(2) = \ldots = Pr(6) = \frac{1}{6}$ eine Gleichverteilung. Sei A das Ereignis, dass eine gerade Augenzahl gewürfelt wird. Dann ist $Pr(A) = Pr(2) + Pr(4) + Pr(6) = \frac{1}{2}$.
2) Der diskrete Wahrscheinlichkeitsraum für einen Münzwurf besteht aus $\Omega = \{0,1\}$ mit $Pr(0) = Pr(1) = 1/2$ (Gleichverteilung), wobei 0 den Kopf und 1 die Zahl der Münze bezeichnen soll.

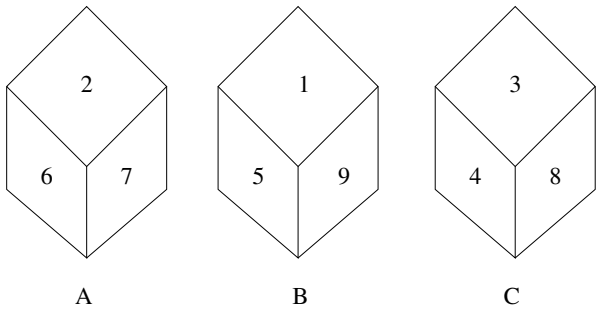
Beispiel 2:
1) Der diskrete Wahrscheinlichkeitsraum für zwei Münzwürfe besteht aus
$\Omega = \{(0,0),(0,1),(1,0),(1,1)\}$ mit der Gleichverteilung $Pr(i,j) = 1/4$. Dabei geht man davon aus, daß die beiden Würfe voneinander unabhängig sind und entweder hintereinander oder gleichzeitig mit zwei unterscheidbaren Münzen erfolgen.
2) Betrachtet man den gleichzeitigen Wurf von zwei ununterscheidbaren Münzen so ist $\Omega = \{\{0,0\},\{0,1\},\{1,1\}\}$. In diesem Fall ist Pr (faire Münzen vorausgesetzt) keine Gleichverteilung, denn $Pr(\{0,0\}) = Pr(\{1,1\}) = 1/4$ und $Pr(\{0,1\}) = 1/2$.

Aus der Definition und dem Basiswissen über Mengen folgen als elementare Eigenschaften für jeden Wahrscheinlichkeitsraum und beliebige Ereignisse:

1) $Pr(\Omega) = 1$ und $Pr(\emptyset) = 0$
2) $A \subseteq B \Rightarrow Pr(A) \leq Pr(B)$
3) $Pr(A \cup B) = Pr(A) + Pr(B) - Pr(A \cap B)$
4) $Pr(A) = \sum_{i=1}^{k} Pr(A_i)$ für jede Partition $A = \bigcup_{i=1}^{k} A_i$
5) $Pr(\Omega \setminus A) = 1 - Pr(A)$
6) $Pr(\bigcup_{i=1}^{k} A_i) \leq \sum_{i=1}^{k} Pr(A_i)$.

Das Ereignis $\Omega \setminus A$ heißt *Komplementärereignis* zu A und wird mit $\overline{A}$ bezeichnet.

Beispiel 3: Wir betrachten die drei fairen Würfel A, B, C mit den folgenden Nichtstandardnummerierungen. Dabei sind die Augenzahlen der verdeckten Seiten gleich den Augen der gegenüberliegenden Seiten.



Was ist die Wahrscheinlichkeit, dass beim Würfeln A vs. B Würfel A gewinnt?
Der Grundraum ist $\Omega = \{(2,1),(2,5),(2,9),(6,1),(6,5),(6,9),(7,1),(7,5),(7,9)\}$ und "A gewinnt gegen B"$= \{(2,1),(6,1),(6,5),(7,1),(7,5)\}$. Wegen der angenommenen Gleichverteilung ist $Pr(\text{A gewinnt gegen B}) = 5/9$. Würfel A gewinnt also gegen B!
Analog zeigt man B gewinnt gegen C mit 5/9 und die eigentliche Überraschung ist, dass C gegen A mit Wahrscheinlichkeit 5/9 gewinnt!
Es gibt also keine Transitivität der Relation "x gewinnt gegen y"!!
Merke: In der Wahrscheinlichkeitsrechnung nichts glauben sondern immer rechnen!

Beispiel 4: Wir betrachten einen nichtfairen Würfel mit

$$Pr(1) = 1/4, Pr(2) = 1/10, Pr(3) = 1/5, Pr(4) = 1/4, Pr(5) = 1/10, Pr(6) = 1/10$$

Dann ist die Wahrscheinlichkeit für gerade Augenzahl bei einmaligem Würfeln 9/20, vgl. mit Beispiel 1.

Beispiel 5: n unterscheidbare faire Münzen werden geworfen oder äquivalent: eine faire Münze wird n mal geworfen. Der Grundraum besteht aus allen 0-1–Strings der Länge n mit Gleichverteilung.
Sei A_j das Ereignis, dass genau j mal eine 1 (also Zahl) dabei ist. Dann ist

$$Pr(A_j) = \binom{n}{j} \cdot \frac{1}{2^n}$$

Sei B das Ereignis, dass eine gerade Anzahl von Einsen dabei sind, anders geschrieben:

$$B = A_0 \cup A_2 \cup \ldots \cup A_{2 \cdot \lfloor \frac{n}{2} \rfloor}$$

Wir zeigen mit Hilfe der binomischen Formel, dass $Pr(B) = \frac{1}{2}$.
Es ist

$$0 = (1-1)^n = \sum_{k=0}^{n} (-1)^k \binom{n}{k}$$

Durch Umstellen erhalten wir:

$$\binom{n}{0} + \binom{n}{2} + \ldots + \binom{n}{2\lfloor \frac{n}{2} \rfloor} = \binom{n}{1} + \binom{n}{3} + \ldots + \binom{n}{2\lfloor \frac{n}{2} \rfloor + 1}$$

Da die Summe der beiden Seiten 2^n ist, folgt für die linke Seite $|B|$ als Summe 2^{n-1}. Weil es insgesamt 2^n viele Strings der Länge n gibt, folgt daraus die Behauptung. Man beachte die Disjunktheit der A_j! □

Beispiel 6: (Geburtstagsparadoxon)
Annahme: Jeder der d =365 Tage im Jahr tritt gleichwahrscheinlich als Geburtstag auf.
Frage: Wie viele zufällige Gäste muss man einladen, damit mit Wahrscheinlichkeit $\geq 1/2$ mindestens zwei von ihnen am selben Tag Geburtstag haben. Sei die Zahl k.
Modellierung: Sei $\Omega = \{1, 2, \ldots, d\}^k$ und sei A Menge der Folgen mit mindestens zwei gleichen Einträgen. Gesucht ist also $Pr(A)$.
Wir wenden einen Standardtrick an und schätzen stattdessen die Wahrscheinlichkeit des Komplementärereignisses $\overline{A}$ ab. Was ist $|\overline{A}|$?

$$|\overline{A}| = d \cdot (d-1) \cdot \ldots \cdot (d-k+1) = d^{\underline{k}}$$

Also ist

$$Pr(A) = 1 - \frac{d^{\underline{k}}}{d^k}$$

Für $d = 365$ und $k \geq 23$ ergibt sich $Pr(A) \geq 1/2$. □

Beispiel 7:

Was ist die Wahrscheinlichkeit, dass bei einem Wurf dreier unterscheidbarer fairer Würfel mindestens ein Würfel eine gewisse Augenzahl i hat.
Antwort: Nur rund 0.42 !!!
Beweis: Sei A_k für $k = 1,2,3$ das Ereignis, dass Würfel k Augenzahl i zeigt. Gesucht ist also $Pr(A_1 \cup A_2 \cup A_3)$!
Nach dem Inklusion-Exklusions-Prinzip ist dies aber:
$Pr(A_1 \cup A_2 \cup A_3) =$
$Pr(A_1) + Pr(A_2) + Pr(A_3) - Pr(A_1 \cap A_2) - Pr(A_1 \cap A_3) - Pr(A_2 \cap A_3) + Pr(A_1 \cap A_2 \cap A_3)$
Also ist

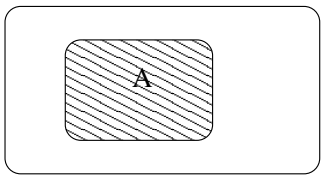
$$Pr(A_1 \cup A_2 \cup A_3) = 3 \cdot 1/6 - 3 \cdot 1/36 + 1/216 = 91/216.$$

Wir interessieren uns für die Wahrscheinlichkeit des Eintretens eines Ereignisses A unter der Bedingung, dass ein anderes Ereignis B eintritt.

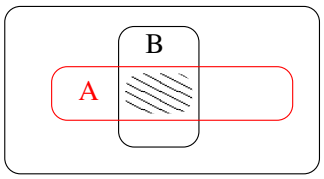
Definition: Sei (Ω, Pr) ein Wahrscheinlichkeitsraum und A, B Ereignisse mit $Pr(B) \neq 0$. Die bedingte Wahrscheinlichkeit $Pr(A|B)$ ist definiert als

$$Pr(A|B) = \frac{Pr(A \cap B)}{Pr(B)}$$

Was bedeutet dies anschaulich? Für die Gleichverteilung kann man $Pr(A)$ deuten als Anteil der Fläche (hier Anzahl der Elementarereignisse) von A an der Gesamtfläche von Ω. Dagegen ist die bedingte Wahrscheinlichkeit $Pr(A|B)$ der Anteil der Fläche von $A \cap B$ an der Fläche von B.



Pr(A)



Pr(A|B)

Beispiel 1: Betrachte 2 Würfe einer fairen Münze.
Ereignis A: Beide Würfe ergeben Kopf.
Ereignis B: Wenigstens ein Wurf ergibt Kopf.
Was ist $Pr(A|B)$?
Offensichtlich ist $A = \{(K,K)\}$ und damit $Pr(A) = 1/4$.
Weiterhin ist $B = \{(K,Z),(Z,K),(K,K)\}$ und somit $Pr(B) = 3/4$. Schließlich $Pr(A \cap B) = 1/4$ und wir erhalten $Pr(A|B) = \frac{\frac{1}{4}}{\frac{3}{4}} = \frac{1}{3}$.

Satz von Bayes:

$$Pr(A|B) = \frac{Pr(A) \cdot Pr(B|A)}{Pr(B)}$$

Beweis:

$$Pr(A|B) \cdot Pr(B) = Pr(A \cap B) = Pr(B \cap A) = Pr(B|A) \cdot Pr(A)$$

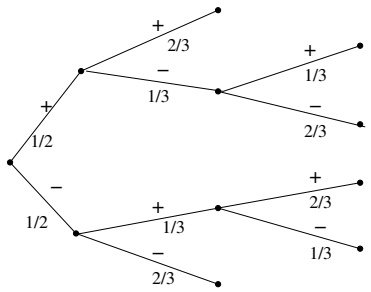
□

Wir illustrieren als nächstes die Arbeit mit sogenannten Baumdiagrammen.

Beispiel 2: Früher mussten Studenten für den Schein im Fach X mindestens zwei von drei Klausuren bestehen. Langjährige Erfahrungen sagen:

- Der durchschnittliche Student S besteht die 1. Klausur mit Wahrscheinlichkeit 1/2.
- Falls S eine Klausur bestanden hat, so besteht er die nächste mit Wahrscheinlichkeit 2/3.
- Falls S eine Klausur nicht bestanden hat, so besteht er die nächste mit Wahrscheinlichkeit 1/3.

Wir definieren Ereignisse:
A= S besteht zwei von drei Klausuren und
B= S besteht erste Klausur.
Wir wollen $Pr(A|B)$ berechnen! Dazu schauen wir uns das folgende Baumdiagramm an:



Ergebnis	Ereignis A	Ereignis B	Wkt.
+ +	Ja	Ja	1/3
+ − +	Ja	Ja	1/18
+ − −	Nein	Ja	1/9
− + +	Ja	Nein	1/9
− + −	Nein	Nein	1/18
− −	Nein	Nein	1/3

Dabei steht + für eine bestandene Klausur und − für nichtbestanden. Die Kanten sind außerdem mit Wahrscheinlichkeiten beschriftet. Wie ist das Diagramm zu lesen? Der unterste Pfad entspricht dem Ereignis, dass beide ersten Klausuren verhauen wurden. Das ist das Produkt aus Pr(1.Klausur verhauen) und Pr(2. Klausur verhauen | 1. Klausur verhauen) also 1/3.
Wenn nicht nur zwei Schritte sondern mehr gemacht werden, benutzen wir den verallgemeinerten Sachverhalt: Falls $Pr(A_1 \cap A_2 \cap \ldots \cap A_n) \neq 0$ so ist

$$Pr(A_1 \cap A_2 \cap \ldots \cap A_n) = Pr(A_1) \cdot Pr(A_2|A_1) \cdot \ldots \cdot Pr(A_n|A_1 \cap A_2 \cap \ldots \cap A_{n-1})$$

Um obiges Beispiel abzuschließen:
Wir erhalten $Pr(A) = Pr(B) = 1/3 + 1/18 + 1/9 = 1/2$, $Pr(A \cap B) = 1/3 + 1/18 = 7/18$ und damit ist $P(A|B) = 7/9$.
Wir können auch nach der sogenannten "*A Posteriori Wahrscheinlichkeit*" $Pr(B|A)$ fragen: Was ist die Wahrscheinlichkeit, dass die erste Klausur bestanden wurde, unter der Bedingung, dass zwei der drei Klausuren bestanden wurden. Schaut man sich die Werte

an, so ist $Pr(B|A) = 7/9$. Also ist hier $Pr(A|B) = Pr(B|A)$, das ist aber Zufall.

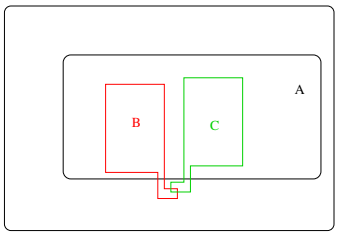
Man überlegt sich anhand der Definition, dass dies genau dann eintritt, falls $Pr(A \cap B) = 0$ oder $Pr(A) = Pr(B)$. Letzteres ist im obigen Beispiel der Fall.

Fakt: Mit dem Inklusions-Exklusions-Prinzip haben wir (nachrechnen!), dass

$$Pr(A \cup B|C) = Pr(A|C) + Pr(B|C) - Pr(A \cap B|C)$$

□

Aber die folgende Gleichung ist im Allgemeinen falsch!!!



$$Pr(A|B \cup C) = Pr(A|B) + Pr(A|C) - Pr(A|B \cap C)$$

Ein generisches Gegenbeispiel wird in der Abbildung gezeigt: Dabei sind $Pr(A|B)$, $Pr(A|C)$ und $Pr(A|B \cup C)$ fast 1, während $Pr(A|B \cap C)$ Null ist.

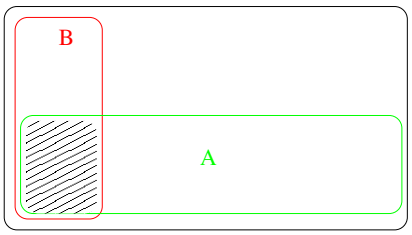
Eng mit der bedingten Wahrscheinlichkeit in Zusammenhang steht das Konzept von *unabhängigen Ereignissen*.

Definition: Zwei Ereignisse $A, B \subseteq \Omega$ heißen unabhängig, falls gilt:

$$Pr(A \cap B) = Pr(A) \cdot Pr(B)$$

.

Anschaulich heißt dies, dass der Anteil der "Fläche" von $A \cap B$ an der von B der gleiche ist wie das Verhältnis der Fläche von A zum ganzen Raum Ω.



Sind A, B unabhängig und $Pr(B) > 0$, so ist $Pr(A|B) = \frac{Pr(A \cap B)}{Pr(B)} = \frac{Pr(A) \cdot Pr(B)}{Pr(B)} = Pr(A)$.
Man kann das Konzept noch etwas allgemeiner fassen, indem man sagt, dass eine Menge

$\{A_1, A_2, \ldots, A_k\}$ mit $k > 1$ unabhängig ist, falls gilt:

$$Pr(A_1 \cap A_2 \cap \ldots \cap A_k) = Pr(A_1) \cdot Pr(A_2) \cdot \ldots \cdot Pr(A_k)$$

Wenn in einer Familie von Ereignissen beliebige Teilmengen von verschiedenen Ereignissen unabhängig sind, so spricht man auch von einer *unabhängigen Familie* (engl. mutually independent), im Gegensatz zur *paarweisen* Unabhängigkeit, bei der man nur fordert, dass jedes Paar von Ereignissen unabhängig ist.

Beispiel 3: Wir betrachten einen Wurf mit einem fairen Standardwürfel und definieren die Ereignisse. $A = \{1,2,3\}, B = \{1,4,5\}, C = \{1,2,3,4\}$.
Frage: Sind diese Ereignisse paarweise unabhängig und sind sie als Tripel unabhängig?
Lösung: Wir haben $Pr(A) = Pr(B) = 1/2$ und $Pr(C) = 2/3$.
Da $Pr(A \cap B \cap C) = 1/6 = (1/2)(1/2)(2/3)$ ist das Tripel tatsächlich unabhängig.
Was ist mit der paarweisen Unabhängigkeit? Hier sieht man, dass B und C unabhängig sind, aber sowohl A und B wie auch das Paar A und C sind nicht unabhängig!

Beispiel 4: Wir werfen zwei unterscheidbare Münzen. Bei beiden fällt Kopf mit Wahrscheinlichkeit p und Zahl mit $1-p$ für ein $0 \leq p \leq 1$. Wir betrachten Ereignisse:
A= Beide Münzen zeigen das gleiche Ergebnis.
B= 1. Münze zeigt Kopf.
Frage: Für welche Werte von p sind diese beiden Ereignisse unabhängig?
Wir haben

$$Pr(A) = p^2 + (1-p)^2$$
$$Pr(B) = p(1-p) + p^2 = p$$
$$Pr(A \cap B) = p^2$$

Die Ereignisse sind unabhängig, falls $p^2 = p(2p^2 - 2p + 1)$. Das ist nur der Fall für $p = 0$, $p = 1/2$ und $p = 1$.

Merke:

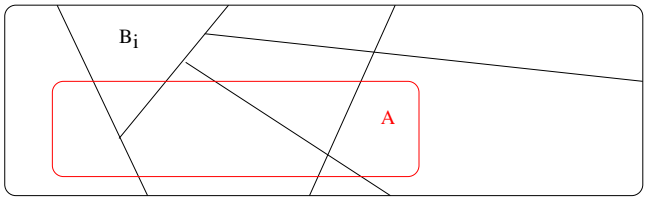
- Unabhängigkeit ist eine Eigenschaft mehrerer Ereignisse
- Disjunkte Ereignisse mit positiver Wahrscheinlichkeit sind niemals unabhängig.
- Unabhängigkeit und paarweise Unabhängigkeit einer Familie von Ereignissen sind verschiedene Konzepte.

Zum Schluss noch der Satz von der *totalen Wahrscheinlichkeit*, der in vielen Anwendungssituationen weiterhilft.

Satz: Sei eine Partition eines Wahrscheinlichkeitsraumes Ω in Mengen $B_i, i \in I$ gegeben. Für ein beliebiges Ereignis A gilt:

$$Pr(A) = \sum_{i \in I} Pr(A|B_i) \cdot Pr(B_i)$$

Beweis: Wenn die B_i eine Zerlegung von ganz Ω sind, so induzieren die nichtleeren (disjunkten!) Mengen $A \cap B_i, i \in I$ eine Partition von A.

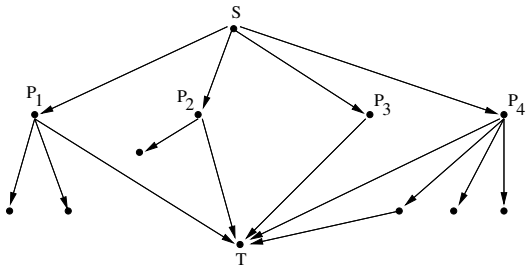


Also ist

$$Pr(A) = \sum_{i \in I} Pr(A \cap B_i)$$

Die Formel für die bedingte Wahrscheinlichkeit $Pr(A|B_i) = \frac{Pr(A \cap B_i)}{Pr(B_i)}$ liefert sofort das Ergebnis. □

Beispiel 5:
Betrachte ein gerichtetes Wegenetz. An jedem Verzweigungspunkt wird gleichverteilt eine der ausgehenden Kanten gewählt.
Frage: Was ist die Wahrscheinlichkeit, dass bei Start in S ein Weg nach T gewählt wird (Ereignis A)?



Lösung: Wir definieren Ereignisse B_i als die Wege, die durch P_i gehen.
Also ist $Pr(B_i) = 1/4$ für $1 \leq i \leq 4$.
Die Formel für die totale Wahrscheinlichkeit liefert

$$Pr(A) = 1/4(1/3 + 1/2 + 1 + 2/4) = 7/12$$

5.4 Zufallsvariable, Erwartungswert, Spezielle Verteilungen

Definition: Sei (Ω, Pr) ein Wahrscheinlichkeitsraum. Eine *Zufallsvariable (Zufallsgröße)* ist eine Funktion

$$X : \Omega \to \mathbb{R}$$

Eine Zufallsvariable (ZV) weist also den Ergebnissen eines Zufallsexperiments Zahlen zu, die etwas über das Ergebnis aussagen.
Die Ergebnisse eines Münzwurfs sind Kopf oder Zahl. Eine (sehr einfache) ZV weist dem Ergebnis Kopf die 0 zu und Zahl die 1. Oder betrachten wir zwei Würfel und wir definieren eine ZV als Maximum der beiden Augenwerte usw.

Wir fragen, mit welcher Wahrscheinlichkeit ein gewisse Zahl $x \in \mathbb{R}$ als Wert der ZV angenommen wird und stellen fest, dass dies nur von der Wahrscheinlichkeitsverteilung Pr auf Ω abhängt: Je größer die Wahrscheinlichkeit, dass ein $\omega \in \Omega$ als Ergebnis des Zufallsexperiments auftritt, desto größer die Wahrscheinlichkeit, dass $x = X(\omega)$ als Wert der ZV angenommen wird.

Formal: Für $x \in \mathbb{R}$ sei das Ereignis $(X = x) = \{\omega \in \Omega | X(\omega) = x\}$ und wir definieren auf dem Bild von X ein Wahrscheinlichkeitsmaß durch

$$Pr_X(x) = Pr(X = x) = \sum_{\omega \in \Omega, X(\omega) = x} Pr(\omega)$$

Wir können also den Wertebereich einer ZV ansehen als neuen Ereignisraum mit der induzierten Wahrscheinlichkeitsverteilung.

Beispiel 1: Betrachten wir einen Wurf mit zwei unterscheidbaren fairen Würfeln. Dann ist $\Omega = \{1,2,3,4,5,6\}^2$ und sei $X_{\max}$ die ZV, die einem Ergebnis (i,j) den Wert $\max\{i,j\}$ zuordnet. Das Bild von $X_{\max}$ ist die Menge $\{1,2,3,4,5,6\}$. Wie sieht die induzierte Wahrscheinlichkeitsverteilung aus?
Zum Bsp. ist $Pr(X_{\max} = 3) = Pr(\{(1,3),(3,1),(2,3),(3,2),(3,3)\}) = 5/36$. Für die anderen Werte haben wir: $Pr(X_{\max} = 1) = 1/36, Pr(X_{\max} = 2) = 3/36$, $Pr(X_{\max} = 4) = 7/36$
$Pr(X_{\max} = 5) = 9/36$ und $Pr(X_{\max} = 6) = 11/36$.

Man assoziiert mit einer ZV bestimmte Werte (Zahlen), die etwas über das Verhalten der ZV aussagen. Der wichtigste ist der Erwartungswert.

Definition: Sei (Ω, Pr) ein Wahrscheinlichkeitsraum und X eine Zufallsvariable. Der *Erwartungswert* von X ist definiert als

$$E(X) = \sum_{\omega \in \Omega} X(\omega) \cdot Pr(\omega)$$

Man mache sich klar, was das heißt: $E(X)$ ist der "durchschnittliche" Wert, den die ZV annimmt. Im Spezialfall der Gleichverteilung, also falls für jedes Elementarereignis ω gilt $Pr(\omega) = 1/|\Omega|$, so ist der Erwartungswert nichts anderes als das arithmetische Mittel der Werte $X(\omega)$, summiert über ω. Im Allgemeinen können wir auch summieren über die (verschiedenen) Werte, die die ZV annehmen kann, dann ist jeder einzelne Wert gewichtet mit der Wahrscheinlichkeit, dass er angenommen wird.
Genauer, es gilt der folgende Fakt.

Fakt 1: Sei $Im(X)$ das Bild der ZV $X : \Omega \to \mathbb{R}$. Es gilt

$$E(X) = \sum_{x \in Im(X)} x \cdot Pr(X = x)$$

Beweis:

$$E(X) = \sum_{\omega \in \Omega} X(\omega) \cdot Pr(\omega) = \sum_{x \in Im(X)} \sum_{\omega \in (X = x)} X(\omega) \cdot Pr(\omega) =$$

$$= \sum_{x \in Im(X)} \sum_{\omega \in (X=x)} x \cdot Pr(\omega) = \sum_{x \in Im(X)} x \cdot (\sum_{\omega \in (X=x)} Pr(\omega)) = \sum_{x \in Im(X)} x \cdot Pr(X = x)$$

□

Der Vorteil dieses Faktes ist, dass man zur Berechnung des Erwartungswertes nur über einen Bereich summieren muss, der i.A. viel kleiner als der Grundraum Ω ist.

Beispiel 1': Was ist der Erwartungswert der ZV $X_{\max}$ aus Beispiel 1?

$$E(X_{\max}) = \sum_{x \in \{1,2,\ldots,6\}} x \cdot Pr(X_{\max} = x)$$

$$= 1 \cdot (1/36) + 2 \cdot (3/36) + \ldots + 6 \cdot (11/36) = 161/36 \approx 4.47$$

Beispiel 2: Wir betrachten die erwartete Augenzahl beim einmaligen Werfen eines fairen Würfels: Hier ist die ZV X die Identität und wir haben

$$E(X) = 1 \cdot (1/6) + 2 \cdot (1/6) + \ldots + 6 \cdot (1/6) = 21/6 = 3.5$$

Wichtig: Der Erwartungswert muss wie in beiden Beispielen gesehen nicht Element des Bildes der ZV sein!

Beispiel 3: Drei "Freunde" A, B und C spielen folgendes Münzspiel. Jeder wettet auf den Ausgang eines fairen Münzwurfs und setzt 2 Euro. Wer Recht hat, teilt sich die 6 Euro. Wenn alle falsch tippen, bekommt jeder seinen Einsatz zurück. Was ist der erwartete Reingewinn von A? Wenn alle wirklich raten, ist der 0 Euro. Nachrechnen!
A wundert sich, dass er am Abend soviel verloren hat. Das liegt daran, dass B und C sich abgesprochen haben und immer auf verschiedene Ergebnisse setzen. Hier ist als Baumdiagramm die Analyse dieser Variante:
Pro Spiel verliert A erwartet einen halben Euro und man kann die Strategie von B und C leicht weiter variieren, so dass es nicht so schnell auffällt ...

Oft zählen ZV's nur irgendwas, dann ist das Bild Teilmenge der natürlichen Zahlen und für diese gilt insbesondere:

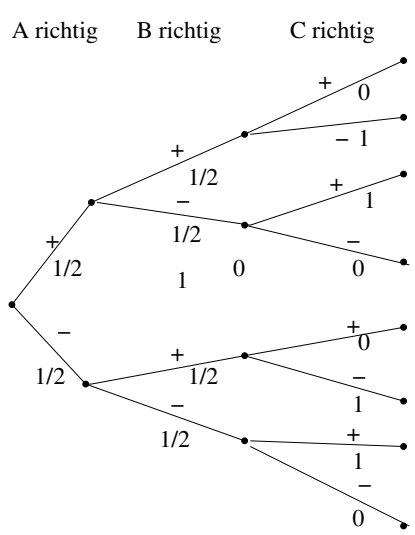
Fakt 2: Für eine ZV $X : \Omega \to \mathbb{N}$ gilt

$$E(X) = \sum_{i=1}^{\infty} i \cdot Pr(X = i)$$

Beweis: Folgt sofort aus Fakt 1. Für $i = 0$ ist der Summand 0. □

Fakt 3: Für eine ZV $X : \Omega \to \mathbb{N}$ gilt

$$E(X) = \sum_{i=0}^{\infty} Pr(X > i)$$



Wahrscheinlichkeit	Reingewinn von A in Euro
0	0
1/4	1
1/4	1
0	4
0	−2
1/4	−2
1/4	−2
0	0

E(A's Reingewinn)= 0(0) + 1(1/4)+1(1/4)+4(0+(−2)(0)+ (−2)(1/4)+ (−2)(1/4) + 0(0)= −1/2

Beweis: Das Ereignis $(X > i)$ ist disjunkte Vereinigung der Ereignisse $(X = i+1), (X = i+2), \ldots$. Also ist

$$Pr(X > i) = \sum_{j=i+1}^{\infty} Pr(X = j)$$

Nun schreiben wir $\sum_{i=1}^{\infty} i \cdot Pr(X = i)$ wie folgt in Form eines unendlichen Dreiecks:

$$\begin{array}{ccccccccc} Pr(X=1) & + & Pr(X=2) & + & Pr(X=3) & + & Pr(X=4) & + \ldots \\ & & Pr(X=2) & + & Pr(X=3) & + & Pr(X=4) & + \ldots \\ & & & & Pr(X=3) & + & Pr(X=4) & + \ldots \\ & & & & & & Pr(X=4) & + \ldots \\ & & & & & & & \ldots \end{array}$$

Die Summe der Summen der einzelnen Spalten ist $\sum_{i=1}^{\infty} i \cdot Pr(X = i)$ also nach Fakt 2 der $E(X)$. Die Summe der Summen über die einzelnen Zeilen ist $\sum_{i=0}^{\infty} Pr(X > i)$. Nach dem Prinzip des doppelten Abzählens ist dies auch $E(X)$. □

Beispiel 4 Ein Programm crasht am Ende jeder vollen Stunde mit Wahrscheinlichkeit p, falls es da noch läuft. Sei X die ZV, die die Stunden bis zum Crash zählt. Was ist $E(X)$?
Lösung: $E(X) = \sum_{i=0}^{\infty} Pr(X > i)$
Was ist $Pr(X > i)$? Der Crash erfolgt nach der i–ten Stunde genau dann, wenn es bis zur i–ten Stunde nicht gecrasht hat, also $Pr(X > i) = (1-p)^i$.
Damit ist $E(X) = \sum_{i=0}^{\infty} (1-p)^i = 1/p$, vgl. Summenformel einer geometrischen Reihe.□

Von zentraler Bedeutung für die Anwendungen ist die sogenannte **Linearität des Erwartungswertes**. Dahinter steckt folgender einfache Sachverhalt:

Seien $X, Y : \Omega \to \mathbb{R}$ beliebige ZV's und c eine reelle Zahl. Wir definieren neue ZV's $(X+Y)$ und (cX) durch

$$(X+Y)(\omega) = X(\omega) + Y(\omega)$$

$$(cX)(\omega) = c \cdot X(\omega)$$

Aus der Definition des Erwartungswertes folgt sofort (Nachrechnen!!):

$$E(X+Y) = E(X) + E(Y) \quad \text{und} \quad E(cX) = c \cdot E(X)$$

Man beachte, dass man beides kombinieren und auf endliche Summen mittels vollständiger Induktion verallgemeinern kann:

$$E(c_1X_1 + c_2X_2 + \ldots + c_nX_n) = c_1E(X_1) + c_2E(X_2) + \ldots + c_nE(X_n)$$

Beispiel 5: n Leute geben jeder seinen Hut an der Garderobe ab.Nach der Vorstellung bekommt jeder einen Hut zurück, aber eben zufällig. Was ist die erwartete Anzahl von Leuten, die ihre eigenen Hüte zurückbekommen?
Lösung: Sei X die ZV, die die Anzahl der eigenen Hüte zählt. Ω ist hier die Menge der Permutationen der Zahlen (Leute) von 1 bis n.
Wir schreiben $X = X_1 + X_2 + \ldots + X_n$. Dabei ist X_i eine ZV mit möglichen Werten 0 oder 1, die sagt, ob Person i ihren Hut bekommen hat. Dies geschieht mit Wahrscheinlichkeit $1/n$, also $E(X_i) = 1 \cdot (1/n)$ für jedes i.
Also ist $E(X) = E(X_1) + \ldots + E(X_n) = n \cdot (1/n) = 1$ □
Durch einen Wahrscheinlichkeitsraum (Ω, Pr) und eine ZV $X : \Omega \to \mathbb{R}$ wird, wie wir gesehen haben, auf dem Bild $Im(X)$ ein Wahrscheinlichkeitsmaß Pr_X induziert. Dies war definiert durch

$$Pr_X(x) = Pr(X = x) = \sum_{\omega \in \Omega, X(\omega) = x} Pr(\omega)$$

Neben der einfachen Gleichverteilung, die zum Beispiel beim Würfeln mit einem fairen Würfel und der Identität als ZV auf der Menge $\{1,2,3,4,5,6\}$ induziert wird, gibt es weitere Wahrscheinlichkeitsverteilungen, die man so kanonisch erhält.

Bernoulli–Verteilung:
Ein *Bernoulli–Experiment* ist ein Zufallsexperiment mit genau zwei möglichen Ergebnissen: Erfolg mit Wahrscheinlichkeit p und Misserfolg mit Wahrscheinlichkeit $1-p$. Dabei ist $0 \leq p \leq 1$ eine feste Zahl. Man denke etwa an einen Münzwurf.
Mit so einem Experiment kann man eine einfache ZV X assoziieren, die Erfolg die Zahl 1 und Misserfolg die 0 zuordnet. Die so auf $\{0,1\}$ induzierte Verteilung heißt *Bernoulli–Verteilung*.
Für den Erwartungswert einer Bernoulli–verteilten ZV gilt:

$$E(X) = 0 \cdot (1-p) + 1 \cdot p = p$$

Binomialverteilung:
Wir führen ein Bernoulli–Experiment n mal unabhängig durch, $n > 0$ eine feste Zahl. Sei X die ZV, die die Anzahl k von Erfolgen zählt. Also $Im(X) = \{0, 1, \ldots, n\}$. Die von

X induzierte Verteilung auf $Im(X)$ heißt *Binomialverteilung*. Genauer, sei $b(k,n,p)$ die Wahrscheinlichkeit, dass genau k Erfolge bei den n Bernoulli–Experimenten (mit Parameter p) eintreten. Es gilt

Satz:

$$b(k,n,p) = \binom{n}{k} p^k \cdot (1-p)^{n-k}$$

Beweis: Vollständige Induktion über n
Induktionsanfang: $n = 1$
Fall $k = 0$: $b(0,1,p) = \binom{1}{0} p^0 (1-p) 1 = 1-p$
Fall $k = 1$: $b(1,1,p) = \binom{1}{1} p^1 (1-p)^0 = p$
Induktionsschritt: Nehmen wir an, die Aussage gilt für n Experimente.
Das Ereignis, dass bei $n+1$ Experimenten genau k Erfolge dabei sind, zerfällt in zwei disjunkte Ereignisse:

- k Erfolge bei den ersten n Versuchen und Misserfolg beim $(n+1)$–sten Versuch
- $k-1$ Erfolge bei den ersten n Versuchen und Erfolg beim $(n+1)$–sten Versuch

Wegen der Unabhängigkeit der Versuche ergibt sich mit Induktionsannahme und Pascalschem Dreieck:

$$b(k,n+1,p) = b(k,n,p) \cdot (1-p) + b(k-1,n,p) \cdot p =$$

$$= \binom{n}{k} p^k \cdot (1-p)^{n-k+1} + \binom{n}{k-1} p^k \cdot (1-p)^{n-k+1} = \binom{n+1}{k} p^k (1-p)^{n+1-k}$$

□

Wir vergewissern uns, dass die $b(k,n,p)$ tatsächlich eine Wahrscheinlichkeitsverteilung auf $\{0,1,\ldots,n\}$ definieren:

$$\sum_{k=0}^{n} \binom{n}{k} p^k (1-p)^{n-k} = (p+1-p)^n = 1^n = 1$$

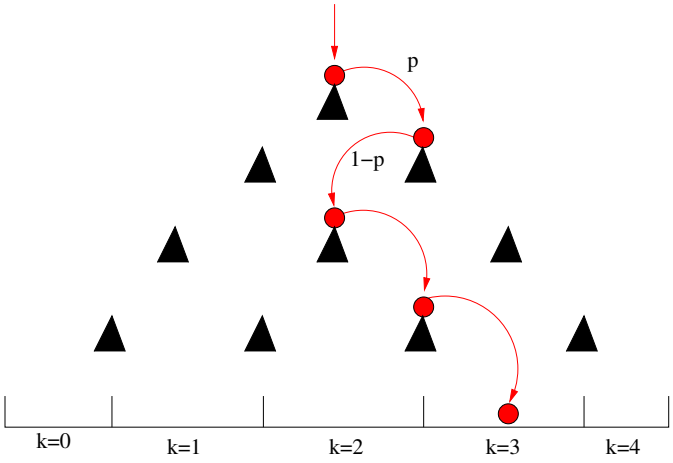
Wir haben den binomischen Satz benutzt.
Was ist $E(X)$?
Wir benutzen folgenden Standardtrick. Die ZV X lässt sich schreiben als Summe von Zufallsvariablen:

$$X = X_1 + X_2 + \ldots + X_n$$

X_i ist dabei die ZV, die das Ergebnis des i–ten Experiments beschreibt. X_i ist 1 bei Erfolg und 0 sonst. Damit ist für alle i X_i Bernoulli–verteilt und somit ist $E(X_i) = p$. Wegen der Linearität des Erwartungswertes ist

$$E(X) = E(X_1) + E(X_2) + \ldots + E(X_n) = n \cdot p$$

Man kann das Zufallsexperiment, dass zur Binomialverteilung führt, auch schön mit dem sogenannten Galton–Brett veranschaulichen: Bälle fallen auf die oberste Spitze und springen mit Wahrscheinlichkeit p nach rechts und mit $1-p$ nach links auf die nächste Spitze. Wiederholt man dies mit ausreichend vielen Bällen, so sieht man anhand des Füllstandes der einzelnen Fächer unten die typische Kurve der Binomialverteilung.



Geometrische Verteilung:
Wir betrachten ein Bernoulli–Experiment mit positiver Erfolgswahrscheinlichkeit $p > 0$, man denke an einen Münzwurf mit Kopf als Erfolg..
Nun führen wir ein neues Zufallsexperiment durch, das darin besteht, die Münze solange zu werfen, bis zum ersten Mal Kopf fällt. Elementarereignisse ω sind Folgen der Form $ZZZZ\ldots ZK$ bestehend aus $k-1$ mal Z für Zahl und einmal Kopf, mit $k > 0$. Einem solchen Elementarereignis ω weisen wir die Wahrscheinlichkeit zu:

$$Pr(\omega) = (1-p)^{k-1} \cdot p$$

Diese Elementarereignisse bilden einen (unendlichen!) Wahrscheinlichkeitsraum, denn

$$\sum_{k=1}^{\infty} (1-p)^{k-1} \cdot p = p \cdot \sum_{k=0}^{\infty} (1-p)^k = p \cdot \frac{1}{1-(1-p)} = 1$$

Dabei haben wir die Formel für die geometrische Reihe benutzt.
Sei X die ZV, die der Folge $Z^{k-1}K$ die Zahl k zuordnet. X induziert eine Wahrscheinlichkeitsverteilung auf der unendlichen Menge $\{1,2,3,\ldots\}$ durch $Pr_X(k) = (1-p)^{k-1} \cdot p$.
Was ist der Erwartungswert einer solchen geometrisch verteilten ZV?
Wir setzen $q = 1-p$ und erhalten:

$$E(X) = \sum_{k=1}^{\infty} k \cdot (1-p)^{k-1} \cdot p = \frac{p}{q} \cdot \sum_{k=1}^{\infty} k \cdot q^k$$

Nebenrechnung: Was ist $\sum_{k=1}^{\infty} k \cdot q^k$?
Wir kennen den Grenzwert der geometrischen Reihe für $0 \leq q < 1$

$$\sum_{k=0}^{\infty} q^k = \frac{1}{1-q}$$

Durch Differenzieren beider Seiten nach q erhalten wir

$$\sum_{k=1}^{\infty} k \cdot q^{k-1} = \frac{1}{(1-q)^2} = \frac{1}{p^2}$$

Jetzt noch mit q multiplizieren:

$$\sum_{k=1}^{\infty} k \cdot q^k = \frac{q}{p^2}$$

Damit ergibt sich für den Erwartungswert:

$$E(X) = \frac{p}{q} \cdot \frac{q}{p^2} = \frac{1}{p}$$

Bei Erfolgswahrscheinlichkeit 1/3 muss man also erwartet dreimal die Münze werfen, bis Erfolg sich einstellt.

5.5 Das Coupon–Collector–Problem

Dies ist eine schöne Anwendung für geometrisch verteilte ZV's.
Das Problem lässt sich am besten anhand der folgenden Fragen erklären:

- Wie oft muss man im Erwartungswert einen fairen Würfel werfen, bis man alle Augenzahlen gesehen hat?
- Wie viele zufällige Leute müssen sich im Erwartungswert versammeln, bis jeder der 365 Geburtstage im Jahr wenigstens einmal vertreten ist?
- Wie viele Bonbons muss man im Erwartungswert zufällig in ein Menge von r Kindern werfen, bis jedes Kind wenigstens ein Bonbon bekommen hat?

Wir stellen zunächst fest, dass alle drei Fragen einem Schema folgen. Man nennt es das *Coupon–Collector–Problem*. Wir entwickeln die Antwort anhand der letzten Frage:
Sei X die ZV, die die Anzahl der geworfenen Bonbons zählt.
Jedes Kind fängt ein geworfenes Bonbon mit Wahrscheinlichkeit $1/r$. Der Trick besteht nun darin, den gesamten Prozess in Phasen einzuteilen.
Sei t_i die Anzahl der geworfenen Bonbons, bis zum ersten Mal i verschiedene Kinder wenigstens einen Bonbon haben. Offensichtlich ist $t_0 = 0$ und $t_1 = 1$.
Sei X_i die ZV, die die geworfenen Bonbons vom $(t_{i-1}+1)$–sten bis zum t_i–ten Bonbon zählt. Offensichtlich ist

$$X = X_1 + X_2 + \ldots + X_r \text{ und } E(X) = \sum_{i=1}^{r} E(X_i)$$

Was ist $E(X_i)$? Die zentrale Beobachtung besteht darin, dass jedes X_i geometrisch verteilt ist. Was ist die Erfolgswahrscheinlichkeit des zugehörigen Bernoulli–Experiments?
Misserfolg tritt ein, wenn eines der $i-1$ Kinder, die schon etwas haben, wieder ein Bonbon bekommen. Das passiert mit Wahrscheinlichkeit $\frac{i-1}{r}$. Damit ist die Erfolgswahrscheinlichkeit
$1 - \frac{i-1}{r} = \frac{r-i+1}{r}$. Für den Erwartungswert ergibt sich $E(X_i) = \frac{r}{r-i+1}$
Insgesamt erhalten wir:

$$E(X) = \sum_{i=1}^{r} \frac{r}{r-i+1} = r \cdot \sum_{i=1}^{r} \frac{1}{i} = r \cdot H_r \approx r \ln r$$

Dabei ist H_r eine Bezeichnung für $\sum_{i=1}^{r} \frac{1}{i}$, sie heißt r–te *Harmonische Zahl* und wächst ungefähr wir der natürliche Logarithmus $\ln r$.
Für die beiden ersten Eingangsfragen heißt dies, man muss erwartet 14.7 mal würfeln bzw. es müssen 2364.6 Personen anwesend sein.

5.6 Ein Random Walk und eine randomisierte Strategie *

Aufgabe I: Alice und Bob stehen in einer Reihe. Zwischen ihnen stehen die Studenten $S_1, S_2, \ldots, S_k$. Das Gesamtbild ist also

$$A, S_1, S_2, \ldots, S_k, B$$

S_1 hat eine Flasche Wasser mitgebracht. Diese wird jeweils mit Wahrscheinlichkeit 0.5 an den linken bzw. rechten Nachbarn weitergegeben bei Start in S_1.
Was ist die Wahrscheinlichkeit p_k, dass Alice die Flasche vor Bob erhält?

Lösung:
Es sollte klar sein, dass die Chancen für Alice mit wachsendem k steigen.
Fall $k = 1$: A und B bekommen die Flasche mit Wahrscheinlichkeit 0.5, also $p_1 = 0.5$

Fall $k > 1$:

$$p_k = Pr(\text{erster Schritt nach links}) \cdot Pr(A \text{ hat Flasche vor } B| \text{ erster Schritt nach links}) +$$

$$+Pr(\text{erster Schritt nach rechts}) \cdot Pr(A \text{ hat Flasche vor } B|\text{erster Schritt nach rechts})$$

Damit haben wir:

$$p_k = \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot Pr(A \text{ hat Flasche vor } B| \text{ erster Schritt nach rechts })$$

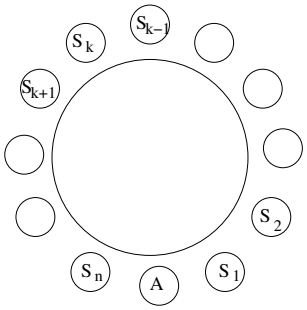
Nun der Trick: Wenn der erste Schritt nach rechts war, so hat S_2 die Flasche. Wenn sie dann A vor B erreichen soll, muss sie zuerst irgendwann wieder zu S_1. Die Wahrscheinlichkeit S_1 vor B zu erreichen ist p_{k-1}, denn dies ist unsere Ausgangsfrage mit S_1 in der Rolle von A. Angenommen, die Flasche ist wieder bei S_1 gelandet, so haben wir wieder die Wahrscheinlichkeit p_k Alice vor Bob zu erreichen. Also:

$$p_k = \frac{1}{2} + \frac{1}{2} \cdot p_{k-1} \cdot p_k$$

Nimmt man den Wert für p_1, rechnet die nächsten aus, so kommt man auf die leicht mit vollständiger Induktion zu verifizierende Formel:

$$p_k = \frac{k}{k+1}$$

Aufgabe II: Alice und n Studenten $S_1, S_2, \ldots, S_n$ sitzen in dieser zyklischen Ordnung an einem runden Tisch. Alice hat eine Flasche Wasser mitgebracht. Diese wird jeweils mit Wahrscheinlichkeit 0.5 an den linken bzw. rechten Nachbarn weitergegeben, wobei Alice anfängt. Es gewinnt derjenige Student S_k, der die Flasche als letzter zum ersten Mal bekommt. Berechne die Wahrscheinlichkeit, dass für ein gegebenes k der Student S_k gewinnt.
Lösung:
Die Frage, dass S_n (bzw. symmetrisch S_1) gewinnt ist einfach:



Wir schauen uns die Ausgangssituation an: A hat die Flasche und wir brechen die zyklische Ordnung bei S_n auf. Das Bild ist dann:

$$S_n, A, S_1, S_2, \ldots, S_{n-1}$$

S_n gewinnt, falls S_{n-1} die Flasche vor ihm bekommt. Nach der Analyse in obigem Teil I geschieht dies mit Wahrscheinlichkeit $1 - \frac{n-1}{n} = \frac{1}{n}$.

Wie sieht die Situation aus, in der S_k mit $1 < k < n$ gewinnt?
Die zentrale Einsicht ist: Beide seine Nachbarn müssen die Flasche vor ihm bekommen! Also S_{k-1} vor S_{k+1} oder umgedreht. Im Fall, dass S_{k+1} die Flasche zuerst hat, sieht die Situation in diesem Moment wie folgt aus

$$S_k, S_{k+1}, \ldots, S_n, A, S_1, \ldots, S_{k-1}$$

Diese Analyse kennen wir aus obiger Aufgabe I:
S_{k-1} bekommt die Flasche vor S_k mit Wahrscheinlichkeit $1 - p_{n-1} = \frac{1}{n}$. Insgesamt mit dem Satz über die totale Wahrscheinlichkeit:

$$Pr(\ S_k \text{ gewinnt }) = Pr(\ S_{k+1} \text{ vor } S_{k-1}\ ) \cdot Pr(\ S_k \text{ gewinnt} \mid S_{k+1} \text{ vor } S_{k-1}\ ) +$$

$$+ Pr(\ S_{k-1} \text{ vor } S_{k+1}\ ) \cdot Pr(\ S_k \text{ gewinnt} \mid S_{k-1} \text{ vor } S_{k+1}\ )$$

Dies ergibt:

$$Pr(\ S_k \text{ gewinnt }) = Pr(\ S_{k+1} \text{ vor } S_{k-1}\ ) \cdot \frac{1}{n} + Pr(\ S_{k-1} \text{ vor } S_{k+1}\ ) \cdot \frac{1}{n} = \frac{1}{n}$$

Das überraschende Fazit: Alle gewinnen mit der gleichen (!) Wahrscheinlichkeit $\frac{1}{n}$. $\square$

Was wir in diesen beiden Aufgaben analysiert haben, sind Eigenschaften von sogenannten *Random Walks*, hier die Bewegung der Flasche. Diese ist nicht determiniert, sondern zu jedem Zeitpunkt gibt es bestimmte Übergangswahrscheinlichkeiten zum nächsten Zustand des Prozesses.

Zum Abschluss des Abschnittes über diskrete Wahrscheinlichkeitstheorie ein überzeugendes Beispiel, das zeigt, wie man Zufallsexperimente beim Entwurf von Algorithmen einsetzen kann.

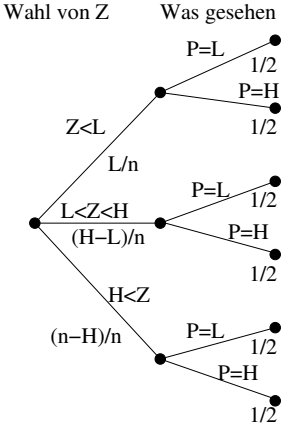
Problemstellung: Alice wählt zwei ganze Zahlen L, H aus dem Bereich $\{0,1,2,\ldots,100\}$ aus. Diese Zahlen sollen verschieden sein, $L < H$. Sie steckt jede Zahl in einen Umschlag und zeigt die beiden äußerlich nicht zu unterscheidenden Umschläge Bob. Seine Aufgabe ist es, auf den Umschlag mit der größeren Zahl zu zeigen!

An sich ist dies ein hoffnungsloses Unterfangen, denn Bob hat keinerlei Information. Reines Raten hat eine Gewinnchance von 50 Prozent. Nun erlaubt Alice Bob, in einen Umschlag seiner Wahl hineinzuschauen. Dort sieht Bob eine Zahl P. Seine Aufgabe ist es zu sagen, ob $P = L$ oder $P = H$ ist.
Gibt es jetzt eine Strategie mit Gewinnwahrscheinlichkeit >50 Prozent? 50 Prozent erreicht man schon ohne reinzuschauen.

Die etwas überraschende Antwort ist Ja. Zunächst muss man sich klar machen, dass Alice sich bei der Auswahl der Zahlen keinerlei Wahrscheinlichkeitsverteilung unterwirft, sie möchte das Spiel auch gewinnen.
Der Trick besteht darin, dass sich Bob bei seiner Antwort des Augangs eines Zufallsexperiments bedient, und so etwas nennt man eine *randomisierte Strategie.*

Bobs Strategie: Bob wählt zufällig und gleichverteilt eine Zahl $Z \in \{0.5, 1.5, 2.5, \ldots 99.5\}$, die er als Entscheidungshilfe benutzt, indem er annimmt, dass diese Zahl zwischen L und H liegt, also $L < Z < H$ gilt. Sieht er eine Zahl $P < Z$, so sagt er, P sei die kleinere Zahl L, und ansonsten sagt er, dass er die größere Zahl H gesehen hat.

Analyse: Wir nehmen an, der Grundbereich seien die ganzen Zahlen $\{0,1,2,\ldots,n\}$. Die Analyse unterscheidet danach, wie Z zum tatsächlichen Intervall (L,H) liegt: An den Kanten steht wie gehabt, was mit welcher Wahrscheinlichkeit passiert.



Bob gewinnt	Wkt
Nein	L/2n
Ja	L/2n
Ja	(H–L)/2n
Ja	(H–L)/2n
Ja	(n–H)/2n
Nein	(n–H)/2n

Für das Ereignis "Bob gewinnt" ergibt sich als Wahrscheinlichkeit:

$$Pr(\text{Bob gewinnt}) = \frac{L}{2n} + \frac{H-L}{2n} + \frac{H-L}{2n} + \frac{n-H}{2n} \geq \frac{1}{2} + \frac{1}{2n}$$

Dabei haben wir benutzt, dass $H - L \geq 1$ gilt, denn es sind ja verschiedene Zahlen. Für $n = 100$ sind Bobs Gewinnchancen also 50.5 Prozent!!!

5.7 Abzählen III: Lineare Rekursionsgleichungen

Beispiel 1: Das klassische Problem der "Türme von Hanoi" liefert für $n > 1$ die Rekursionsgleichung

$$f(n) = 2 \cdot f(n-1) + 1$$

mit der *Anfangsbedingung* $f(1) = 1$.
Die Aufgabe besteht darin für $f(n)$ eine *geschlossene Form* zu finden, also einen Ausdruck für $f(n)$, bei dem auf der rechten Seite nicht wieder auf andere Werte der Funktion f zugegriffen wird. Die Lösung ist hier

$$f(n) = 2^n - 1$$

Man überzeugt sich leicht mittels vollständiger Induktion davon, dass diese Funktion obige Anfangsbedingung und die Rekursionsgleichung erfüllt.
Wie findet man diese geschlossene Form?

Methode 1: Raten und Verifizieren
Diese Methode kann man zwar immer probieren, sie wird aber nur in einfachen Fällen zum Erfolg führen. Man rechnet einfach Werte von f für kleine n aus, rät die geschlossene Form und verifiziert sie mit vollständiger Induktion.
Hier ist

$$f(1) = 1, f(2) = 2 \cdot f(1) + 1 = 3, f(3) = 2 \cdot f(2) + 1 = 7, f(4) = 2 \cdot f(3) + 1 = 15, \text{ usw.}$$

Nun rät man $f(n) = 2^n - 1$ und verifiziert das. Fertig!

Methode 2: Einsetzen und Vereinfachen
Man setzt die Rekursionsvorschrift (rückwärts) immer wieder ein bis hin zur Anfangsbedingung. Nach jedem Einsetzen vereinfacht man den entstehenden Ausdruck und versucht dabei ein "Muster" zu erkennen. Im Beispiel

$$\begin{aligned}
f(n) &= 2 \cdot f(n-1) + 1 & \\
&= 2 \cdot (2f(n-2) + 1) + 1 & \text{Einsetzen!} \\
&= 4 \cdot f(n-2) + 3 & \text{Vereinfachen!} \\
&= 4 \cdot (2f(n-3) + 1) + 3 & \text{Einsetzen!} \\
&= 8 \cdot f(n-3) + 7 & \text{Vereinfachen!} \\
&= \text{usw.} & \\
&= 2^i \cdot f(n-i) + 2^i - 1 & \text{Muster erkannt} \\
&= 2^{n-1} \cdot f(1) + 2^{n-1} - 1 & \text{Anfangsbedingung nutzen!} \\
&= 2^{n-1} \cdot 1 + 2^{n-1} - 1 = 2^n - 1 &
\end{aligned}$$

Zum Schluss sollte man immer die Lösung nochmals verifizieren. Das Schwierige bei diesem Ansatz ist das Finden des Musters, insbesondere wenn man zuviel vereinfacht!

Beispiel 2 Die rekursiv definierte Folge der Fibonacci–Zahlen lautet

$$f(0) = 0,\ f(1) = 1,\ f(n) = f(n-1) + f(n-2) \ \text{ für } n > 1$$

Versucht man die geschlossene Form mittels Methode 1 oder 2 zu finden, so wird man kläglich scheitern.

Wir nehmen mal an, dass die Lösung der Rekursionsgleichung ohne die Anfangsbedingung die Form hat:

$$f(n) = c \cdot x^n \text{ für Parameter } c,x \in \mathbb{R}$$

Wenn diese Annahme stimmt, so müsste also gelten:

$$c \cdot x^n = c \cdot x^{n-1} + c \cdot x^{n-2}$$

Wir teilen durch $c \cdot x^{n-2}$ und erhalten

$$x^2 = x + 1$$

und damit sind die Lösungen für x

$$x_1 = \frac{1+\sqrt{5}}{2} \text{ und } x_2 = \frac{1-\sqrt{5}}{2}$$

Damit sind $f(n) = c_1 \cdot (\frac{1+\sqrt{5}}{2})^n$ und $f(n) = c_2 \cdot (\frac{1-\sqrt{5}}{2})^n$ für beliebige $c_1, c_2 \in \mathbb{R}$ Lösungen der Fibonacci–Rekursionsgleichung (ohne Randbedingungen), ebenso wie die Summe von Lösungen wieder eine Lösung ist. Wie findet man nun c_1 und c_2? Da helfen die Randbedingungen: Es muss gelten

$$f(0) = c_1 \cdot (\frac{1+\sqrt{5}}{2})^0 + c_2 \cdot (\frac{1-\sqrt{5}}{2})^0 = 0$$

$$f(1) = c_1 \cdot (\frac{1+\sqrt{5}}{2})^1 + c_2 \cdot (\frac{1-\sqrt{5}}{2})^1 = 1$$

Wir lösen dieses Gleichungssystem mit den Unbekannten c_1, c_2 und erhalten

$$c_1 = \frac{1}{\sqrt{5}} \quad c_2 = -\frac{1}{\sqrt{5}}$$

Die geschlossene Form für die Folge der Fibonacci–Zahlen lautet somit

$$f(n) = \frac{1}{\sqrt{5}} \cdot (\frac{1+\sqrt{5}}{2})^n - \frac{1}{\sqrt{5}} \cdot (\frac{1-\sqrt{5}}{2})^n$$

Anmerkungen:

1. Man rechne die Anfangswerte von $f(n)$ nach, um sich davon zu überzeugen, dass es tatsächlich ganze Zahlen sind!

2. Das Wachstum der Folge wird durch den ersten Term bestimmt, denn $|\frac{1-\sqrt{5}}{2}| < 1$ und damit strebt $(\frac{1-\sqrt{5}}{2})^n$ sehr schnell gegen 0. Also

$$f(n) \approx \frac{1}{\sqrt{5}} \cdot (\frac{1+\sqrt{5}}{2})^n$$

   Dies ist ein exponentielles Wachstum, denn $\frac{1+\sqrt{5}}{2} > 1$.

3. Die Zahl $\frac{1+\sqrt{5}}{2}$ heißt *Goldener Schnitt* und tritt in Mathematik und anderen Wissenschaften in verschiedensten Zusammenhängen immer wieder auf.

4. Definiert man die Fibonacci–Zahlen mit den Anfangsbedingungen $f(0) = 1, f(1) = 1$ so ergibt sich als geschlossene Form

$$f(n) = \frac{1}{\sqrt{5}} \cdot (\frac{1+\sqrt{5}}{2})^{n+1} - \frac{1}{\sqrt{5}} \cdot (\frac{1-\sqrt{5}}{2})^{n+1}$$

   Nachrechnen!

Definition Eine lineare homogene Rekursionsgleichung vom Grad d hat die Form

$$f(n) = a_1 \cdot f(n-1) + a_2 \cdot f(n-2) + \ldots + a_d \cdot f(n-d)$$

wobei die a_i reelle Koeffizienten sind, zusammen mit d Anfangsbedingungen wie etwa Werte für $f(0), f(1), \ldots, f(d-1)$.

Homogene lineare Rekursionsgleichungen werden in Verallgemeinerung des Fibonacci–Beispiels wie folgt gelöst.

Methode 3: Charakteristische Gleichung
Wir nehmen an, die geschlossene Form lautet $f(n) = x^n$ für eine reelle Zahl x. Damit ergibt sich:

$$x^n = a_1 \cdot x^{n-1} + \ldots + a_d \cdot x^{n-d}$$

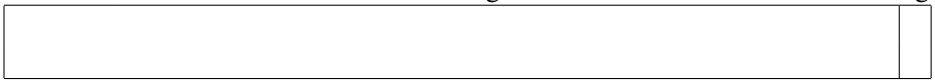
Nach dem Teilen durch x^{n-d} ergibt sich die *Charakteristische Gleichung*

$$x^d = a_1 \cdot x^{d-1} + \ldots + a_d$$

Diese Gleichung vom Grad d hat d Wurzeln. Es gilt

- Ist r eine einfache Wurzel, so ist r^n Lösung der Rekursionsgleichung (ohne Randbedingung).
- Falls r eine Wurzel mit Vielfachheit k ist, so sind $r^n, nr^n, n^2r^n, \ldots n^{k-1}r^n$ Lösungen der Rekursionsgleichung (ohne Randbedingung)
- Die Summe von mit Konstanten multiplizierten Lösungen (eine sogenannte Linearkombination) der Rekursionsgleichung ist wieder Lösung.

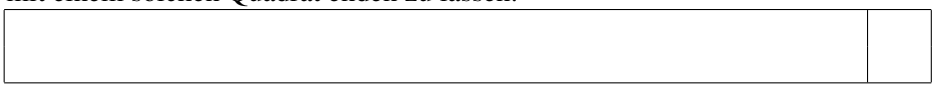
Beispiel 3· Sei A_n die Anzahl der Belegungen eines $2 \times n$–Rechtecks mit 1×2 Dominosteinen. Mit Anzahl der Belegungen ist gemeint, die Anzahl der Möglichkeiten das Rechteck voll zu belegen, z.B. $A_2 = 2$. Man bestimme die Rekursion für A_n und berechne A_n.
Wie ändert sich das Ergebnis, wenn auch 2×2–Steine zur Verfügung stehen?
Lösung: Um die Rekursionsformel für die Anzahl A_n der Belegung bei einem Rechteck zu finden, unterscheiden wir die zwei Möglichkeiten die letzten Steine hinzulegen.



Legt man den letzten Stein senkrecht hin, gibt es A_{n-1} Möglichkeiten das Rechteck zu füllen. Liegen am Ende des Rechtecks zwei Steine waagerecht, so gibt es A_{n-2} Möglichkeiten. Insgesamt also

$$A_n = A_{n-1} + A_{n-2}$$

Möglichkeiten. Durch Zählen der Möglichkeiten für $n = 1$ und $n = 2$ erhalten wir die Startwerte $A_1 = 1$ und $A_2 = 2$. Setzen wir $A_0 = 1$, stimmt dies mit der Folgendefinition überein und wir erkennen die obige Fibonacci–Folge aus Beispiel 2.
Lassen wir auch 2×2–Steine zu, gibt es zusätzlich noch A_{n-2} Möglichkeiten das Rechteck mit einem solchen Quadrat enden zu lassen.



Die Rekursionsformel ist dann $A_n = A_{n-1} + 2A_{n-2}$ und die Startwerte sind $A_1 = 1$ und $A_2 = 3$. Nach dem bekannten Schema lässt sich die geschlossene Lösung berechnen. Die charakteristische Gleichung ist

$$x^2 = x + 2$$

mit den Lösungen $x_1 = 2$ und $x_2 = -1$. Also ist $A_n = c_1 2^n + c_2(-1)^n$. Für $n = 1$ und $n = 2$ ergibt sich zusammen mit den Startwerten:

$$\begin{aligned} 1 &= 2c_1 - c_2 \\ 3 &= 4c_1 + c_2 \end{aligned}$$

Dieses Gleichungssystem hat die Lösung $c_1 = \frac{2}{3}$ und $c_2 = \frac{1}{3}$. Damit lautet die geschlossene Darstellung der Rekursion

$$A_n = \frac{2}{3}2^n + \frac{1}{3}(-1)^n$$

Beispiel 4 Eine charakteristische Gleichung habe die Wurzeln r_1, r_2, r_3, r_3. Das heißt, r_3 ist doppelte Nullstelle. Die allgemeine Lösung der Rekursionsgleichung lautet

$$f(n) = c_1 r_1^n + c_2 r_2^n + c_3 r_3^n + c_4 n r_3^n$$

Nun werden die Randbedingungen genutzt, um die Konstanten zu finden.

Beispiel 5: Sei $f(0) = 0, f(1) = 1$ und $f(n) = 2f(n-1) - f(n-2)$.
Es handelt sich um eine homogene lineare Rekursionsgleichung vom Grad 2.
Die charakteristische Gleichung lautet:

$$x^2 = 2x - 1$$

Damit ist $x = 1$ doppelte Nullstelle. Die Lösung der Rekursionsgleichung hat also die Gestalt

$$f(n) = c_1 \cdot 1^n + c_2 \cdot n \cdot 1^n = c_1 + c_2 n$$

Wir bestimmen die Konstanten mit Hilfe der Randbedingungen:

$$f(0) = 0 = c_1 + c_2 \cdot 0$$

$$f(1) = 1 = c_1 + c_2 \cdot 1$$

Dies gibt die Lösungen $c_1 = 0$ und $c_2 = 1$. Wir erhalten als geschlossene Form

$$f(n) = n$$

Dies hätte man mit Methode 1 sicher viel schneller gefunden ...

Beispiel 6: Eine Kreisscheibe ist in $n > 1$ nummerierte Sektoren unterteilt, die mit $m > 1$ Farben so eingefärbt werden sollen, dass keine zwei benachbarten Sektoren dieselbe Farbe bekommen.
Man stelle eine Rekursionsgleichung für die Anzahl der verschiedenen Färbungen auf und löse sie.

Lösung:
Der Rekursionsanker ist:

$$f(2) = m(m-1) \quad \text{und} \quad f(3) = m(m-1)(m-2)$$

Die Rekursionsformel lautet:

$$f(n) = (m-2)f(n-1) + (m-1)f(n-2)$$

Das sieht man wie folgt. Wir betrachten die letzten 3 Sektoren. Wenn diese 3 verschiedene Farben haben, dann entsteht durch das Weglassen des vorletzten Sektors eine zulässige Lösung für ein Rad mit $n-1$ Sektoren. Umgedreht, eine zulässige Färbung von $n-1$ Sektoren kann zur Lösung für n Sektoren dadurch erweitert werden, dass man eine von $m-2$ möglichen Farben für den vorletzten Sektor benutzt.
Wenn der drittletzte und der letzte Sektor die gleiche Farbe haben, so erhält man eine solche Färbung aus einer zulässigen Färbung von $n-2$ Sektoren, indem man den letzten Sektor mit seiner Farbe verdoppelt und dazwischen einen Sektor mit einer der $m-1$ möglichen restlichen Farben schiebt.
Die charakteristische Gleichung lautet:

$$x^2 = (m-2)x + (m-1)$$

Die Lösungen davon sind: $x_1 = m-1$ und $x_2 = -1$.
Demnach hat $f(n)$ die Gestalt:

$$f(n) = c_1(m-1)^n + c_2(-1)^n$$

Der Rekursionsanker liefert zwei Gleichungen, mit denen c_1 und c_2 bestimmt werden können. Diese sind:

$$m(m-1) = c_1(m-1)^2 + c_2 \quad \text{und} \quad m(m-1)(m-2) = c_1(m-1)^3 - c_2$$

Die Lösungen sind: $c_1 = 1$ und $c_2 = m-1$.
Damit ist die geschlossene Form der Rekursionsgleichung:

$$f(n) = (m-1)^n + (-1)^n(m-1)$$

Definition Eine lineare inhomogene Rekursionsgleichung vom Grad d hat die Form

$$f(n) = a_1 \cdot f(n-1) + a_2 \cdot f(n-2) + \ldots + a_d \cdot f(n-d) + g(n)$$

wobei die a_i reelle Koeffizienten sind, zusammen mit d Anfangsbedingungen wie etwa Werte für $f(0), f(1), \ldots, f(d-1)$ und einer Funktion $g(n)$, die nicht die Nullfunktion ist.

Hier ist das Kochrezept zur Lösung solcher inhomogener Rekursionsgleichungen anhand eines Beispiels.

Beispiel 7: Sei $f(1) = 1$ und $f(n) = 4f(n-1) + 3^n$. Dies ist eine inhomogene lineare Rekursion vom Grad 1 und $g(n) = 3^n$.
Schritt 1: Löse die homogene Rekursionsgleichung (ohne Randbedingung)

$$f(n) = 4f(n-1)$$

Die Lösung hat die Gestalt $f(n) = c \cdot 4^n$ mit $c \in \mathbb{R}$.
Schritt 2: Finde eine sogenannte spezielle Lösung für die inhomogene Rekursion (ohne Randbedingung)
Wir raten: $a \cdot 3^n$ für ein $a \in \mathbb{R}$ ist spezielle Lösung.

$$a \cdot 3^n = 4 \cdot a \cdot 3^{n-1} + 3^n$$

Das ergibt $a = -3$ und tatsächlich erfüllt $f(n) = -3 \cdot 3^n = -3^{n+1}$ die Rekursionsgleichung.
Schritt 3: Spezielle und homogene Lösung addieren und Konstanten finden!

$$f(n) = c \cdot 4^n - 3^{n+1}$$

Aus der Randbedingung folgt

$$f(1) = 1 = c \cdot 4 - 3^2 \text{ also } c = 5/2$$

Damit ist die geschlossene Form für unsere inhomogene Rekursionsgleichung

$$f(n) = \frac{5}{2} \cdot 4^n - 3^{n+1}$$

Anmerkungen: Der schwierigste Teil ist sicher das Finden einer speziellen Lösung in Schritt 2. Dazu die folgenden Tipps:

1. Versuche spezielle Lösung in der gleichen Form wie $g(n)$ zu finden!

2. Das heißt, falls $g(n) = const$, versuche zunächst $f(n) = c$, danach $f(n) = bn + c$, dann $f(n) = an^2 + bn + c$ usw.

3. Falls $g(n)$ Polynom ist, versuche zunächst ein Polynom vom gleichen Grad, dann ein Grad höher usw.

4. Falls $g(n)$ eine Exponentialfunktion wie im Beispiel 3^n ist, so versuche zunächst $f(n) = c \cdot 3^n$, dann $f(n) = b \cdot n \cdot 3^n + c \cdot 3^n$ usw.

Übung: Man löse das Eingangsbeispiel der Türme von Hanoi nach diesem Schema!

6 Graphentheorie

6.1 Einführung und Grundlagen

Ein *Graph* beschreibt Beziehungen (eine binäre Relation) zwischen den Elementen einer Menge von Objekten. Die Objekte werden als Knoten des Graphen bezeichnet; besteht zwischen zwei Knoten eine Beziehung, so sagen wir, dass es zwischen ihnen eine Kante gibt.

Definition Ein endlicher *ungerichteter* Graph G ist ein Paar (V,E) bestehend aus einer endlichen Knotenmenge V und einer Kantenmenge E von Knotenpaaren $e = \{u,v\}$, mit $u,v \in V$.
Im Gegensatz dazu ist endlicher *gerichteter Graph* G ein Paar (V,E) bestehend aus einer endlichen Knotenmenge V und einer Kantenmenge E von geordneten Knotenpaaren $e = (u,v)$ mit $u,v \in V$. Eine solche Kante (u,v) ist gerichtet von u nach v.

Anmerkungen:

1. Man beachte, dass für eine ungerichtete Kante gilt $\{u,v\} = \{v,u\}$, während im gerichteten Fall $(u,v) \neq (v,u)$.

2. Eine Kante von einem Knoten zu sich selbst wird als *Schleife (loop)* bezeichnet. Meistens werden wir schleifenlose Graphen betrachten und zusätzlich auch ausschließen, dass es zwischen einem Knotenpaar mehrere Kanten gibt (dann wäre die Kantenmenge durch eine *Multimenge* zu beschreiben). Solche Graphen ohne loops und Mehrfachkanten heißen *schlicht (simple)*.

3. Ungerichtete schlichte Graphen kann man auch definieren, indem man verlangt, dass die Kantenmenge E Teilmenge von $\binom{V}{2}$, der Menge aller 2–elementigen Teilmengen von V ist.

4. Alternativ kann man ungerichtete Graphen auch auffassen als gerichtete Graphen, in denen die zugrundeliegende binäre Relation symmetrisch ist, also mit jeder Kante (u,v) auch $(v,u) \in E$ gilt.

Darstellung von Graphen

1. **Graphische Darstellung**: Die Knoten werden als Punkte (meistens in der Ebene) gezeichnet, die Kanten als ungerichtete bzw. gerichtete Strecken, Streckenzüge bzw. Kurvenstücke, die die entsprechenden Punkte verbinden. Es gibt vielfältige andere Darstellungsformen mit dem Ziel, Graphen "schön" zu zeichnen; dies ist Gegenstand eines aktuellen Forschungsgebietes der Informatik: dem *Graph Drawing*.

2. **Adjazenzliste**: Ein Knoten v ist *adjazent (benachbart)* zu einem Knoten u, wenn es eine Kante von u nach v gibt.
Wir beschreiben den Graphen, indem wir für jeden Knoten alle seine adjazenten Knoten in einer verketteten Liste (als Datenstruktur) zusammenfassen.

3. **Adjazenzmatrix**: Wir nummerieren die Knoten aus V mit 1 bis n und bilden eine $n \times n$–Matrix A. Der Eintrag $A_{i,j}$ sei 1, wenn es eine Kante von i nach j gibt, und

0 sonst.
Man beachte, dass ungerichtete Graphen immer eine symmetrische Adjazenzmatrix haben.

Für die Behandlung algorithmischer Probleme sind in den allermeisten Fällen Adjazenzlisten die Datenstruktur der Wahl, weil sie die wahre Größe (d.h., den Speicherbedarf) eines Graphen (Anzahl der Kanten + Knoten) widerspiegeln, während Adjazenzmatrizen per Definition $|V|^2$ Einträge haben.

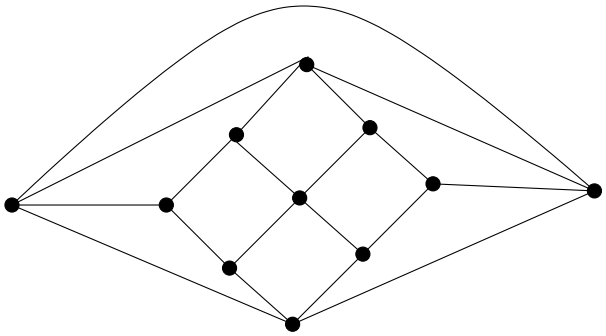
6.1.1 Beispiele für algorithmische Aufgabenstellungen

Wir wollen beispielhaft einige graphentheoretische und graphenalgorithmische Aufgaben skizzieren, die die Entwicklung der Graphentheorie stark beeinflusst haben.

1. **Färbungsprobleme**: Wir wollen die Knoten eines Graphen so einfärben, dass benachbarte Knoten verschiedene Farben bekommen. Diese Frage hat zum Beispiel folgende praktische Relevanz. Man denke an Radiosender, deren Empfangsgebiete sich überlappen. Man soll möglichst wenige Frequenzen (Farben) den Sendern so zuordnen, dass jeder Sender in seinem Einzugsgebiet störungsfrei zu empfangen ist.
   Historisch gesehen spielt in diesem Kontext "Das 4-Farben-Problem" eine wichtige Rolle: Man stelle sich die Welt (Kugeloberfläche) mit einer beliebigen politischen Landkarte vor. Wir definieren einen Graphen, indem wir jedem Land einen Knoten zuordnen und zwei Knoten mit einer Kante verbinden, wenn sie einen gemeinsamen Grenzabschnitt haben. Wie viele Farben braucht man, um die Länder so einzufärben, dass benachbarte Länder verschiedene Farben haben. Appel und Haken (1978) haben (mit Computerhilfe, 1200 CPU-Stunden) "bewiesen", dass vier Farben immer ausreichen! Einen Beweis völlig ohne Computer gibt es bis heute nicht.

2. **Eulersche Graphen**: Man entscheide für Graphen, ob man die Kanten so durchlaufen kann, dass man jede Kante genau einmal benutzt und am Schluss wieder am Startknoten steht. Der Ausgangspunkt für diese Frage war das von Euler 1736 gelöste sogenannte Königsberger Brückenproblem.

3. **Hamiltonsche Graphen**: Dies sind solche Graphen, die man so durchlaufen kann, dass man jeden Knoten genau einmal besucht bis man zum Ausgangsknoten zurückkehrt. Während man für das vorherige Entscheidungsproblem "Ist ein gegebener Graph Eulersch?" eine effiziente algorithmische Lösung kennt, ist das entsprechende Entscheidungsproblem hier algorithmisch schwer (NP-vollständig).



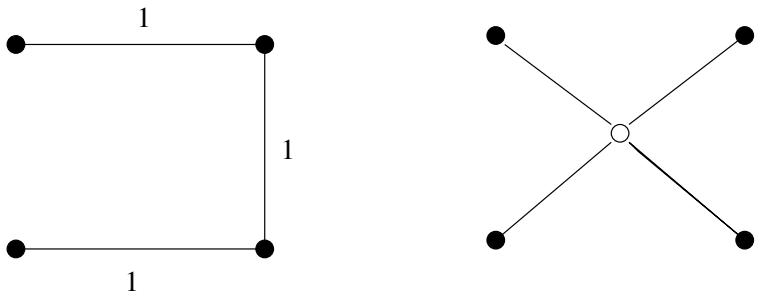
Ein Hamiltonscher Graph: Entfernt man die gekrümmte kante, dann nicht mehr!

4. **Travelling Salesperson Problem (TSP)**: Oft hat man es mit bewerteten Graphen zu tun, das heißt Kanten und/oder Knoten haben zusätzliche Informationen wie Gewichte, Längen, Farben etc.
   Ein Beispiel ist das TSP. Wir haben n Städte gegeben als Punkte in der Ebene. Für jedes Paar (u,v) von Städten kennt man die Kosten, um von u nach v zu kommen. Man entwerfe für den Handelsreisenden eine geschlossene Tour, die alle Städte besucht und minimale Gesamtkosten hat. Auch dies ist ein algorithmisch schweres Problem.

5. **Kürzeste Wege**: Ein Straßennetz kann man einfach als ungerichteten Graphen modellieren, in dem die einzelnen Kanten Längen haben. Ein Routenplaner muss dann bei Eingabe zweier Knoten A,B einen Weg (Kantenzug) mit minimaler Gesamtlänge von A nach B berechnen. Wir werden effiziente algorithmische Lösungen für dieses Problem kennenlernen.

6. **Planare Graphen**: Welche Graphen lassen sich so in der Ebene zeichnen, dass sich Kanten nicht schneiden, also sich höchstens in Knoten berühren? Wie kann man sie charakterisieren und algorithmisch schnell erkennen? Dazu lieferte Kuratowski (1931) eine wunderschöne Charakterisierung planarer Graphen durch verbotene Teilgraphen. Interessanterweise kann man planare Graphen immer so kreuzungsfrei zeichnen, dass die Kanten sogar Strecken sind (Fáry, 1948).

7. **Netzwerke**: Gegeben seien n Punkte in der Ebene. Wir wollen einen Graphen auf dieser Knotenmenge so konstruieren, dass jeder Punkt von jedem über einen Kantenzug erreichbar ist und die Summe der Kantenlängen minimal ist.
   Variante 1: (Minimaler aufspannender Baum) Der Graph hat genau die gegebenen Punkte als Knoten. Diese Frage hat effiziente algorithmische Lösungen!
   Variante 2: (Steinerbaumproblem) Man kann in der Lösung zusätzliche Verzweigungsknoten benutzen. Diese Variante ist algorithmisch schwer!!



Ein minimal aufspannender Baum der Gesamtlänge 3 und ein Steinerbaum mit Länge ~2.84

6.1.2 Grundlegende Begriffe

Sei im folgenden $G = (V, E)$ ein schlichter ungerichteter Graph.

Definition: Der *Grad* eines Knoten v in einem ungerichteten Graphen ist die Anzahl $deg_G(v)$ seiner Nachbarn.

In einem gerichteten Graphen bezeichnet $indeg_G(v)$ die Anzahl der in v ankommenden Kanten, den *Ingrad*; während $outdeg_G(v)$ die Anzahl der in v startenden Kanten angibt.

Es gilt der folgende Sachverhalt, auch als Handschlag-Lemma bekannt.

Satz: Für einen ungerichteten schlichten Graphen $G = (V, E)$ haben wir

$$\sum_{v \in V} deg_G(v) = 2 \cdot |E|$$

Beweis: Wir benutzen das Prinzip des doppelten Abzählens. Wir betrachten eine Matrix I (Inzidenzmatrix), deren Zeilen mit den Knoten indiziert sind, während die Spalten den Kanten entsprechen. Wir setzen den Eintrag $I_{i,j} = 1$, falls die j–te Kante den i–ten Knoten enthält und ansonsten sei der Eintrag 0. Wieviele Einsen hat die Matrix? Das können wir auf zwei Arten abzählen: In jeder Spalte stehen genau zwei Einsen, also insgesamt $2|E|$. In der Zeile des Knotens v stehen genau $deg_G(v)$ Einsen, also insgesamt $\sum_{v \in V} deg_G(v)$ Einsen. □
Der Satz hat ein einfaches aber überraschendes Korollar.

Korollar: In jedem ungerichteten Graphen ist die Anzahl der Knoten mit ungeradem Grad gerade.

Beweis: Die Summe aller Knotengrade ist gerade. Die Teilsumme der geraden Knotengrade ist auch gerade, also ist der Rest, die Summe der ungeraden Knotengrade, auch gerade. Das heißt dieser Rest muss eine gerade Anzahl von Summanden haben. □

Definition: Ein Graph $G' = (V', E')$ ist *Untergraph* von $G = (V, E)$, falls $V' \subseteq V$ und $E' \subseteq E$ ist. G' heißt *induzierter Untergraph*, falls außerdem gilt $E' = E \cap \{\{u, v\} | u, v \in V'\}$.

Definition: Zwei Graphen $G = (V, E)$ und $G' = (V', E')$ heißen *isomorph*, wenn es eine Bijektion $\phi : V \to V'$ gibt mit der Eigenschaft $\forall u, v \in V : \{u, v\} \in E \Leftrightarrow \{\phi(u), \phi(v)\} \in E'$.

Das Testen, ob zwei Graphen isomorph sind, ist im Allgemeinen (vermutlich) algorithmisch schwer (tatsächlich ist der Komplexitätsstatus nicht bekannt), für eingeschränkte Klassen wie z.B. die planaren Graphen kennt man effiziente Algorithmen.

Es gibt verschiedene Operationen, die aus Graphen neue Graphen erzeugen, wie Vereinigung, Durchschnitt aber auch zum Beispiel die Komplementbildung.

Definition: Der schlichte Graph $G^c = (V, E^c)$ ist das Komplement des schlichten Graphen $G = (V, E)$, falls für jedes $u, v \in V$ gilt, dass $\{u, v\} \in E$ genau dann wenn $\{u, v\} \notin E^c$.

Wir lernen im Folgenden einige wichtige, weil zumindestens in der Theorie häufig auftretende Graphen kennen.

1. Der *vollständige Graph* K_n, $n \geq 1$ eine natürliche Zahl, besteht aus n Knoten und allen möglichen $\binom{n}{2}$ Kanten.
   Jeder Graph ist natürlich Untergraph eines vollständigen Graphen.

2. Der *vollständige bipartite Graph* $K_{n,m}$, mit $n, m \geq 1$, besteht aus $n + m$ Knoten und allen $n \cdot m$ Kanten, die jeweils einen der n Knoten mit einem der m Knoten verbinden.
   Untergraphen eines vollständigen bipartiten Graphen heißen *bipartit*.

3. Der *Hyperwürfel* Q_n hat als Knoten alle 0-1-Folgen der Länge n, zwei Folgen werden durch eine Kante verbunden, wenn sie sich genau an einer Stelle unterscheiden, also Hamming–Abstand 1 haben.
   Der Q_n hat 2^n Knoten und $n \cdot 2^{n-1}$ Kanten. Diese Anzahl der Kanten folgt sofort aus dem Handschlaglemma, da jeder der 2^n Knoten den Grad n hat.

4. Der *Weg* P_n, $n \geq 0$, besteht aus $n+1$ Knoten $v_1, \ldots, v_{n+1}$ und den n Kanten $\{v_i, v_{i+1}\}$ für $1 \leq i \leq n$. Als Länge von P_n bezeichnen wir die Anzahl n seiner Kanten.

5. Der *Kreis* C_n, $n \geq 3$, entsteht aus dem Weg P_{n-1} durch das Hinzufügen der Kante $\{v_1, v_n\}$. Die Länge von C_n sei wieder die Anzahl der Kanten, also n.

6. Ein Graph heißt *k-regulär*, wenn alle Knoten den Grad k haben.
   Alle Kreise sind also 2–regulär, der K_n ist $(n-1)$–regulär und der Q_n ist n–regulär.

7. *Bäume*, also zusammenhängende Graphen ohne Kreise, sind die in der Informatik am häufigsten auftretenden Graphen.

6.2 Zusammenhang und Abstand in ungerichteten Graphen

Definition: Seien u und v Knoten in einem ungerichteten Graphen $G = (V, E)$. Wir sagen, dass v von u *erreichbar* ist, wenn es in G einen Weg als Untergraphen gibt, der u mit v verbindet.

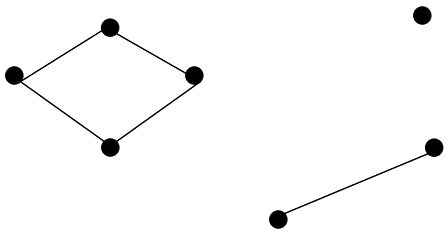
Erreichbarkeit ist eine binäre Relation über der Knotenmenge V. Offensichtlich ist diese Relation eine Äquivalenzrelation, denn sie ist:

- **reflexiv**: u ist mit sich selbst verbunden (durch den P_0);
- **symmetrisch**: Der Weg, der u mit v verbindet, verbindet auch v mit u;
- **transitiv**: Seien u, v, w Knoten und $u = u_1, \ldots, u_k = v$ die Knoten eines Weges von u nach v sowie $v = v_1, \ldots, v_l = w$ die Knoten eines Weges von v nach w. Sei i der kleinste Index mit $u_i \in \{u_1, \ldots, u_k\} \cap \{v_1, \ldots, v_l\}$. Wir haben $u_i = v_j$ für ein $1 \leq j \leq l$ und die Knoten $u_1, \ldots u_i, v_{j+1}, \ldots, v_l$ definieren einen Weg von u nach w.

Die Erreichbarkeitsrelation zerlegt also die Knotenmenge V in Äquivalenzklassen $V_1, \ldots, V_k$. Die von diesen Mengen induzierten Untergraphen heißen *Zusammenhangskomponenten* des Graphen G.

Definition: Ein ungerichteter Graph heißt *zusammenhängend*, wenn er genau eine Zusammenhangskomponente hat.

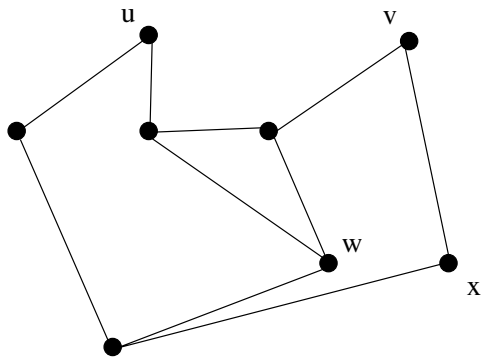
Beispiel:



Dieser Graph hat 3 Zusammenhangskomponenten

Definition: Seien u, v Knoten in einem ungerichteten Graphen $G = (V, E)$. Sind u und v in einer gemeinsamen Zusammenhangskomponente von G, so definieren wir ihren *Abstand* $d_G(u, v)$ als Länge (Anzahl der Kanten) eines kürzesten Weges von u nach v in G.
Gehören sie zu verschiedenen Komponenten, so existiert kein Weg zwischen u und v und wir setzen $d_G(u, v) = \infty$.
Der *Durchmesser* $D(G)$ des Graphen ist definiert als das Maximum über alle paarweisen Abstände zwischen Knoten.

Beispiel:



d(u,w)=2 , d(u,v)=3 , D(G)=3

6.3 Charakterisierung bipartiter Graphen

Es erweist sich in vielen Fällen als nützlich, mehrere äquivalente Charakterisierungen ein und derselben Graphklasse zu haben. Im Falle der bipartiten Graphen, die wir als Untergraphen der vollständigen bipartiten Graphen eingeführt hatten, liefert dies der folgende Satz.

Satz: Ein Graph ist genau dann bipartit, wenn alle in ihm als Untergraph enthaltenen Kreise gerade Länge haben.

Beweis: Zunächst überlegt man sich, dass wir den Graphen als zusammenhängend voraussetzen können, ansonsten führt man den folgenden Beweis für jede Zusammenhangskomponente.
Sei $G = (V,E)$ bipartit, das heißt, $V = A \cup B$ mit $A \cap B = \emptyset$ und Kanten verbinden nur Knoten aus A mit Knoten aus B. Sei des weiteren C ein Kreis in G. C benutzt abwechselnd Knoten aus A und B und hat somit gerade Länge.
Wir zeigen die andere Richtung. Wir fixieren einen beliebigen Knoten $u \in V$. Wir definieren:

$$A = \{v \in V |\ d(u,v) \text{ gerade }\}, B = V \setminus A$$

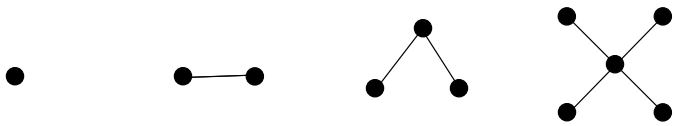
Die Knoten in B haben also ungeraden Abstand von u. Zu zeigen ist: Es gibt keine Kanten zwischen Knoten aus A (bzw. zwischen Knoten aus B).
Wir führen einen indirekten Beweis.
Wir nehmen an, es gibt eine Kante $\{w_1,w_2\}, w_1,w_2 \in B$ (für A analog) und finden einen Widerspruch zur Annahme, dass alle Kreise gerade Länge haben.
Wir betrachten kürzeste Wege von u zu w_1 und zu w_2. Diese Wege haben die gleiche ungerade Länge! (wegen der Kante zwischen w_1 und w_2) Sei z der letzte gemeinsame Knoten auf beiden Wegen. Dann bilden die beiden Wegabschnitte von z nach w_1 bzw. nach w_2 zusammen mit der Kante $\{w_1,w_2\}$ einen Kreis ungerader Länge $2d_G(z,w_1)+1$.
Widerspruch! □

Man beachte, dass der Satz insbesondere gilt für Graphen, die gar keine Kreise besitzen, also Bäume und Wälder.

6.4 Bäume und ihre Charakterisierung

Definition: Ein zusammenhängender ungerichteter Graph ist ein *Baum*, wenn er keinen Kreis enthält.
Ein ungerichteter Graph, dessen Zusammenhangskomponenten Bäume sind, heißt *Wald*.
Ein Baum zusammen mit einem ausgezeichneten Knoten (*der Wurzel*) heißt gewurzelter Baum.

Beispiel:



Baumbeispiele

Definition: Sei $G = (V, E)$ ungerichtet und zusammenhängend mit $|V| = n$. Ein Untergraph T auf allen n Knoten, der ein Baum ist, heißt *aufspannender Baum* von G.

Analog definiert man *aufspannende Wälder* für allgemeine ungerichtete Graphen. Diese bestehen aus aufspannenden Bäumen für die einzelnen Zusammenhangskomponenten.

Beobachtung: Jeder zusammenhängende Graph hat einen aufspannenden Baum, dieser ist aber nur eindeutig, wenn der Graph selbst ein Baum ist. Dann ist T der Graph selbst.□

Satz: Folgende Aussagen sind äquivalent für $G = (V, E)$.

(1) $G = (V, E)$ ist ein Baum.

(2) Je zwei Knoten sind durch genau einen Weg verbunden.

(3) G ist zusammenhängend und es gilt: $|E| = |V| - 1$.

Beweis: $(1) \Rightarrow (2)$ und $(2) \Rightarrow (1)$ sind einfache indirekte Schlüsse. Die erste Implikation sieht man z.B. wie folgt. Angenommen es gibt zwei Knoten u, v, zwischen denen nicht genau ein Weg verläuft. Dies könnte kein Weg sein, dann ist der Graph nicht zusammenhängend, oder es gibt mindestens zwei Wege, dann entsteht ein Kreis. In jedem Fall ist aber G kein Baum.
Wir zeigen $(1) \Rightarrow (3)$:
Zunächst hat jeder Baum Knoten vom Grad 1, diese nennt man *Blätter*. Das sieht man wie folgt. Seien $u_1, u_2, \ldots, u_i$ die Knoten eines längsten Weges in G. Alle Nachbarn von u_1 liegen auf diesem Weg, sonst wäre er nicht längster Weg. Nur u_2 kann Nachbar sein, sonst gäbe es einen Kreis.
Wir entfernen u_1 und die Kante $\{u_1, u_2\}$ aus G und erhalten einen zusammenhängenden Restgraphen G'. Dieser hat genau einen Knoten und eine Kante weniger als G. Wenn wir dies iterieren, bleibt zum Schluss genau ein Knoten ohne Kanten übrig. Also $|E| = |V| - 1$.

$(3) \Rightarrow (1)$: Sei $T = (V, E')$ aufspannender Baum von G, damit $|V| - |E'| = 1$. Aber nach Vorraussetzung gilt auch $|V| - |E| = 1$. Allerdings ist $E' \subseteq E$ und die einzige Möglichkeit hierfür ist $E = E'$. □

6.5 Grundlegende graphentheoretische Algorithmen *

Neben der Untersuchung struktureller Eigenschaften von Graphklassen ist es für praktische Anwendungen eine zentrale Aufgabe, möglichst effiziente algorithmische Lösungen zu finden zur Bestimmung graphentheoretischer Parameter und Eigenschaften.

Wie bestimmt man den Abstand zwischen Knoten eines Graphen, wie testet man, ob er zusammenhängend ist, und welche Datenstrukturen eignen sich dafür?

6.5.1 Graphdurchmustern: Breitensuche und Tiefensuche

Im Folgenden wollen wir Graphen systematisch von einem Startknoten aus durchmustern, das heißt, alle Knoten und Kanten 'anschauen'.
Sei $G = (V, E)$ ein ungerichteter (oder gerichteter) Graph gegeben durch seine Adjazenzlistendarstellung.

Breitensuche BFS

Die Breitensuche (breadth first search) startet in $s \in V$. Wir schauen uns zuerst alle Nachbarn von s an, danach die Nachbarn der Nachbarn usw. bis wir alle Knoten und Kanten erreicht haben.

Wir geben den Knoten 'Farben', diese symbolisieren ihren aktuellen Zustand:

- weiß: Knoten wurde noch nicht gesehen; zu Beginn sind alle Knoten weiß
- grau: Knoten wurde schon gesehen, wir müssen aber noch überprüfen, ob er noch weiße Nachbarn hat
- schwarz: Knoten ist erledigt, Knoten selbst und alle seine Nachbarn wurden gesehen.

Die noch zu untersuchenden grauen Knoten werden in einer Warteschlange Q verwaltet. Als Datenstruktur ist eine Warteschlange dadurch charakterisiert, dass Objekte in einer linearen Ordnung gehalten werden, neue Objekte kann man nur am Ende einfügen und zugreifen bzw. entfernen kann man nur den Kopf der Schlange, also das Objekt, was am längsten in der Schlange war. Es wird das FIFO-Prinzip umgesetzt: First–In–First–Out.
Knoten, deren Farbe von weiß nach grau wechselt, werden ans Ende der Schlange eingefügt. Die Schlange wird vom Kopf her abgearbeitet. Ist die Schlange leer, sind alle von s erreichbaren Knoten erledigt.
Mit BFS kann der Abstand $d[u] = d_G(s, u)$ eines erreichbaren Knotens u von s berechnet werden (Beweis später). Intuitiv sollte dies aber klar sein, denn der Algorithmus besucht zunächst alle Knoten im Abstand i von s, danach erst alle im Abstand $i + 1$. Gleichzeitig wird ein Baum von kürzesten Wegen von s zu allen erreichbaren Knoten aufgebaut. Dieser ist dadurch beschrieben, dass man für jeden Knoten einen Zeiger $\pi[u]$ auf den

Vorgängerknoten auf einem kürzesten Weg von s nach u aufrechterhält. Ist dieser noch nicht bekannt oder existiert gar nicht, so ist der Zeiger auf *NIL* gesetzt.
Im folgenden Pseudocode dienen die Zeilen 01 bis 08 der Initialisierung.

Breitensuche BFS(G,s):

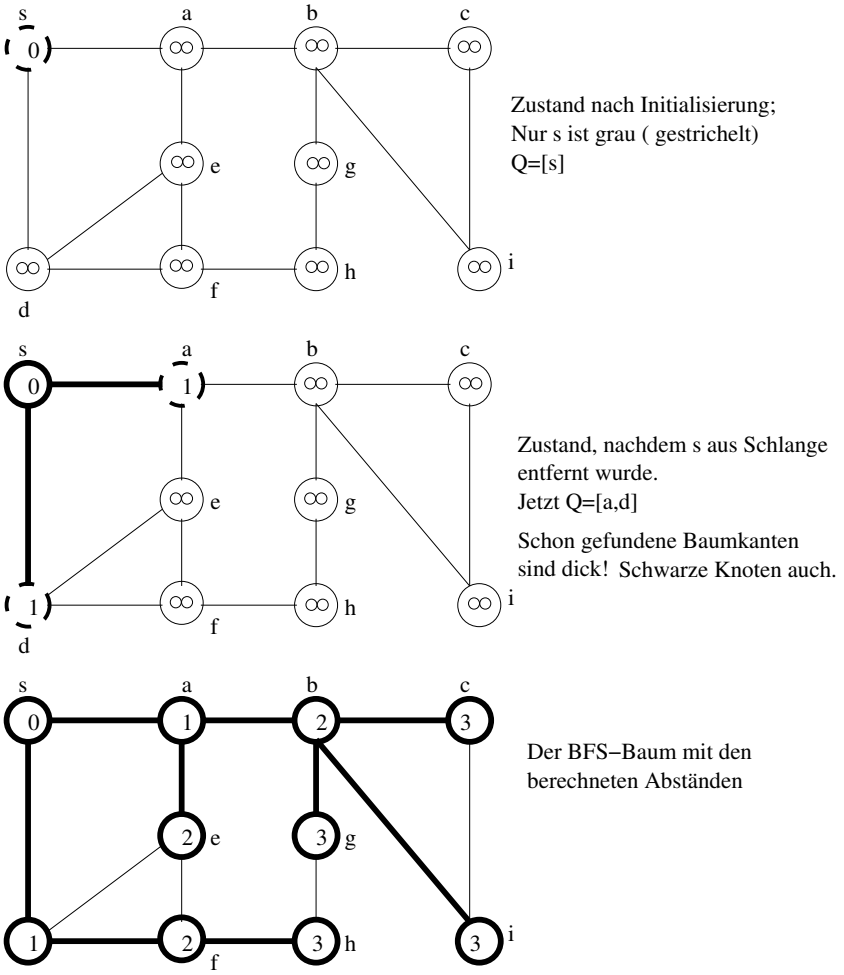
```
01 for jede Ecke u ∈ V(G)\{s}
02     do Farbe[u] ← weiß
03        d[u] ← ∞
04        π[u] ← NIL
05 Farbe[s] ← grau
06 d[s] ← 0
07 π[s] ← nil
08 Q ← {s}
09 while Q≠ ∅
10       do u ← Kopf[Q]
11          for jeden Nachbarn v ∈ Adj[u]
12              do if Farbe[v] = weiß
13                    then Farbe[v] ← grau
14                         d[v] ← d[u]+1
15                         π[v] ← u
16                         Setze v ans Ende von Q
17          Entferne Kopf aus Q
18          Farbe[u] ← schwarz
```

Die Komplexität des BFS–Algorithmus ist offensichtlich $O(|V|+|E|)$. Man beachte, dass der konstruierte Baum kürzester Wege von der Reihenfolge der Knoten in den Adjazenzlisten abhängt, der Abstand der Knoten selbst natürlich nicht. Genauer, verschiedene Adjazenzlistendarstellungen ein und desselben Graphen können nichtisomorphe Breitensuchbäume produzieren.

Ist der Graph G ungerichtet und zusammenhängend, so liefert BFS einen aufspannenden Baum. Hat G mehrere Zusammenhangskomponenten, so kann man leicht einen aufspannenden Wald erzeugen. Man startet die BFS–Suche erneut bei einem beliebigen noch weißen Knoten.

Was kann man damit noch anfangen? Auch der Test, ob ein ungerichteter Graph bipartit (also 2-färbbar) ist, ist jetzt sehr einfach. Wir konstruieren einen aufspannenden Wald, färben dessen Knoten mit zwei Farben und testen abschließend, ob die Graphkanten, die nicht im Wald sind, korrekt 2–gefärbt sind.

Beispiel: Im Beispiel werden die Adjazenzlisten als lexikographisch sortiert angenommen.



Wir wollen jetzt zeigen, dass man mit BFS korrekt die graphentheoretischen Abstände in G vom Startknoten s berechnet, also für alle Knoten v gilt: $d[v] = d_G(s,v)$.

Vorbetrachtung: Folgende Beobachtungen werden nützlich sein.

- Falls v später als u in die Warteschlange aufgenommen wird, so gilt für die berechneten Werte $d[u] \leq d[v]$.
- Zu jedem Zeitpunkt während des Algorithmus gilt für jeden Knoten v, dass $d[v] \geq d_G(s,v)$.
- Zu jedem Zeitpunkt gilt für die Warteschlange, dass die Differenz der $d[\,]$–Werte des letzten Knoten und des Kopfes der Queue ≤ 1 ist.

Beweis der BFS–Korrektheit:

Wir führen eine Induktion über den graphentheoretischen Abstand $d_G(s,v)$ durch.
Induktionsanfang: Falls $d_G(s,v) = 0$ so ist $v = s$ und laut Initialisierung $d[s] = 0$.

Induktionsannahme: Aussage richtig für alle Knoten u mit $d_G(s,u) < d_G(s,v)$
Induktionsschritt: Sei Knoten z Vorgänger von v auf einem tatsächlich kürzestem Weg von s nach v. Nach Annahme ist $d[z] = d_G(s,z)$ und

$$d_G(s,v) = d_G(s,z) + 1 = d[z] + 1$$

Entsprechend der Vorbetrachtung ist also $d[v] \geq d_G(s,v) > d[z]$. Also wird z vor v aus der Schlange entfernt. Wir betrachten den Zeitpunkt, bei dem z Kopf ist.

Fall 1: v ist in diesem Moment schon in Q. Dann ist

$$d[v] \leq d[z] + 1 = d_G(s,z) + 1 = d_G(s,v)$$

und wegen der Vorbetrachtung haben wir $d[v] = d_G(s,v)$.

Fall 2: v wird erst jetzt wegen der Kante zwischen z und v in Q aufgenommen und zwar mit dem Wert $d[v] = d[z] + 1 = d_G(s,v)$, was zu zeigen war. □

Bemerkung: Der eigentliche, tiefere Grund, weshalb dieser Beweis funktioniert, ist die banale Feststellung, dass ein kürzester Weg (hier von s zu v) als Teilwege wieder kürzeste Wege enthält (hier von s nach z). Dies werden wir bei anderen Kürzeste–Wege–Problemen wiederfinden.

Tiefensuche DFS

Die Idee der *Tiefensuche* (depth first search) ist einfach. Hat ein Knoten, den man besucht, noch unentdeckte Nachbarn, so geht man zum ersten solchen Nachbarn, den man findet, und von dort in die 'Tiefe' zu einem noch unentdeckten Nachbarn des Nachbarn, falls es ihn gibt. Das macht man rekursiv, bis man nicht mehr in die Tiefe gehen kann. Dann geht man solange zurück (backtracking–Schritte), bis wieder eine Kante in die Tiefe geht, bzw. alles Erreichbare besucht wurde.

Die Farben haben wieder dieselbe Bedeutung wie beim BFS, ebenso die π–Zeiger, die die entstehende Baumstruktur (genauer Wald-) implizit speichern. Der DFS berechnet nicht die Abstände vom Startknoten.

Man kann aber jedem Knoten ein Zeitintervall zuordnen, indem er 'aktiv' ist, was für verschiedene Anwendungen interessant ist. Dazu läßt man eine globale Uhr mitlaufen, und merkt sich für jeden Knoten u den Zeitpunkt $d[u]$, bei dem u entdeckt wird und den Zeitpunkt $f[u]$, bei dem u erledigt ist, da der Algorithmus alles in der 'Tiefe' unter u gesehen hat. Daher gilt für beliebige zwei Knoten im Graphen, dass entweder eines der Zeitintervalle voll im anderen enthalten ist, oder beide disjunkt sind.

Die geeignete Datenstruktur, um DFS zu implementieren, ist ein *stack (Kellerspeicher, Stapel)*. Dieser wird hier implizit angelegt, um die Rekursionsaufrufe zu verwalten.
Ist der Graph in Adjazenzlistenform gegeben, so läuft DFS ebenfalls in Zeit $O(|V|+|E|)$.

Wie beim BFS hängen die entstehenden DFS–Bäume von der Reihenfolge in den Adjazenzlisten ab. Verschiedene Reihenfolgen können zu nichtisomorphen DFS–Bäumen führen.

DFS(G)

01 **for** *jede Ecke u ∈ V(G)*
02 **do** *Farbe[u] ← weiß*
03 π*[u] ← NIL*
04 Zeit ← 0
05 **for** *jede Ecke u∈V(G)*
06 **do if** *Farbe[u] = weiß*
07 *DFS-visit(u)*

DFS-visit(u)

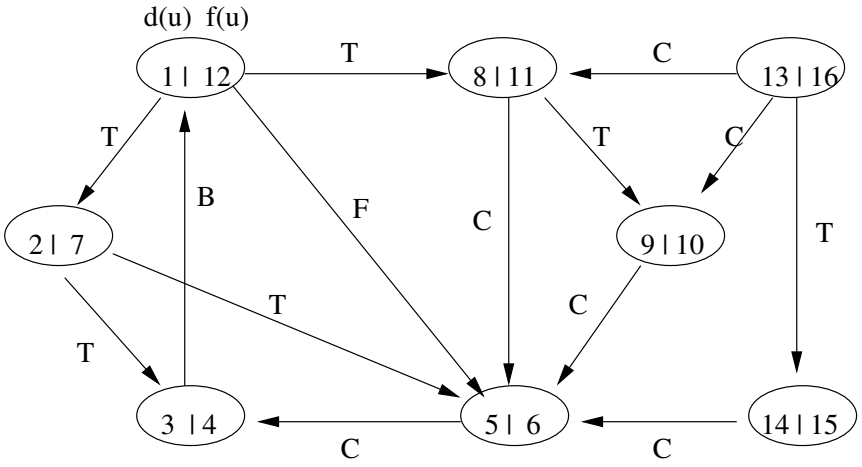
01 Farbe[u] ← grau
02 d[u] ← Zeit ← Zeit+1
03 **for** *jede Ecke v ∈ Adj[u]*
04 **do if** *Farbe[v] = weiß*
05 **then** π*[v] = u*
06 *DFS-visit(v)*
07 Farbe[u] ← schwarz
08 f[u] ← Zeit ← Zeit+1

Der Stack ist eine Datenstruktur, die das LIFO–Prinzip (*last–in–first–out*) umsetzt. Das heißt, die zu speichernden Daten (hier die grau werdenden Knoten) sind linear angeordnet und man kann ein neuen Eintrag nur als neues "oberstes" Element in den Stack einfügen (mit einer *push*–Operation) bzw. das oberste Element entfernen (mit einer *pop*–Operation, also beim DFS wenn der Knoten schwarz wird). Als weitere Funktionalität bietet ein Stack die Abfrage nach seiner Größe (*size*–Operation), die Boolesche Anfrage *isEmpty* und die Ausgabe des obersten Eintrages (*top*–Operation, ohne den Eintrag zu entfernen).
Sie werden verschiedene Realisierungen einer Stack–Datenstruktur noch kennen lernen und in Java implementieren (ALP II+III).

Man kann die Kanten eines Graphen nach ihrer Rolle bei einer DFS–Durchmusterung klassifizieren. Wir unterscheiden:

- Tree-Kanten (T–Kanten): von grauen nach weißen Knoten
- Back–Kanten (B–Kanten): von grau nach grau
- Forward–Kanten (F–Kanten): von grau nach schwarz und zwar von Vorfahren zu Nachkommen im DFS–Baum
- Cross–Kanten (C–Kanten): alle restlichen Graphkanten

Im folgenden Beispiel sind die Zeitintervalle und die Kantenklassifikation illustriert:



DFS−Suche. Zeitintervalle der Knoten und Kantenklassifikation

Die Zeitintervalle haben eine sehr schöne Eigenschaft. Liegt Knoten v unter Knoten u, so ist $(d[v], f[v]) \subset (d[u], f[u])$. Also Knoten, die tiefer liegen, werden später entdeckt, sind aber eher fertig. Liegt v nicht unter u und u nicht unter v, so sind die Intervalle disjunkt.

6.5.2 Gerichtete azyklische Graphen

Gerichtete azyklische Graphen (dag's) sind gerichtete Graphen ohne gerichtete Kreise. Diese spielen in der Informatik an verschiedenster Stelle eine Rolle, zum Beispiel bei Vererbungshierarchien in Java. Oder man stelle sich eine Menge von Jobs vor, die linear zu ordnen sind. Dabei gibt es Einschränkungen derart, dass ein Job a vor einem anderen Job b bearbeitet werden muß, repräsentiert durch eine gerichtete Kante von a nach b. Gesucht ist eine lineare Ordnung (eine sogenannte *topologische Sortierung*), die alle diese Constraints berücksichtigt.
Definition: Eine topologische Sortierung eines gerichteten Graphen ist eine Nummerierung seiner Knoten derart, dass aus $(u, v) \in E$ folgt $u \leq v$ in der Nummerierung.

Offensichtlich muss der gerichtete Graph ein dag sein, um eine topologische Sortierung zuzulassen. Das es dann aber immer geht, überlegt man sich wie folgt, vgl. Abschnitt über Halbordnungen.

Satz: Jeder dag hat eine Quelle (Knoten mit Ingrad 0) und eine Senke (Ausgrad 0).

Beweis: Wir wählen einen beliebigen Knoten v_0 und testen, ob er Quelle ist. Falls nicht, betrachten wir Kante (v_1, v_0) und testen, ob v_1 Quelle ist usw. Angenommen, wir haben einen gerichteten Weg $v_i \to \ldots \to v_0$ von Nichtquellen schon gefunden. Dann gilt $v_{i+1} \notin \{v_0, \ldots, v_i\}$, denn sonst gäbe es einen gerichteten Kreis. Dieser Prozess muss abbrechen mit einer gefundenen Quelle, denn der Graph ist endlich.
Eine Senke findet man analog, indem man ausgehende Kanten verfolgt. □

Diesen Satz kann man auch algorithmisch umsetzen: Suche eine Senke, diese bekommt die größte Nummer in der topologischen Sortierung. Entferne die Senke und die einge-

henden Kanten. Dies liefert einen dag und man iteriert.
Das ist zwar einfach, aber auch nicht besonders effizient, denn v_0 könnte schlecht gewählt sein. Es geht besser mit DFS:

Fakt: Ein gerichteter Graph ist ein dag genau dann, wenn ein DFS–Durchmustern keine Back–Kanten produziert.

Beweis:
($\Rightarrow$): Angenommen, es gäbe eine Back–Kante (u,v). Das heißt, v ist Vorfahre von u im dfs–Wald. Damit gibt es einen gerichteten Weg von v nach u und (u,v) schließt einen gerichteten Kreis.
($\Leftarrow$): Angenommen, G enthält einen gerichteten Kreis c. Sei v der erste Knoten in c, der beim dfs entdeckt wird und sei (u,v) Kante in c. Es gibt also zum Zeitpunkt $d[v]$ einen gerichteten Weg vom grauen Knoten v zum weißen Knoten u, der nur weiße Knoten benutzt.
u ist Nachkomme von v im dfs–Wald, also $f[u] < f[v]$ und (u,v) ist Back–Kante. □

Basierend auf dieser Einsicht kann man topologisches Sortieren wie folgt realisieren: Führe ein DFS für den dag durch; wenn immer ein Knoten schwarz wird, gib ihn aus.

Behauptung: Dies ergibt eine topologisch invers sortierte Knotenfolge.

Beweis: Zu zeigen, wenn $(u,v) \in E$, dann ist $f(v) \leq f(u)$. Wir betrachten den Moment, wenn die Kante (u,v) vom DFS untersucht wird. Zu diesem Zeitpunkt ist u grau. v kann nicht grau sein wegen obigen Fakts. Also bestehen nur die beiden Möglichkeiten, v ist weiß oder schwarz. Aber in beiden Fällen ist offensichtlich $f(v)$ kleiner als $f(u)$, denn weiße Knoten, die unter u liegen, sind "eher" fertig als u und schwarze Knoten sind ja schon völlig abgearbeitet. □

6.5.3 Einfache Anwendungen von Breiten– und Tiefsuche

Breiten– und Tiefensuche treten als Subroutine in sehr vielen Graphalgorithmen auf. Einige algorithmische Probleme lassen sich aber unmittelbar damit lösen. Dazu zählen:

1. Testen, ob ein ungerichteter Graph zusammenhängend ist.
2. Anzahl der Zusammenhangskomponenten zählen und einen aufspannenden Baum (Wald) berechnen.
3. Testen, ob eine Zusammenhangskomponente genau einen Kreis enthält.
4. Testen, ob ein Graph bipartit ist und falls ja, eine gültige 2–Färbung der Knoten berechnen.
5. Testen, ob ein Graph eine Eulertour besitzt und falls ja, eine solche bestimmen.
6. **Nur mit BFS** kann man graphentheoretische Abstände und daraus abgeleitete Größen wie den Durchmesser bestimmen.

7. **Mit DFS** kann man effizient testen, ob ein gerichteter Graph ein DAG ist und falls ja eine topologische Sortierung berechnen.

6.6 Das Minimum–Spanning–Tree Problem: MST *

[Hinweis: Ergänzendes Material, nicht in Vorlesung präsentiert]
Sei $G = (V,E)$ ein ungerichteter zusammenhängender Graph und w eine Gewichtsfunktion, die jeder Kante eine positive reelle Zahl zuordnet, dies könnte zum Beispiel deren Länge sein.
Wir wissen, G hat einen aufspannenden Baum. Sei dies $T = (V,E')$. Wir definieren das Gesamtgewicht von T als

$$w(T) = \sum_{e \in E'} w(e)$$

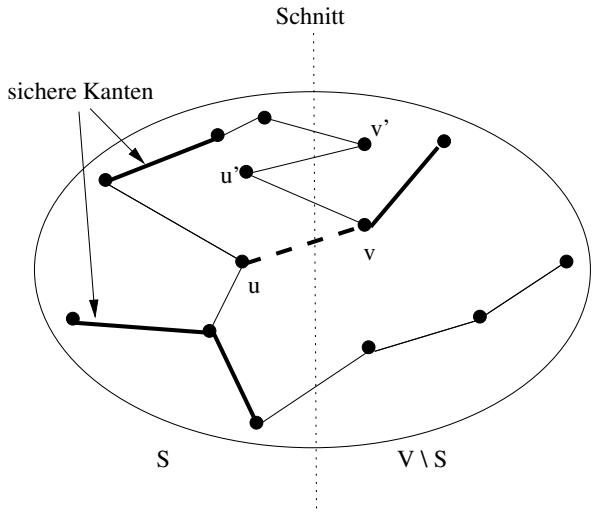
Aufgabe: Finde einen aufspannenden Baum mit minimalem Gesamtgewicht!

Fakt: Offensichtlich hat jeder kantengewichtete zusammenhängende Graph einen minimal aufspannenden Baum (MST) und im Allgemeinen muss der auch nicht eindeutig sein. □
Wir werden zuerst einen generischen MST–Algorithmus kennenlernen und danach zwei konkrete Umsetzungen.

Definition: Ein *Schnitt* von G ist eine Zerlegung $(S, V \setminus S)$ seiner Knotenmenge.
Eine Kantenmenge A *respektiert* einen Schnitt $(S, V \setminus S)$, falls keine Kante aus A einen Knoten aus S mit einem aus $V \setminus S$ verbindet.
Eine Kantenmenge A heißt *sicher*, wenn es einen MST T von G gibt, so dass A in der Kantenmenge von T enthalten ist.

Satz: (Generischer MST–Algorithmus)
Sei G ein gewichteter ungerichteter zusammenhängender Graph und sei A eine sichere Menge von Kanten, die einen Schnitt $(S, V \setminus S)$ respektiert. Wir betrachten eine leichteste Kante uv, mit $u \in S$, $v \in V \setminus S$. Dann ist $A \cup \{uv\}$ sicher.

Beweis: Sei T ein MST, der A enthält. Wir nehmen an, dass uv nicht zu T gehört, ansonsten ist nichts zu beweisen.
Wenn wir die Kante uv zu T hinzunehmen, entsteht genau ein Kreis C. Wir betrachten in C alle Kanten xy mit $x \in S, y \in V \setminus S$. Außer uv muss es wenigstens noch eine weitere solche Kante geben. Sei dies $u'v'$.
Streichen wir $u'v'$ aus $T \cup \{uv\}$, so entsteht ein aufspannender Baum T'. Da nach Annahme $w(uv) \leq w(u'v')$ und T ein MST ist, folgt sofort, T' ist MST und mithin ist $A \cup \{uv\}$ sicher. □



Der MST T und die Kante uv definieren Kreis!

Der Satz kann algorithmisch umgesetzt werden. Man startet mit A leer, sucht sich einen Schnitt, der von A respektiert wird (dies sind am Anfang alle), nimmt die leichteste Kante über den Schnitt hinzu usw.

Die beiden MST–Algorithmen von Prim und Kruskal sind konkrete Umsetzungen davon.

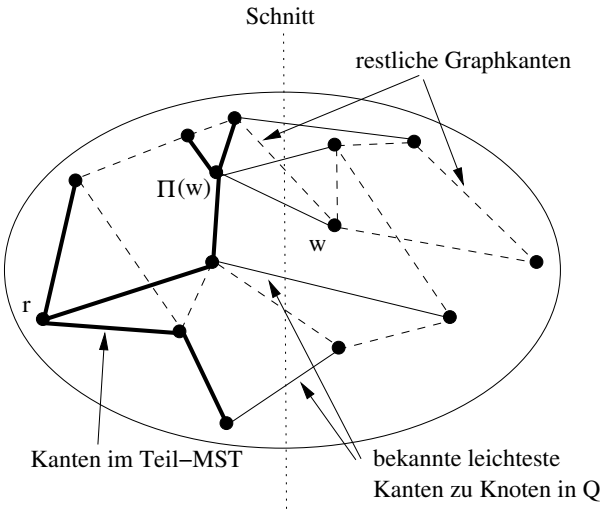
6.6.1 Der MST–Algorithmus von Prim

Dieser ist konzeptionell sehr einfach. Wir lassen den MST–Baum von einem beliebigen Startknoten r aus 'wachsen'. Die Frage ist, um welche Kante die Teillösung in einem Schritt erweitert wird. Wir schauen uns den Schnitt an, der die Teilösung vom Rest trennt. In einer Prioritätsschlange verwalten wir alle noch nicht erreichten Knoten w zusammen mit einem Schlüssel, der angibt, was im Moment das Gewicht einer leichtesten Kante ist, mittels derer w von einem der bereits in den Teilbaum aufgenommenen Knoten aus erreichbar ist.
Diese Schlüssel benötigen ggf. entsprechende Updates. Die π–Zeiger speichern wieder den aktuell gefundenen Teil–MST mit Wurzel r sowie die bekannten kürzesten Kanten zu Knoten in Q.

MST-Prim(G, w, r)

```
01 Q ← V(G)
02 for jedes u ∈ Q
03       key[u] ← ∞
04 key[r] ← 0
05 π[r] ← NIL
06 while Q ≠ ∅
07       do u ←Extract-Min(Q)
08          for jedes v ∈ Adj[u]
09             do if v ∈ Q und w(u,v) < key[v]
10                   then π[v] ← u
11                        key[v] ← w(u,v)
```

Die Komplexität des Algorithmus hängt ab von der konkreten Realisierung des abstrakten Datentyps Prioritätsschlange ab, wir werden einige kennen lernen. Benutzt man für die Implementierung der Prioritätswarteschlange einen binären Heap (s. ALP III) , so ist die Komplexität $O(|V|\log|V| + |E|\log|V|) = O(|E|\log|V|)$.



6.6.2 Der MST–Algorithmus von Kruskal

Zuerst werden die Kanten nach aufsteigenden Gewichten sortiert. Danach wird in dieser Reihenfolge, mit der leichtesten Kante beginnend, getestet, ob eine Kante einen Kreis im bisher konstruierten Wald schließt (falls ja wird die Kante verworfen). Falls nein wird die sichere Menge um diese Kante erweitert.

Das Interessanteste dabei ist die verwendete Datenstruktur, eine sogenannte Union-Find–Struktur zur Verwaltung von Partitionen einer Menge, hier der Knotenmenge V.

Die von der Datenstruktur unterstützten Operationen sind:

- makeSet(v): Aus einem Element $v \in V$ wird eine Menge gemacht.
- union(u,v): Die beiden Mengen, zu denen u bzw. v gehören, werden vereinigt.
- findSet(v): Bestimme die Menge, zu der v gehört.

Dazu stelle man sich vor, dass jede Menge einen Repräsentanten hat, auf den alle Elemente der Menge zeigen. Bei der union–Operation müssen also Zeiger umgehangen werden. Eine einfache Realisierung besteht darin, bei union zweier Mengen, alle Zeiger der kleineren Menge auf den Repräsentanten der größeren umzuleiten.
Man überlegt sich wie folgt, dass damit insgesamt nur höchstens $|V|\log|V|$ mal Zeiger umgeleitet werden. Betrachten wir einen konkreten Knoten $v \in V$. Nach dem ersten Umhängen seines Zeigers landet er in einer Menge, die mindestens doppelt so groß ist, also mindestens zwei Knoten hat. Wenn er das nächste Mal angefasst wird, landet er in einer Menge mit ≥ 4 Knoten usw. Insgesamt kann er aber nur $\log_2|V|$ mal umgelenkt werden, danach ist ganz V erreicht. Das gilt für jeden Knoten, also insgesamt höchstens

$|V|\log|V|$ Zeiger werden während des gesamten Algorithmus umgehangen. (Dies ist ein Beispiel einer sogenannten amortisierten Analyse.)

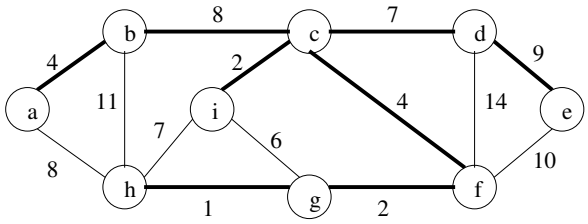
Hier ist der Pseudocode für Kruskals Algorithmus.

MST-Kruskal(G,w)

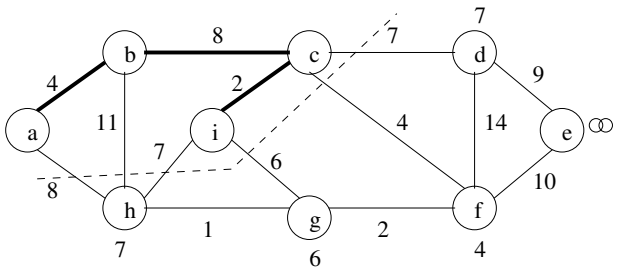
```
01 A ← ∅
02 for jeden Knoten v ∈ V(G)
03     do makeSet(v)
04 Sortiere die Kanten aus E in nichtfallender Reihenfolge
   entsprechend ihres Gewichts
05 for jede Kante (u,v) in sortierter Reihenfolge
06     do if find-Set(u) ≠ find-Set(v)
07           then A ← A∪ {(u,v)}
08                union(u,v)
09 return A
```

Mit der oben erwähnten Union–Find–Implementierung ist die Komplexität des Algorithmus $O(|E|\log|E|+|V|\log|V|)$.

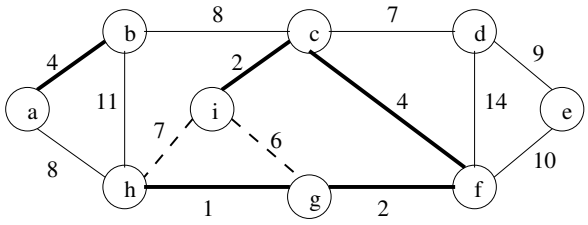
Beispiel:



Die dicken Kanten bilden einen MST mit Gewicht 37
(die Kante bc könnte auch durch ah ersetzt werden)



Prim−Algorithmus: Zustand , nachdem a,b,c,i in dieser
Reihenfolge aus Prioritätswarteschlange entfernt wurden.
An den verbleibenden Knoten ist der aktuelle Schlüssel
vermerkt.



Kruskal−Algorithmus: Nach der Aufnahme der ersten
5 Kanten werden in den nächsten beiden Schritten die
Kanten ig und ih verworfen, da sie jeweils Kreise schließen.

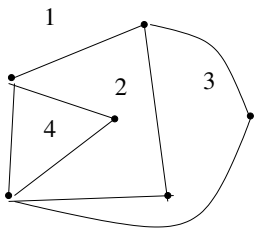
6.7 Die Euler–Formel für planare Graphen; Maximales Matching

6.7.1 Die Euler–Formel

Definition: Ein ungerichteter Graph heißt *planar*, falls er sich so in der Ebene zeichnen lässt, dass sich Kanten nur in gemeinsamen Endknoten berühren.
Eine Zeichnung ist dabei die Zuordnung von Punkten für die Knoten des Graphen und die Kanten werden dargestellt durch stetige Kurven (ohne Selbstschnitte), die die entsprechenden Punkte verbinden.

Tatsächlich kann man planare Graphen immer so zeichnen, dass die Kanten durch Strecken dargestellt werden, aber das ist für das Folgende nicht wichtig. Jede Zeichnung zerlegt die Ebene in Regionen, die durch Kanten begrenzt sind. Eine Region ist dabei eine maximale Menge von Punkten, so dass jeder Punkt von jedem anderen erreichbar ist, ohne eine Kante zu kreuzen. Im nachfolgenden Beispiel sind dies 4 Regionen.

Sei v die Anzahl der Knoten, e die Anzahl der Kanten und f die Anzahl der Regionen



(“Facetten”) in der Zeichnung.

Satz: (Euler-Formel)
Für jede Zeichnung eines zusammenhängenden planaren Graphen $G = (V, E)$ gilt:

$$v - e + f = 2$$

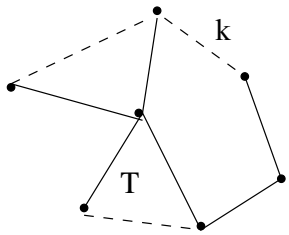
Beweis: Vollständige Induktion über e.
Induktionsanfang: $e = 0$. Dann besteht der Graph nur aus einem Knoten und die Zeichnung hat eine Facette.

$$1 - 0 + 1 = 2$$

Induktionsschritt: Wir nehmen an, die Aussage ist richtig für Graphen mit e Kanten.
Habe G jetzt $e + 1$ Kanten.
Fall 1: G ist ein Baum. Dann hat G $e + 2$ Knoten und jede Zeichnung hat nur eine Facette.
Also ist:

$$(e + 2) - (e + 1) + 1 = 2$$

Fall 2: G hat einen Kreis C. Sei T ein aufspannender Baum. Der Kreis C hat eine Kante k, die nicht zu T gehört. (Die Nichtbaumkanten sind in der Abb. gestrichelt.)



Entfernen wir diese Kante aus der Zeichnung von G, so entsteht die Zeichnung eines zusammenhängenden planaren Graphen G' mit gleichvielen v' Knoten wie G, mit e Kanten und $f' = f - 1$, wobei f die Facettenzahl von G ist. Für G' gilt nach Induktionsannahme:

$$2 = v' - e + f - 1 = v - (e + 1) + f$$

Das ist aber die Behauptung für G. □

Im Beweis haben wir folgende zwei Fakten ohne Beweis benutzt:

1. Jede Zeichnung eines Baumes hat genau eine Facette.
2. (Jordanscher Kurvensatz) Jede doppelpunktfreie geschlossene Kurve in der Ebene teilt die Ebene in zwei Regionen.

Korollar: Ein planarer einfacher Graph kann höchstens $3v - 6$ Kanten enthalten.

Beweis: Die Maximalanzahl von Kanten wird offensichtlich dann erreicht, wenn alle Regionen durch genau 3 Kanten begrenzt sind.
Dann ist mit doppeltem Abzählen $2e = 3f$, also $f = 2e/3$. Setzt man dies in die Eulerformel ein, so ergibt sich die Behauptung. □

Planare Graphen sind also "dünn" (*sparse*), das heißt, sie enthalten nur linear viele Kanten in der Anzahl der Knoten, verglichen mit allgemeinen Graphen die quadratisch viele Kanten enthalten können. Dies ist wichtig für Komplexitätsanalysen von Graphalgorithmen auf planaren Graphen!

6.7.2 Reguläre Polyeder

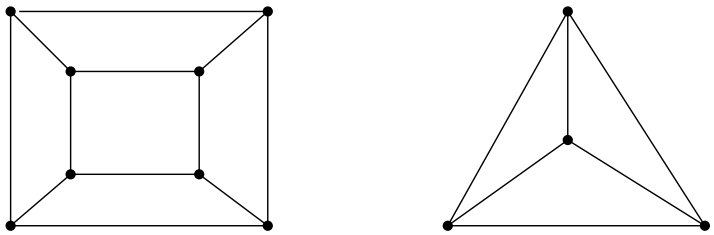
Wir benutzen die Eulerformel um zu klassifizieren, welche regulären Polyeder (*Platonische Körper*) es geben kann.

Definition
Ein Polyeder ist eine konvexe 3–dimensionale Region, die durch eine endliche Anzahl polygonaler Flächen begrenzt wird.
Ein reguläres Polyeder ist eines, bei dem die Randflächen alle identische reguläre n–Ecke sind und an jedem Knoten genau m viele solche Flächen zusammenstoßen.

Beispiel ist der Würfel, bei dem an jeder Ecke drei Quadrate sich treffen. Die Frage ist, für welche n und m gibt es solche regulären Polyeder.

Beobachtung Die Kanten und Ecken eines Polyeders definieren einen planaren Graphen.

Dazu stelle man sich etwa vor, dass die Oberfläche des Polyeders aus Gummi ist, man in eine Fläche ein Loch schneidet und dieses dann immer größer macht. Hier sind die Graphen des Würfels und des Tetraeders.



Aus dem Handschlag–Lemma folgt für die Graphen regulärer Polyeder, dass gilt $m \cdot v = 2e$. Ebenso ist $n \cdot f = 2e$. Dies eingesetzt in die Eulerformel ergibt:

$$\frac{1}{n} + \frac{1}{m} = \frac{1}{2} + \frac{1}{e}$$

Wir wissen außerdem, dass $n,m \geq 3$ gelten muss. Denn eine Ecke eines Polyeders wird von mindestens drei Seiten definiert und eine Seitenfläche hat mindestens 3 Ecken.
Andererseits kann n und m nicht größer als 5 sein, denn sonst kann die Gleichung nicht erfüllt werden. Damit ergeben sich genau 5 Fälle, für die es dann tatsächlich auch reguläre Polyeder gibt. Diese sind:

n	m	v	e	f	reg. Polyeder
3	3	4	6	4	Tetraeder
4	3	8	12	6	Würfel
3	4	6	12	8	Oktaeder
3	5	12	30	20	Ikosaeder
5	3	20	30	12	Dodekaeder

6.7.3 Der Heiratssatz: Maximales Matching in bipartiten Graphen

Sei $G = (V,E)$ ein ungerichteter Graph.

Definition: Ein *Matching* in G ist eine Menge von Kanten, von denen keine zwei einen Knoten gemeinsam haben.

Aufgabe ist es ein Matching mit maximaler Kardinalität zu finden. Insbesondere im Fall eines bipartiten Graphen lassen sich viele reale Situationen so modellieren. Man denke an eine Menge von Maschinen A und eine Menge von Jobs B. Jede einzelne Maschine a kann bestimmte Jobs $N(a) \subseteq B$ ausführen, aber nur einen pro Zeiteinheit. Man möchte die Jobs so auf die Maschinen verteilen, dass möglichst viele Jobs gleichzeitig bearbeitet werden. Wir repräsentieren den Fakt, dass a einen Job b ausführen kann, durch eine Kante zwischen a und b und dann sind $N(a)$ alle Nachbarn von a und wir suchen ein maximales Matching. Der folgende Satz gibt ein notwendiges und hinreichendes Kriterium (das *Heiratskritriterium*) dafür an, dass gleichzeitig alle $a \in A$ mit entsprechenden Elementen aus B *verheiratet* werden können.
Dieses sagt, dass die Nachbarschaft jeder Teilmenge von A mindestens so groß sein muss wie die Menge selbst.

Satz: (M. Hall)
Sei $G = (V,E)$ mit $V = A \cup B$, $A \cap B = \emptyset$ ein bipartiter Graph.
G hat ein Matching M der Größe $|A| \Leftrightarrow \forall A' \subseteq A: \ |N(A')| \geq |A'|$ mit $N(A') = \bigcup_{a \in A'} N(a)$.

Beweis:
($\Rightarrow$): Das ist klar, denn $\{b \in B | \exists a \in A' : \{a,b\} \in M\} \subseteq N(A')$.
($\Leftarrow$): (vollständige Induktion nach $|A|$)
Induktionsanfang: $|A| = 1$
Wir matchen das einzige a mit einem Element aus der Nachbarschaft $N(A)$, fertig.
Induktionsschritt: Sei $|A| \geq 2$.
Wir betrachten 2 Fälle.
(1) $\forall A' \subset A: \ |N(A')| > |A'|$
Wir wählen ein a und fixieren eine beliebige zu a inzidente Kante fürs Matching. Wir

entfernen die beiden Knoten und die Kante aus dem Graphen und beobachten, dass die Heiratsbedingung für $A \setminus \{a\}$ im neuen Graphen gilt und wir somit dafür die Induktionsvoraussetzung anwenden können.
(2) $\exists A^* \subset A : \ |N(A^*)| = |A^*|$
Man beachte, dass A^* echte Teilmenge von A ist. Wir wählen nach Induktionsvoraussetzung für diese Menge A^* das Matching mit $N(A^*)$ und entfernen die beteiligten Knoten und Kanten aus dem Graph. Der Restgraph ist immer noch bipartit und erfüllt die Heiratsbedingung. Denn für $A' \subseteq A \setminus A^*$ gilt $|N(A')| \geq |A'|$. Dies folgt aus der Tatsache, dass in G galt $|N(A' \cup A^*)| \geq |A' \cup A^*|$ wegen der Heiratsbedingung und von den Mengen auf beiden Seiten wurden $|N(A^*)| = |A^*|$ Elemente entfernt. □

Wer wissen will, wie man ein maximales Matching in einem bipartiten Graphen algorithmisch findet, lese etwas über *Flüsse in Netzwerken*, z.B. hier:
`http://www.inf.fu-berlin.de/lehre/WS06/mafi1/material/fluss.pdf`

7 Resolutionskalkül und Prädikatenlogik

Die Grundlagen der Aussagenlogik wurden bereits Kapitel 1 besprochen. Bisher dienten Aussageverknüpfungen nur als Mittel, um bestimmte Sachverhalte in eine formale Sprache zu übertragen und um die Struktur von mathematischen Beweisen besser zu verstehen. Die Hauptmotivation zur Beschäftigung mit diesem Gebiet greift aber wesentlich weiter: Zum einen geht es um das formale (und möglichst automatische) Ableiten von neuen Aussagen aus einer gegebenen Menge von Aussagen (Voraussetzungen), zum anderen um Verfahren zur Prüfung, ob eine Aussage allgemeingültig (Tautologie) oder erfüllbar ist. Wir werden sehen, dass beide Problemstellungen eng zusammenhängen. Das führt uns auf die Suche nach Methoden zum automatisierten Beweisen vom mathematischen Theoremen und letztlich zur Frage, ob, in welchem Sinne und wie die Tätigkeit von Mathematikern durch Computer ersetzt werden kann.

7.1 Tautologien, Modelle und aussagenlogisches Folgern

Die folgenden Definitionen erweitern den Begriff der Erfüllbarkeit von Formeln auf Formelmengen und zeigen verschiedene Regeln auf, mit denen man erfüllbare Formelmengen erweitern kann.

Definition: Sei $F \in \mathcal{F}_n$ eine Formel und $X \subseteq \mathcal{F}$ eine Menge von Formeln ($\mathcal{F} = \bigcup_{i=1}^{\infty} \mathcal{F}_n$). F (bzw. X) wird erfüllbar genannt, wenn es eine Belegung $\omega \in \Omega_n$ (bzw. $\omega : Var \longrightarrow \{0,1\}$) gibt, so dass $\Phi_n(\omega, F) = 1$ (bzw. $\Phi(\omega, G) = 1$ für alle $G \in X$) gilt. Man sagt dann, ω *erfüllt* F (bzw. ω *ist ein Modell für* X) und schreibt $\omega \models \alpha$ (bzw. $\omega \models X$).

Die Formel F wird *allgemeingültig, logisch gültig* oder eine *Tautologie* genannt, wenn $\omega \models \alpha$ für alle ω. Wir schreiben $\models F$, falls F eine Tautologie ist, und $\not\models F$, falls F keine Tautologie ist.
Eine Formel, die unerfüllbar ist, wird *Kontradiktion* genannt.

Satz: Eine Formel F ist eine Tautologie genau dann, wenn $\neg F$ eine Kontradiktion ist.

Beispiele: Die Formeln $F \vee \neg F$ und $F \Leftrightarrow F$ sind Tautologien (kurz: $\models F \vee \neg F$ und $\models F \Leftrightarrow F$). Allgemein sind zwei Formeln F und G genau dann äquivalent, wenn $F \Leftrightarrow \beta$ eine Tautologie ist.
Die Formel $F \wedge \neg F$ ist eine Kontradiktion.

Für beliebige Formeln F, G, H kann man die folgenden Tautologien bilden:

$F \Rightarrow F$	(*Selbstimplikation*)
$F \Rightarrow (G \Rightarrow F)$	(*Prämissenbelastung*)
$(F \Rightarrow (G \Rightarrow H)) \Rightarrow (G \Rightarrow (F \Rightarrow H))$	(*Prämissenvertauschung*)
$(F \Rightarrow G) \Rightarrow ((G \Rightarrow H) \Rightarrow (F \Rightarrow H))$	(*gewöhnlicher Kettenschluss*)
$(F \Rightarrow (G \Rightarrow H)) \Rightarrow ((F \Rightarrow G) \Rightarrow (F \Rightarrow H))$	(*Fregescher Kettenschluss*)

Definition: Die Formel F wird eine *aussagenlogische Folgerung* aus der Formelmenge X genannt, falls für alle Modelle ω von X auch $\omega \models \alpha$ gilt. Man schreibt dafür $X \models F$. Ist $X = \{F_1, \ldots, F_n\}$, dann kann für $X \models F$ auch $F_1, \ldots, F_n \models F$ geschrieben werden.

Drei wichtige Folgerungsbeziehungen, die die Grundlage für zahlreiche Beweise in der Mathematik bilden, sollen hier besonders hervorgehoben werden,
der *Modus Ponens:*

$$F, F \Rightarrow G \models G,$$

der *Beweis durch Fallunterscheidung:*

$X, F \models G$ und $X, \neg F \models G$ impliziert $X \models G$

und das *Deduktionstheorem:*

$X, F \models G$ impliziert $X \models F \Rightarrow G$

Man kann beweisen, dass die Implikation im Deduktionstheorem sogar eine Äquivalenz ist.

Die Frage nach Algorithmen, die entscheiden, ob eine gegebene Formelmenge erfüllbar ist, gehört zu den grundlegenden Aufgabenstellungen in der künstlichen Intelligenz. Aber bereits das Erfüllbarkeitsproblem für einzelne Formeln ist *NP–vollständig*. Obwohl der Begriff der NP–Vollständigkeit hier nicht genauer erläutert werden kann, sollen einige Konsequenzen aus diesem Fakt genannt werden:

1) Ist eine gegebene Formel F erfüllbar, dann gibt es dafür einen kurzen Beweis (man rechnet für eine "geeignete" Belegung ω nach, dass $\omega F = 1$ ist).

2) Es ist kein in polynomieller Zeit laufender Algorithmus bekannt, der die Erfüllbarkeit von Formeln entscheidet (natürlich kann man ωF für alle Belegungen ω berechnen, aber es gibt exponentiell viele Belegungen).

3) Würde man einen Polynomialzeit-Algorithmus finden, der die Erfüllbarkeit von Formeln entscheidet, so könnte man daraus Polynomialzeit-Algorithmen für alle anderen NP–vollständigen Probleme ableiten (z.B. Hamiltonkreis und TSP in Graphen und zahlreiche schwierige Optimierungsprobleme)

Auch die Probleme, zu entscheiden, ob eine gegebene Formel eine Tautologie bzw. eine Kontradiktion ist, weisen die gleichen Schwierigkeiten auf. Die Möglichkeit diese Probleme aufeinander zurückzuführen basiert auf der Tatsache, dass für jede Formel F die folgenden drei Aussagen äquivalent sind:

F ist erfüllbar

F ist keine Kontradiktion

$\neg F$ ist keine Tautologie

Satz (Endlichkeitssatz, compactness theorem):

Eine Menge X von Formeln ist genau dann erfüllbar, wenn jede der endlichen Teilmengen von X erfüllbar ist.

Wir verzichten hier auf den Beweis des Satzes und merken an, dass seine typische Anwendung in Kontraposition erfolgt: Ist eine unendliche Formelmenge $\{F_1, F_2, \ldots\}$ nicht erfüllbar, dann gibt es ein n, so dass bereits $\{F_1, F_2, \ldots, F_n\}$ nicht erfüllbar ist.

7.2 Resolutionskalkül

Ein *Kalkül* ist eine Kollektion von syntaktischen Umformungsregeln, die unter gegebenen Voraussetzungen aus bereits vorhandenen Formeln neue Formeln erzeugen. Der *Resolutionskalkül* besteht aus einer einzigen Umformungsregel – der sogenannten *Resolution*. Das gesamte Verfahren dient dazu, die Unerfüllbarkeit einer Formelmenge zu testen und gegebenenfalls nachzuweisen. Das Verfahren ist einfach, aber auf Grund der NP–Vollständigkeit des Erfüllbarkeitsproblems muss man damit rechnen, dass auch dieses Verfahren für einige Eingaben exponentielle Laufzeit erfordert.

Sei eine endliche Formelmenge $X = \{F_1, \ldots, F_n\}$ gegeben. Wir setzen voraus, daß alle Formeln bereits in KNF vorliegen. Da die Formelmenge X genau dann erfüllbar (unerfüllbar) ist, wenn die Formel $F = F_1 \wedge \ldots \wedge F_n$ erfüllbar (unerfüllbar) ist, reicht es aus, die Unerfüllbarkeit einer KNF–Formel

$$F = (l_{1,1} \vee l_{1,2} \vee \ldots \vee l_{1,n_1}) \wedge \ldots\ldots \wedge (l_{k,1} \vee l_{k,2} \vee \ldots \vee l_{k,n_k})$$

zu testen, wobei alle $l_{i,j} \in \{x_1, x_2, \ldots\} \cup \{\neg x_1, \neg x_2, \ldots\}$ Literale sind. Zur Vereinfachung wird F als eine Menge $\mathcal{K}_F$ von Klauseln geschrieben, welche die einzelnen Disjunktionsglieder repräsentieren:

$$\mathcal{K}_F = \{\{l_{1,1}, l_{1,2}, \ldots, l_{1,n_1}\}, \ldots\ldots, \{l_{k,1}, l_{k,2}, \ldots, l_{k,n_k}\}\}$$

Für jedes Literal l definieren wir $\bar{l} = \begin{cases} \neg p_i & falls\ l = p_i \\ p_i & falls\ l = \neg p_i \end{cases}$

Definition: Seien K_1, K_2 Klauseln und l ein Literal mit $l \in K_1$ und $\bar{l} \in K_2$. Dann wird die Klausel $R = (K_1 \setminus \{l\}) \cup (K_2 \setminus \{\bar{l}\})$ ein *Resolvent* von K_1 und K_2 genannt.

Zur Darstellung nutzt man die folgende Diagrammschreibweise:

Die leere Klausel wird explizit als Resolvent zugelassen. Sie wird durch das Symbol $\square$ bezeichnet. Die leere Klausel gilt als nicht erfüllbar, also als eine Kontradiktion.

Resolutions–Lemma: Sei F eine Formel in KNF, dargestellt als Klauselmenge $\mathcal{K}_F$ und sei R ein Resolvent zweier Klauseln aus $\mathcal{K}_F$. Dann sind F und die durch $\mathcal{K}_F \cup \{R\}$ dargestellte Formel F' logisch äquivalent.

Beweis: Eine Richtung in dieser Äquivalenz ist einfach zu zeigen:
Wenn $\omega : Var \longrightarrow \mathbb{B}$ eine erfüllende Belegung für F' ist, dann nimmt jede Klausel aus F' unter ω den Wert 1 an. Damit ist ω aber auch erfüllende Belegung für F.

Für die Gegenrichtung ist eine etwas genauere Analyse notwendig. Wir nehmen an, dass $\omega : Var \longrightarrow \mathbb{B}$ eine erfüllende Belegung für F ist und wollen zeigen, dass dann auch

alle Klauseln von F' unter ω den Wert 1 annehmen. Bis auf den Resolventen R folgt das aus der Voraussetzung. Um es auch für R zu zeigen, betrachten wir seine Entstehung

$$R = (K \setminus \{l\}) \cup (K' \setminus \{\bar{l}\}), \text{ wobei } l \in K \text{ und } \bar{l} \in K'$$

und machen eine Fallunterscheidung danach, welchen Wert das Literal l unter der Belegung ω hat:

- $\omega(l) = 0$: Die Klausel K kann nicht durch das Literal l den Wert 1 bekommen, also muss ein anderes Literal l' in K auftreten, das unter ω den Wert 1 hat. Dann ist $l' \in R$ und folglich nimmt R unter ω den Wert 1 an.
- $\omega(l) = 1$, d.h. $\omega(\bar{l}) = 0$: Die Klausel K' kann nicht durch das Literal $\bar{l}$ den Wert 1 bekommen, also muss ein anderes Literal l' in K' auftreten, das unter ω den Wert 1 hat. Dann ist $l' \in R$ und folglich nimmt R unter ω den Wert 1 an. $\square$

Definition: Für eine beliebige Klauselmenge $\mathcal{K}$ definiert man:

$$\begin{array}{lcl} Res(\mathcal{K}) & = & \mathcal{K} \cup \{R \mid R \textit{ ist Resolvent zweier Klauseln aus } \mathcal{K}\} \\ Res^0(\mathcal{K}) & = & \mathcal{K} \\ Res^{n+1}(\mathcal{K}) & = & Res(Res^n(\mathcal{K})) \\ Res^*(\mathcal{K}) & = & \bigcup_{n=1}^{\infty} Res^n(\mathcal{K}) \end{array}$$

Beispiel: Sei $\mathcal{K} = Res^0(\mathcal{K}) = \{\{x_1, x_2, \neg x_3\}, \{\neg x_1, x_4\}, \{x_2, \neg x_4\}\}$.
Dann ist
$Res^1(\mathcal{K}) = \mathcal{K} \cup \{\{x_2, \neg x_3, x_4\}, \{\neg x_1, x_2\}\}$,
$Res^2(\mathcal{K}) = Res^1(\mathcal{K}) \cup \{\{x_2, \neg x_3\}\}$ und
$Res^*(\mathcal{K}) = Res^2(\mathcal{K}) =$
$= \{\{x_1, x_2, \neg x_3\}, \{\neg x_1, x_4\}, \{x_2, \neg x_4\}, \{x_2, \neg x_3, x_4\}, \{\neg x_1, x_2\}, \{x_2, \neg x_3\}\}$.

Man beachte, dass die durch die Klauselmenge $\mathcal{K}$ dargestellte Formel
$F = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_4) \wedge (x_2 \vee \neg x_4)$ erfüllbar ist, denn jede Belegung ω mit $\omega(x_2) = \omega(x_4) = 1$ ist ein Modell für F.

Da eine endliche Klauselmenge $\mathcal{K}$ nur endlich viele Literale enthält, ist auch die Menge der ableitbaren Resolventen endlich (eine Untermenge der Potenzmenge aller vorkommenden Literale). Folglich kann auch $Res^*(\mathcal{K})$ in endlich vielen Schritten erzeugt werden, denn gilt $Res^{n+1}(\mathcal{K}) = Res^n(\mathcal{K})$ für ein $n \in \mathbb{N}$, dann ist $Res^*(\mathcal{K}) = Res^n(\mathcal{K})$. So liefert der folgende Satz die Grundlage für ein endliches Verfahren, das die Nichterfüllbarkeit von Formeln (und Formelmengen) entscheidet.

Resolutionssatz: Eine Formel F in KNF, dargestellt durch die Klauselmenge $\mathcal{K}_F$ ist genau dann unerfüllbar, wenn $\square \in Res^*(\mathcal{K}_F)$.

Die Aussage, dass $\square \in Res^*(\mathcal{K}_F)$ die Unerfüllbarkeit von F impliziert, wird als *Korrektheit des Resolutionskalküls* bezeichnet. Sie lässt sich leicht aus der Beobachtung ableiten, dass $\square$ nur Resolvent von zwei Klauseln der Form $\{x_i\}$ und $\{\neg x_i\}$ sein kann. Da bereits $x_i \wedge \neg x_i$ nicht erfüllbar ist, folgt die Nichterfüllbarkeit von F aus dem Resolutionslemma (mehrfache Anwendung).

Die entgegengesetzte Implikation (aus der Unerfüllbarkeit von F folgt $\square \in Res^*(\mathcal{K}_F)$) wird *Vollständigkeit des Resolutionskalküls* genannt. Sie kann durch Induktion über die

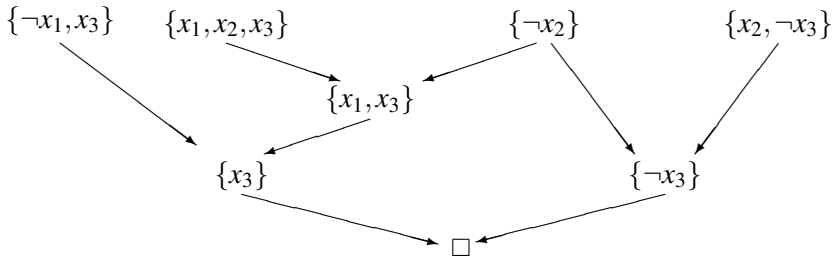
Anzahl der in F auftretenden Primformeln bewiesen werden. Wir verzichten an dieser Stelle auf den Beweis und verweisen auf das Buch von Schöning.

Der folgende Pseudocode beschreibt einen Algorithmus, der die Unerfüllbarkeit einer Formel F entscheidet, die durch eine Klauselmenge $\mathcal{K}$ gegeben ist:

repeat
  $\mathcal{J} := \mathcal{K}$;
  $\mathcal{K} := Res(\mathcal{J})$;
until ($\square \in \mathcal{K}$) **or** ($\mathcal{J} = \mathcal{K}$);
if $\square \in \mathcal{K}$ **then** "F ist unerfüllbar" **else** "F ist erfüllbar";

Generell sollte man beachten, dass zum Beweis der Unerfüllbarkeit einer Formel nicht unbedingt alle Resolventen gebildet werden müssen. Es reicht aus, nur die Resolventen zu bilden, die bei der *Deduktion* (Herleitung) von $\square$ eine Rolle spielen.

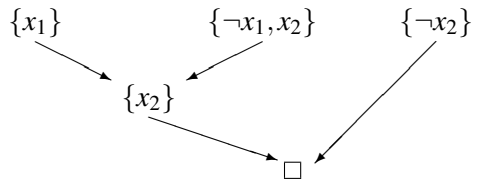
Beispiel: Sei $\mathcal{K} = \{\{x_1, x_2, x_3\}, \{\neg x_1, x_3\}, \{x_2, \neg x_3\}, \{\neg x_2\}\}$. Wir veranschaulichen die Deduktion der leeren Klausel durch einen sogenannten Resolutionsgraphen.



Auf Grund der bekannten Tatsache, dass F genau dann eine Tautologie ist, wenn $\neg F$ unerfüllbar ist, kann die Resolutionsmethode auch zum Tautologietest eingesetzt werden. Dazu muss aber die Negation $\neg F$ in KNF vorliegen. Gerade wenn F bereits als KNF–Formel gegeben ist, wird es oft sehr aufwändig sein, $\neg F$ in eine äquivalente KNF–Formel zu verwandeln, aber wenn F als DNF–Formel gegeben ist, kann man $\neg F$ durch doppelte Anwendung der deMorganschen Regel leicht in eine KNF verwandeln.

Eine weitere Anwendung für den Resolutionskalkül besteht im Beweis aussagenlogischer Folgerungen der Form $X \models F$. Sie basiert auf der Beobachtung, dass F genau dann aussagenlogische Folgerung aus einer Formelmenge $X = \{F_1, \ldots, F_n\}$ ist , wenn die Formel $F_1 \wedge \ldots \wedge F_n \wedge \neg F$ nicht erfüllbar ist. Vor der eigentlichen Anwendung des Resolutionskalküls muss man also wieder dafür sorgen, dass die Formeln $F_1, \ldots, F_n$ und die Negation $\neg F$ in KNF vorliegen.

Beispiel: Zum Beweis der Abtrennungsregel $\{x_1, x_1 \Rightarrow x_2\} \models x_2$ besteht der erste Schritt darin, $x_1 \Rightarrow x_2$ durch die äquivalente KNF $\neg x_1, \vee x_2$ zu ersetzen, und dann muss man die Deduktion von $\square$ aus $\{\{x_1\}, \{\neg x_1, x_2\}, \{\neg x_2\}\}$ finden:



7.3 Hornformel und Einheitsresolventen

Definition: Variablen werden *positive Literale* genannt, ihre Negationen nennt man *negative Literale*. Ein KNF–Term F wird *Hornformel* genannt, falls jeder Maxterm (Klausel) in F höchstens ein positives Literal enthält.

Beispiel: Die folgende Formel ist eine Hornformel:
$F = (x_1 \vee \neg x_3)(\neg x_1 \vee x_3 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_4) \wedge x_2 \wedge \neg x_4$

Eine besondere Eigenschaft der Klauseln einer Hornformel besteht darin, dass man sie als spezielle Implikationen ohne negierte Variable schreiben kann. Wir unterscheiden dazu drei Fälle:

- Der Maxterm enthält mindestens ein negatives Literal und genau ein positives Literal y :
  $G = \neg x_1 \vee \ldots \vee \neg x_k \vee y \equiv \neg(x_1 \wedge \ldots \wedge x_k) \vee y \equiv (x_1 \wedge \ldots \wedge x_k) \rightarrow y$
- Der Maxterm enthält nur negative Literale:
  $G = \neg x_1 \vee \ldots \vee \neg x_k \equiv \neg(x_1 \wedge \ldots \wedge x_k) \vee 0 \equiv (x_1 \wedge \ldots \wedge x_k) \rightarrow 0$
- Der Maxterm besteht nur aus einem positiven Literal:
  $G = y \equiv 0 \vee y \equiv \neg 1 \vee y \equiv 1 \rightarrow y$

Ein Nachteil der Hornformeln liegt in ihrer eingeschränkten Ausdruckskraft, d.h. es gibt Boolesche Funktionen, die man nicht durch Hornformeln darstellen kann (ein einfaches Beispiel ist die Disjunktion). Dafür kann man das Erfüllbarkeitsproblem für Hornformeln relativ leicht lösen. Grundlage für dieses Verfahren sind die folgenden drei Beobachtungen:

1. Wenn in jeder Klausel einer Hornformel F ein positives Literal auftritt, dann ist F erfüllbar (die erfüllende Belegung setzt alle Variable auf 1).
2. Wenn in jeder Klausel einer Hornformel F ein negatives Literal auftritt, dann ist F erfüllbar (die erfüllende Belegung setzt alle Variable auf 0).
3. Wenn eine Hornformel F eine Klausel enthält, die nur aus einem positiven Literal x_i besteht, dann muss x_i in jeder erfüllenden Belegung von F den Wert 1 bekommen.

Die algorithmische Idee besteht nun darin, schrittweise alle Variablen zu markieren, die in einer erfüllenden Belegung den Wert 1 annehmen müssen. Man markiert eine Variable x_i also nur dann, wenn $\{x_i\}$ eine Klausel der aktuellen Formel ist. Wird x_i irgendwann markiert, hat das zwei Konsequenzen (Beobachtung 3):
1) Alle Klauseln, die x_i enthalten, sind automatisch erfüllt und werden deshalb gestrichen.
2) Alle negativen Literale $\neg x_i$ können nicht mehr zur Erfüllung ihrer Klauseln beitragen, d.h. man streicht alle Vorkommen von $\neg x_i$ (aber nicht die Klauseln selbst!). Entsteht bei diesem Prozess eine leere Klausel, dann ist die Hornformel nicht erfüllbar (Abbruch mit Antwort unerfüllbar), denn wir hatten davor eine Klausel $\{x_i\}$ - als Anlass für die Markierung - und eine Klausel $\{\neg x_i\}$, aus der nach der Streichung die leere Klausel entstanden ist.
Es gibt zwei weitere Abbruchbedingungen für den Algorithmus, nämlich wenn alle Klauseln gestrichen wurden (erfüllbar) und wenn keine weitere Klausel der Form $\{x_j\}$ existiert (erfüllbar nach Beobachtung 2).

Man kann die Idee für den Markierungsalgorithmus auch auf einen speziellen Resolutionskalkül für Hornformeln übertragen. Dabei wird auf das Streichen der erfüllten Klauseln verzichtet, d.h. es gibt wie bisher nur die Abbruchbedingungen, dass man die leere Klausel deduzieren kann oder dass man keine neuen Resolventen bilden kann. Der Unterschied zum vorher beschriebenen Resolutionskakül besteht aber darin, dass man für unerfüllbare Hornformeln die leere Klausel herleiten kann, indem man ausschließlich Resolventen aus einer positiven Klausel $\{x_i\}$ mit einer anderen Klausel bildet.

Definition: Sei $\mathcal{K}$ die Klauselmenge einer Hornformel. Ein Resolvent R aus den Klauseln $K_i, K_j \in \mathcal{K}$ wird *Einheitsresolvent* genannt, falls $|K_i| = 1$ oder $|K_j| = 1$. Analog zu $Res(\mathcal{K})$ und $Res^*(\mathcal{K})$ definiert man $1\text{–}Res(\mathcal{K})$ und $1\text{–}Res^*(\mathcal{K})$ mit Einheitsresolventen anstelle von Resolventen. Der Resolutionssatz tritt dann in folgender Gestalt auf.

Satz: Eine durch die Klauselmenge $\mathcal{K}$ dargestellte Hornformel ist genau dann unerfüllbar, wenn $\square \in 1\text{–}Res^*(\mathcal{K})$.

Da die Einheitsresolution die Länge der Klauseln ständig verkürzt, ist diese Methode (für Hornklauseln!) besonders effizient.

Hornformeln und der Markierungsalgorithmus sind eine wichtige Grundlage von logischen Programmiersprachen, insbesondere von der Sprache PROLOG.

7.4 Algebraische Strukturen und Prädikatenlogik

Der Aufbau der Prädikatenlogik hat starke Bezüge zur Beschreibung von mathematischen, insbesondere *algebraischen Strukturen*. Darunter versteht man eine Trägermenge (Individuenbereich) in der gewisse Relationen, Operationen und Konstanten ausgezeichnet sind, die bestimmte Eigenschaften aufweisen.

Beispiel 1: Gruppen

Definition: Eine *Gruppe* $(G, *)$ besteht aus einer Trägermenge G und einer Operation $* : G \times G \longrightarrow G$ mit den folgenden drei Eigenschaften:

(G1) $\forall a, b, c \in G \quad (a * b) * c = a * (b * c)$ (Assoziativität)

(G2) $\exists e \in G \quad \forall a \in G \quad a * e = a = e * a$ (e ist neutrales Element)

(G3) $\forall a \in G \quad \exists \bar{a} \in G \quad a * \bar{a} = e = \bar{a} * a$ ($\bar{a}$ ist inverses Element zu a)

Bekannte Beispiele (man spricht hier von Modellen) sind die ganzen Zahlen $\mathbb{Z}$ mit der Addition als Operation, dem neutralen Element 0 und $-z$ als Zahl, die zu z invers ist, oder die positiven reellen Zahlen $\mathbb{R}^+$ mit Multiplikation, dem neutralen Element 1 und $r^{-1} = \frac{1}{r}$ als Element, das zu r invers ist.

Man bezeichnet die Eigenschaften (G1), (G2) und (G3) auch als die drei Gruppenaxiome. Viele andere Eigenschaften, die alle Gruppen gemeinsam haben, kann man aus diesem Axionem ableiten. Die Gesamtheit der in allen Gruppen gültigen Fakten und Sätze bildet die sogenannte Gruppentheorie. Eine zentrale Motivation für die Entwicklung einer solchen Theorie ist die folgende Vorgehensweise: Es reicht aus, für ein konkretes Modell die Gültigkeit der drei Axiome nachzuweisen, um alle Sätze der Theorie auf das Modell anwenden zu können.

Exemplarisch werden wir hier drei (sehr einfache) Fakten der Gruppentheorie abzuleiten:

1) Das neutrale Element in einer Gruppe ist eindeutig.

Beweis: Angenommen zwei Elemente e und e' einer Gruppe erfüllen (G2). Dann wäre $e * e' = e'$ wegen (G2) für e und $e * e' = e$ wegen (G2) für e'. Damit muss $e = e'$ sein, d.h. das neutrale Element ist eindeutig. □

2) In einer Gruppe ist für jedes Element $a \in G$ das zu a inverse Element eindeutig.

Beweis: Angenommen $\bar{a}$ und $\tilde{a}$ erfüllen beide (G3). Wir betrachten das Gruppenelement $b = (\bar{a} * a) * \tilde{a}$:

$$\begin{array}{rll} (\bar{a} * a) * \tilde{a} = b = & \bar{a} * (a * \tilde{a}) & |(G1) \\ e * \tilde{a} = b = & \bar{a} * e & |(G3) \text{ für } \bar{a} \text{ und für } \tilde{a} \\ \tilde{a} = b = & \bar{a} & |(G2) \end{array}$$ □

3) In einer Gruppe hat jede Gleichung der Form $(a * x) * b = c$ eine eindeutige Lösung für die Varaible x.

Beweis: Die Gleichung wird äquivalent umgeformt, d.h. jeder Schritt der Umformung ist auch umkehrbar. Werden beide Seiten der Gleichung von rechts mit dem zu b inversen Element verknüpft, und auf der linken Seite (G3) und (G2) angewendet, enhält man

$$(a * x) * b = c \Longleftrightarrow a * x = c * \bar{b}$$

Werden analog beide Seiten der neuen Gleichung von links mit $\bar{a}$ verknüpft , so ergibt sich die eindeutige Lösung der Gleichung:

$$a * x = c * \bar{b} \Longleftrightarrow x = \bar{a} * (c * \bar{b})$$ □

Wegen der Eindeutigkeit des inversen Elements kann man an Stelle des Existenzquators in (G3) auch eine Funktion verwenden, die jedem Gruppenelement sein inverses Element zuordnet und eine solche Funktion ist gleichzeitig eine einstellige Operation (eine k-stellige Operation ordnet jedem k-Tupel von Elementen ein Element als Operationsergebnis zu). Analog kann die Existenz des neutralen Elements auch als 0-stellige Operation angesehen werden. Unter dieser Sichtweise können die Gruppenaxiome auch ohne Existenzquantoren formuliert werden:

Definition': Eine *Gruppe* $(G,*)$ besteht aus einer Trägermenge G und einer binären Operation $*: G\times G \longrightarrow G$, einer einstelligen Operation $^-$: $G \longrightarrow G$ und einem konstanten Element $e \in G$ (alternativ einer 0-stelligen Operation) mit den folgenden drei Eigenschaften:

(G1') $\forall a,b,c \in G \quad (a*b)*c = a*(b*c)$ (Assoziativität)

(G2') $\forall a \in G \quad a*e = a = e*a$ (e ist neutrales Element)

(G3') $\forall a \in G \quad a*\bar{a} = e = \bar{a}*a$ ($\bar{a}$ ist inverses Element zu a)

Beispiel 2: Verbände

Ein *Verband* (englisch *Lattice*) ist eine Struktur mit einer Halbordnungsrelation mit der zusätzlichen Eigenschaft, dass alle 2-elementigen (damit letzlich auch alle endlichen) Teilmengen eine obere und untere Grenze (also Supremum und Infimum) haben. Dieses Beispiel soll demonstrieren, dass neben den Operationen auch Relationen sowie Verknüpfungen von beiden eine zentrale Rolle bei der Beschreibung einer mathematischen Struktur spielen können. Die nachfolgende Definition ist nicht die Standarddefinition, sondern eine Beschreibung nach den in Abschnitt 2.4 gegebenen Definitionen von Infimum und Supremum (alternativ gibt es auch eine äquivalente Definition, in der die Halbordnungsrelation nicht explizit verwendet wird).

Definition: Eine Menge L mit einer Relation $\leq \subseteq L\times L$ und zwei Funktionen (oder binären Operationen) $sup, inf : L\times L \longrightarrow L$ ist ein *Verband* (englisch *Lattice*), falls

(L1) $\forall x \in L \quad x \leq x$ (Reflexivität)

(L2) $\forall x \in L\, \forall y \in L \quad x \leq y \wedge y \leq x \Rightarrow x = y$ (Antisymmetrie)

(L3) $\forall x \in L\, \forall y \in L\, \forall z \in L \quad x \leq y \wedge y \leq z \Rightarrow x \leq z$ (Transitivität)

(L4) $\forall x \in L\, \forall y \in L \quad (x \leq sup(x,y) \wedge y \leq sup(x,y) \wedge \forall z \in L \quad x \leq z \wedge y \leq z \Rightarrow sup(x,y) \leq z)$
(sup ist obere Schranke und kleinste obere Schranke)

(L5) $\forall x \in L\, \forall y \in L \quad (inf(x,y) \leq x \wedge inf(x,y) \leq y \wedge \forall z \in L \quad z \leq x \wedge z \leq y \Rightarrow z \leq inf(x,y))$
(inf ist untere Schranke und größte untere Schranke)

Einfache Beispiele sind der Mengenverband $(\mathcal{P}(M),\subseteq,\cup,\cap)$ und der Teilbarkeitsverband $(\mathbb{Z}^+,|,kgV,ggT)$.

Zu den Eigenschaften, die man aus diesen Axiomen ableiten kann, gehören das Assoziativ- und das Kommutativgesetz für die Operationen sup und inf. Das Distributivgesetz kann nicht aus den Axiomen abgeleitet werden (obwohl die genannten Beispiel das vielleicht suggerieren könnten).

7.5 Elementare Sprachen

Wie die Beispiele aus dem letzten Abschnitt zeigen, werden zur Charakterisierung von mathematische Strukturen quantifizierte Prädikate verwendet. Der allgemeine Zugang zu

einer solchen Beschreibung wird im Folgenden formalisiert. Er führt zu den sogenannten *Sprachen 1. Stufe*, auch als *elementare Sprachen* bekannt. Wie schon in der Aussagenlogik beginnen wir mit der Syntax dieser Sprachen und wenden uns danach der Semantik zu. Ein wesentlicher Unterschied zur Aussagenlogik wird darin bestehen, dass man die Begriffe *Term* und *Formel* nicht mehr als Synomyme verwenden kann: In der Prädikatenlogik ist eine Term ein syntaktischer Ausdruck, der auf der semantischen Seite ein Element aus der Struktur beschreibt, d.h. bei einer Auswertung nimmt ein Term immer einen Wert aus der Struktur an. Dagegen ist eine Formel ein syntaktischer Ausdruck, der auf der semantischen Seite eine Aussage beschreibt, also bei einer Auswertung immer einen Wahrheitswert ergibt.

Definition: Das Alphabet einer elementaren Sprache besteht aus:

- einer Menge von Individuenvariablen $Var = \{x_1, x_2, \ldots\}$, wobei x, y, z beliebige Elemente aus dieser Menge bezeichnen sollen;
- den logischen Symbolen $\neg, \wedge, \vee$ sowie dem *Allquantor* $\forall$, dem *Existenzquantor* $\exists$ und dem Zeichen $\equiv$ für die *Identitätsrelation* in der zugrundeliegenden Struktur, weitere logische Symbole wie $\rightarrow$ und $\leftrightarrow$ können später durch Definition eingeführt werden;
- der *nichlogischen Signatur L*, die sich aus einer Menge von Konstantensymbolen (sie werden mit a, b, c bezeichnet), einer Menge von Funktionssymbolen (sie werden mit f, g, h bezeichnet) sowie einer Menge von Prädikatsymbolen (sie werden mit P, R, S bezeichnet). Man setzt dabei voraus, dass jedes Funktionssymbol die Form f_i^k hat wobei der Index i zur Nummerierung der verwendeten Symbole dient und die zusätzliche Zahl k die Stelligigkeit der entsprechenden Funktion beschreibt. Funktionen mit der Stelligkeit $k = 2$ stehen für die üblichen binären Verknüpfungen. Funktionen mit der Stelligkeit $k = 0$ sind eine alternative Schreibweise für Konstanten. Analog wird vorausgesetzt, dass jedes Prädikatsymbol die Form P_i^k hat, wobei wieder i zur Nummerierung und k zur Beschreibung der Stelligkeit verwendet wird. Da insbesondere die zweistelligen Prädikate nur eine andere Darstellung einer Relation sind, werden Prädikatsymbole oft auch Relationssymbole genannt.

Die Syntax der Sprache 1. Stufe über der Signatur L wird durch die (induktive) Definition von *Termen* und *Formeln* bestimmt.

Definition von Termen in L:

1. Alle Variablen und Konstantensymbole sind Terme, die sogenannten *Primterme*.
2. Ist $f = f_i^k \in L$ ein k–stelliges Funktionssymbol und sind $t_1, \ldots, t_k$ Terme, so ist auch $f(t_1, \ldots, t_k)$ ein Term.

Analog wie für Formeln der Aussagenlogik gilt auch hier ein Satz von der eindeutigen Termreduktion (Ist $f(t_1, \ldots, t_k) = g(s_1, \ldots, s_l)$ dann ist $f = g$, $k = l$ und $t_1 = s_1, \ldots, t_k = s_k$) und das Beweisprinzip durch Terminduktion (Haben alle Primterme eine bestimmte Eigenschaft und gilt, dass für beliebige Terme $t_1, \ldots, t_k$ mit dieser Eigenschaft und für alle k–stelligen Funktionssymbole f auch $f(t_1, \ldots, t_k)$ diese Eigenschaft hat, dann haben alle Terme diese Eigenschaft.)

Auch die Mengen $St(t)$ der Subterme eines Terms t und $var(t)$ der in t auftretende Variablen werden induktiv definiert:

- Ist c eine Konstante, dann ist $St(c) = \{c\}$ und $var(c) = \emptyset$.
- Ist x eine Variable, dann ist $St(x) = \{x\}$ und $var(x) = \{x\}$.
- Ist $t = f(t_1, \ldots, t_n)$, dann ist $St(t) = \{t\} \cup \bigcup_{i=1}^{n} St(t_i)$ und $var(t) = \bigcup_{i=1}^{n} var(t_i)$.

Definition von Formeln in L:

1. Sind s,t Terme, so ist $s \equiv t$ eine Formel.
2. Sind $t_1, \ldots, t_k$ Terme und ist $P \in L$ ein k–stelliges Prädikatsymbol, so ist $P(t_1, \ldots, t_k)$ eine Formel.
3. Sind F, G Formeln und ist $x \in Var$, so sind auch $(\neg F), (F \wedge G), (F \vee G)$ und $\forall x F$ Formeln.

Bemerkungen:

- Die durch 1) und 2) gewonnenen Formeln werden als Primformeln bezeichnet.
- Für das Weglassen von Klammern gelten die gleichen Regeln wie in der Aussagenlogik, aber zusätzlich muss man noch die Quantoren und das Symbol $\equiv$ einreihen. Es wird vereinbart, dass $\equiv$ die größte Bindungstärke besitzt, gefolgt von den Quantoren und den logischen Operationen in der bekannten Reihenfolge $\neg, \wedge, \vee$. Geschachtelte Quantoren sind rechtsassoziativ zu klammern.
  Beispiel: Die Formel $\forall x \ \neg x \equiv a \vee \exists y \ y \equiv b \wedge \neg a \equiv b$ bekommt bei vollständiger Klammerung die Gestalt $(\forall x\,(\neg(x \equiv a))) \vee ((\exists y\ (y \equiv b)) \wedge (\neg(a \equiv b)))$
- Weitere logische Symbole können als Abkürzungen eingeführt werden, wie z.B. $F \Rightarrow G := \neg F \vee G$. Prinzipiell könnte man die logische Signatur noch weiter auf $\{\neg, \wedge, \forall, \equiv\}$ einschränken und auch Disjunktion sowie Existenzquantor mit $F \vee G := \neg(\neg F \vee \neg G)$ und $\exists x\ F := \neg\forall x\ \neg F$ darauf zurückführen.
- Wie die Formeldefinition deutlich macht, ist Quantifizierung nur für Individuenvariable vorgesehen. Erweitert man dieses Konzept um Variablen für Untermengen des Individuenbereichs oder für Funktionen und deren Quantifizierung, so erhält man *Sprachen 2. Stufe*, die jedoch andere semantische Eigenschaften haben.

Beispiel: Die direkte Beschreibung der drei Gruppenaxiome erfodert in der Signatur L nur ein zweistelliges Funktionssymbol f. Der Ausdruck $x * (y * z)$ wird dann durch den Term $f(x, f(y,z))$ repräsentiert:

$$\begin{array}{lll} (G1) & \forall x \forall y \forall z & f(x, f(y,z)) \equiv f(f(x,y), z) \\ (G2) & \exists e \forall x & (f(e,x) \equiv x \ \wedge \ f(x,e) \equiv x) \\ (G3) & \forall x \exists y & (f(x,y) \equiv e \ \wedge \ f(y,x) \equiv e) \end{array}$$

Aus den Beispielen zur Prädikatenlogik in Kapitel 1 kennen wir bereits die Vorgehensweise, aus Prädikaten durch Hinzufügung von Quantoren für alle vorkommenden

Variablen eine Aussage zu machen. Um dies in das allgemeine Konzept der Sprachen erster Ordnung übertragen zu können, muss man zwischen freien und (durch Quantoren) gebundenen Variablen unterscheiden können.

Definition: Die Mengen der in einer Formel F auftretende Variablen $var(F)$ bzw. der in F frei vorkommenden Variablen $frei(F)$ werden induktiv aus den Variablenmengen der in F auftretenden Terme abgeleitet:

- Ist F eine Primformel vom Typ $s \equiv t$, so ist
  $var(F) = frei(F) = var(s) \cup var(t)$.
- Ist F eine Primformel vom Typ $P(t_1, \ldots, t_k)$, so ist
  $var(F) = frei(F) = \bigcup_{i=1}^{k} var(t_i)$
- Ist F eine Formel vom Typ $\neg G$, so ist
  $var(F) = var(G)$ und $frei(F) = frei(G)$
- Ist F eine Formel vom Typ $G \wedge H$ oder $G \vee H$, so ist
  $var(F) = var(G) \cup var(H)$ und $frei(F) = frei(G) \cup frei(H)$.
- Ist F eine Formel vom Typ $\forall x\, G$ oder $\exists x\, G$, so ist
  $var(F) = var(G)$ und $frei(F) = frei(G) \setminus \{x\}$.

Formeln ohne freie Variable heißen *geschlossene Formeln* oder *Aussagen*.

Beispiele:

- Die Formeln $\forall x\; x \equiv f(x,x)$, $\forall x \forall y\; x \equiv y$ und $\forall x\; \neg \forall y \forall x\; f(x,y) \equiv f(y,x)$ sind Aussagen.
- $frei(\,\forall x\; \neg x \equiv a \wedge \forall y\; y \equiv f(x,z)\,) = \{x,z\}$
- $frei(\,\forall x\,(\neg x \equiv a \wedge \forall y\; y \equiv f(x,z))\,) = \{z\}$
- In der obigen Beschreibung der Gruppenaxiome ist e eine Variable und ein Nachteil besteht darin, dass das Axiom $(G3)$ keine Aussage ist, denn es enthält die freie Varaible e. Ein Ausweg besteht darin, e als Konstantensymbol in L aufzunehmen, aber dann kann man konsequenterweise auch ein zusätzliches einstelliges Funktionssymbol g einführen, so dass $g(x)$ das zu x inverse Gruppenelement beschreibt:

$$\begin{array}{lll} (G1) & \forall x \forall y \forall z & f(x,f(y,z)) \equiv f(f(x,y),z) \\ (G2) & \forall x & (f(e,x) \equiv x \;\wedge\; f(x,e) \equiv x) \\ (G3) & \forall x & (f(x,g(x)) \equiv e \;\wedge\; f(g(x),x) \equiv e) \end{array}$$

Semantik elementarer Sprachen

Zuerst benötig man die Begriffe L–Stuktur und L–Modell für eine gegebene nichtlogische Signatur L. Dazu muss eine konkrete Trägermenge festgelegt werden, auf der alle Funktions-, Prädikat- und Konstantensymbole durch konkrete Funktionen, Prädikate und Konstanten realisiert werden. Gleichzeitig ist die Trägermenge der Individuenbereich für die Bewertung aller Variablen und für alle Quantoren.

Definition: Eine L–Struktur $\mathcal{A} = (A, L^{\mathcal{A}})$ besteht aus einer nichtleeren Trägermenge A und einer Menge $L^{\mathcal{A}}$, die für jedes Konstantensymbol $c \in L$ eine Konstante $c^{\mathcal{A}} \in A$,

für jedes k–stellige Funktionssymbol $f \in L$ eine Funktion $f^{\mathcal{A}} : A^k \longrightarrow A$ und für jedes k–stellige Prädikatsymbol $P \in L$ eine Funktion $P^{\mathcal{A}} : A^k \longrightarrow \mathbb{B}$ enthält.

Definition: Eine L–Struktur $\mathcal{A} = (A, L^{\mathcal{A}})$ mit einer zusätzlichen Variablenbelegung $\omega : Var \longrightarrow A$ wird ein L–Modell genannt. Man verwendet die Bezeichnung $\mathcal{M} = (\mathcal{A}, \omega)$ und überträgt die Notationen der Struktur entsprechend durch $c^{\mathcal{M}} = c^{\mathcal{A}}$, $f^{\mathcal{M}} = f^{\mathcal{A}}$, $P^{\mathcal{M}} = P^{\mathcal{A}}$ und $x^{\mathcal{M}} = \omega(x)$.

Achtung: In einigen Büchern (z.B. Schöning) sind die Funktionen $\omega : Var \longrightarrow A$ bereits Bestandteil einer Struktur, d.h. eine Struktur in jenem Sinne ist ein L–Modell in Sinne unserer Definition. Die hier getroffene Unterscheidung zwischen Struktur und Modell ist aber sinnvoll, denn wenn z.B. bei den ganzen Zahlen mit Addition von einer Gruppenstruktur gesprochen wird, hat man sicherlich noch keine Belegung für eventuell auftretende Variablen x, y, z festgelegt, sondern will sich das noch offenhalten.

Auswertung von Termen und Formeln

Durch induktive Definition weist ein Modell $\mathcal{M}$ jedem Term t einen Wert $t^{\mathcal{M}}$ aus der Trägermenge A zu: Für Primterme sind $c^{\mathcal{M}} = c^{\mathcal{A}}$ und $x^{\mathcal{M}} = \omega(x)$ bereits definiert, für $t = f(t_1, \ldots, t_k)$ setzen wir $t^{\mathcal{M}} := f^{\mathcal{M}}(t_1^{\mathcal{M}}, \ldots, t_k^{\mathcal{M}})$.

Dadurch kann bei einem gegebenen Modell auch jeder Formel F (induktiv) ein Wahrheitswert $\mathcal{M}(F)$ zugeordnet werden. Das ist einfach für Formeln vom Typ 1 bis 3:

1. Der Wahrheitswert einer Formel vom Typ $s \equiv t$ ist genau dann 1, wenn $s^{\mathcal{M}}$ und $t^{\mathcal{M}}$ (als Elemente von A) identisch sind.

2. Eine Formel vom Typ $P(t_1, \ldots, t_k)$ bekommt den Wert $P^{\mathcal{M}}(t_1^{\mathcal{M}}, \ldots t_k^{\mathcal{M}})$.

3. Der Wahrheitswert einer Formel F vom Typ $\neg G$, $G \wedge H$ bzw. $G \vee H$ wird in naheliegender Weise durch $\neg\mathcal{M}(G)$, $\mathcal{M}(G) \wedge \mathcal{M}(H)$ bzw. $\mathcal{M}(G) \vee \mathcal{M}(H)$ festgelegt.

Komplikationen bereiten Formeln mit Quantoren. Zu ihrer Bewertung werden Modelle mit der Bezeichnung $\mathcal{M}_{[x/d]}$ verwendet, welche auf der gleichen Struktur wie $\mathcal{M}$ basieren und auch für alle Variablen die gleiche Belegung wie $\mathcal{M}$ haben mit der einzigen Ausnahme, dass die Variable x die Belegung d bekommt.

4. Eine Formel F von Typ $\forall x G$ bekommt den Wahrheitswert

$$\mathcal{M}(F) = \begin{cases} 1 & \text{falls für alle } d \in A \text{ gilt: } \mathcal{M}_{[x/d]}(G) = 1 \\ 0 & \text{sonst} \end{cases}$$

5. Eine Formel F von Typ $\exists x G$ bekommt den Wahrheitswert

$$\mathcal{M}(F) = \begin{cases} 1 & \text{falls es ein } d \in A \text{ gibt mit: } \mathcal{M}_{[x/d]}(G) = 1 \\ 0 & \text{sonst} \end{cases}$$

Achtung: Durch die induktive Definition hat die Neubelegung von x durch d in $\mathcal{M}_{[x/d]}$ keinen Einfluss auf Teilformeln von G in denen die Variable x nur gebunden vorkommt. Wir verdeutlichen diesen Umstand am Beispiel einer Struktur mit einem Konstantensymbol a und einer Formel $F = \forall x G$ der folgenden Gestalt:

$$F = \forall x(x \equiv a \vee \forall x \neg(x \equiv a))$$

Demnach ist G die Disjunktion aus den Formeln $H = x \equiv a$ und $J = \forall x \neg(x \equiv a))$. Wenn wir F über der Struktur der natürlichen Zahlen mit dem Wert 5 für die Konstante a auswerten wollen, ist die Subformel J bereits eine Aussage, denn J enthält keine freien Variablen. Es ist klar, dass J eine falsche Aussage ist (also den Wert 0 bekommt), weil (unabhängig vom betrachteten Modell $\mathcal{M}$) im Modell $\mathcal{M}_{[x/5]}$ die Formel $\neg(x \equiv a)$ den Wert 0 hat. Somit nimmt bei der Auswertung von $F = \forall x G$ die Formel G nur für die Modelle in denen x auf 5 abgebildet wird den Wert 1 an, aber wir haben (wieder unabhängig vom Modell) $\mathcal{M}_{[x/4]}(G) = 0$, d.h. F ist eine falsche Aussage. Wir werden diese etwas verwirrende Situation später durch das Umbenennen von Variablen auflösen.

Ähnlich wie in der Aussagenlogik führen wir jetzt die folgenden Begriffe und Notationen ein:

- Bekommt eine Formel F unter dem Modell $\mathcal{M}$ den Wahrheitswert 1, dann sagt man, dass $\mathcal{M}$ die Formel F erfüllt und schreibt dafür $\mathcal{M} \models F$.
- Zwei Formeln F und G heißen *logisch* oder *semantisch äquivalent* (Schreibweise $F \equiv G$), falls für jedes Modell $\mathcal{M}$ gilt: $\mathcal{M} \models F$ genau dann, wenn $\mathcal{M} \models G$.
- $\mathcal{M}$ ist ein *Modell für eine Formelmenge X* (Schreibweise: $\mathcal{M} \models X$), wenn $\mathcal{M}$ jede Formel aus X erfüllt. X heißt *erfüllbar*, wenn es ein Modell besitzt.
- Eine Formel F heißt *allgemeingültig* oder *Tautologie* (Schreibweise: $\models F$), wenn $\mathcal{M} \models F$ für jedes Modell $\mathcal{M}$.
- Eine Formel F *gilt in* $\mathcal{A}$ (Schreibweise: $\mathcal{A} \models F$), falls $(\mathcal{A}, \omega) \models F$ für alle $\omega : Var \longrightarrow A$ und entsprechend $\mathcal{A} \models X$ falls $\mathcal{A} \models F$ für alle $F \in X$.
- Man sagt, daß F aus X folgt (Schreibweise: $X \models F$), falls jedes Modell für X auch F erfüllt.

Beispiele:

1. Wir betrachten die Formel $F = \forall x \exists y \neg x \equiv y$. Wie man leicht sieht, ist F erfüllbar, denn jedes Modell mit einer zwei– oder mehrelementigen Trägermenge erfüllt F. Aber F ist keine Tautologie, weil Modelle mit einelementiger Trägermenge F nicht erfüllen. Ist $\mathcal{A} = (A, L^{\mathcal{A}})$ eine L–Struktur, so gilt $\mathcal{A} \models F$, falls $|A| \geq 2$, und $\mathcal{A} \not\models F$, falls $|A| = 1$.
2. Die Formel $G = \neg \forall x \forall y \neg x \equiv y$ ist eine Tautologie.
3. Sei $\mathcal{A}$ eine Struktur mit einer zweielementigen Trägermenge, dann gilt weder $\mathcal{A} \models x \equiv y$ noch $\mathcal{A} \models \neg x \equiv y$. Ist dagegen F eine beliebige Formel und $\mathcal{M}$ ein beliebiges (zur Signatur passendes) Modell, dann gilt entweder $\mathcal{M} \models F$ oder $\mathcal{M} \models \neg F$.

Wie das Beispiel der Gruppentheorie deutlich macht, möchte man Stukturen betrachten, in denen die Gültigkeit einer bestimmten Formelmenge X vorausgesetzt wird. Ziel ist es dann, Aussagen zu erhalten, die aus X folgen. Dieser Ansatz führt zu der folgenden Begriffsbildung.

Definition: Eine *Theorie* ist eine Menge von Formeln $\mathcal{T}$, die bezüglich Folgerung abgeschlossen ist, d.h. $\mathcal{T} \models F$ impliziert $F \in \mathcal{T}$. Eine Formelmenge X nennt man *Axiomensystem* der Theorie $\mathcal{T}$, falls $\mathcal{T} = \{F \mid X \models F\}$.

Es sei angemerkt, daß auch die Menge aller L-Formeln eine Theorie ist. Man nennt dies eine entartete Theorie, da sie widersprüchliche Aussagen enthält und es kein Modell für sie gibt.

Sei $\mathcal{T}$ eine Theorie mit einem endlichen Axiomensystem $X = \{F_1, \ldots F_n\}$ und F eine Formel. Dann ist F genau dann in $\mathcal{T}$, wenn die Formel $G = \neg F_1 \vee \ldots \vee \neg F_n \vee F$ eine Tautologie ist. Äquivalent formuliert, ist F genau dann in $\mathcal{T}$, wenn die Formel $\neg G = F_1 \wedge \ldots \wedge F_n \wedge \neg F$ nicht erfüllbar ist.

Es ergibt sich die Frage, ob es (ähnlich dem Resolutionskalkül in der Aussagenlogik) ein Entscheidungsverfahren für die Erfüllbarkeit von Formeln elementarer Sprachen gibt. Die Antwort ist negativ: Das Erfüllbarkeitsproblem für Formeln elementarer Sprachen ist **nicht entscheidbar**. Das bedeutet nicht nur, dass man keinen Algorithmus zur Beantwortung dieser Frage kennt, sondern dass es (beweisbar!) keinen solchen Algorithmus gibt.
Trotzdem gibt es aber auch noch eine gute Nachricht: Es konnte ein sogenanntes Semientscheidungsverfahren entwickelt werden, welches zumindest für unerfüllbare Formeln in endlicher Zeit mit der richtigen Antwort stoppt, aber für erfüllbare Formeln eventuell nie abbricht. Dieses Verfahren und insbesondere seine theoretischen Grundlagen übersteigen den Rahmen unserer Vorlesung. Wir wollen aber zumindest die ersten Schritte in dieser Richtung verfolgen, bei denen es darum geht, Formeln aus Sprachen der ersten Stufe äquivalent so umzuwandeln, dass sie gewisse Normalformen annehmen, welche sich später als wichtige Voraussetzung für die Anwendung des Semientscheidungsverfahren herausstellen werden.

7.6 Normalformen

Zur Umwandlung von Formeln in eine der (noch einzuführenden) Normalformen wird ein Vorrat an äquivalenten Umformungen benötigt. Die Wichtigtsen sind im nachfolgenden Satz zusammengestellt.

Satz: Für beliebige Formeln F und G gelten die folgenden Äquivalenzen:

$$(1) \qquad \neg\forall x\, F \equiv \exists x \neg F \qquad\qquad \neg\exists x\, F \equiv \forall x \neg F$$

$$(2) \qquad \forall x\, F \wedge \forall x\, G \equiv \forall x\, (F \wedge G) \qquad\qquad \exists x\, F \vee \exists x\, G \equiv \exists x\, (F \vee G)$$

$$(3) \qquad \forall x \forall y\, F \equiv \forall y \forall x\, F \qquad\qquad \exists x \exists y\, F \equiv \exists y \exists x\, F$$

Ist G eine Formel mit $x \notin frei(G)$, dann gilt darüber hinaus:

$$(4) \qquad (\forall x\, F) \wedge G \equiv \forall x\, (F \wedge G) \qquad (\forall x\, F) \vee G \equiv \forall x\, (F \vee G)$$

$$(5) \qquad (\exists x\, F) \wedge G \equiv \exists x\, (F \wedge G) \qquad (\exists x\, F) \vee G \equiv \exists x\, (F \vee G)$$

Zur wirkungsvollen Nutzung der in diesem Satz enthaltenen Regeln kann analog zur Situation in der Aussagenlogik ein Substitutionstheorem verwendet werden.

Definition: Ist F eine Formel, x eine Variable und t ein Term, dann bezeichnet $F[x/t]$ die aus F entstehende Formel, wenn jedes freie Auftreten von x in F durch den Term t ersetzt wird.

In Analogie zur Aussagenlogik kann aus der Formeläquivalenz $F \equiv G$ auf die Äquivalenz der substituierten Formeln $F[x/t] \equiv G[x/t]$ geschlossen werden.

Eine typische Anwendung der obigen Regeln in Kombination mit den Gesetzen der Booleschen Algebra zeigt das folgende Beispiel über einer Struktur mit einem Konstantensymbol a, einem zweistelligen Funktionssymbol f und einem zweistelligen Prädikatsymbol P:

$$
\begin{array}{ll}
\neg(\forall x \neg x \equiv y \wedge \exists z P(y,z)) \vee \forall w \exists x P(f(a,w),x) & \\
\equiv (\neg\forall x \neg x \equiv y \vee \neg\exists z P(y,z)) \vee \forall w \exists x P(f(a,w),x) & \text{deMorgan} \\
\equiv (\exists x x \equiv y \vee \forall z \neg P(y,z)) \vee \forall w \exists x P(f(a,w),x) & \text{(1) und Doppelnegation} \\
\equiv \exists x (x \equiv y \vee \forall z \neg P(y,z)) \vee \forall w \exists x P(f(a,w),x) & \text{(5)} \\
\equiv \forall w \exists x P(f(a,w),x) \vee \exists x (x \equiv y \vee \forall z \neg P(y,z)) & \text{Kommutativgesetz} \\
\equiv \forall w (\exists x P(f(a,w),x) \vee \exists x (x \equiv y \vee \forall z \neg P(y,z))) & \text{(4)} \\
\equiv \forall w \exists x (P(f(a,w),x) \vee (x \equiv y \vee \forall z \neg P(y,z))) & \text{(2)} \\
\equiv \forall w \exists x (\forall z \neg P(y,z) \vee (P(f(a,w),x) \vee x \equiv y)) & \text{Assoz.- u. Kommutativgesetz} \\
\equiv \forall w \exists x \forall z (\neg P(y,z) \vee P(f(a,w),x) \vee x \equiv y)) & \text{(5)}
\end{array}
$$

Darüber hinaus kann man die Substitutionen aber auch zur sogenannten *gebundenen Umbenennung* von Variablen verwenden. Im Folgenden werden wir $Q, Q_1, Q_2, \ldots$ für beliebige Quantorensymbole (also jeweils $\forall$ oder $\exists$) verwenden und mit $\overline{Q}$ den jeweils anderen Quantor bezeichnen (also $\overline{\forall} = \exists$ und $\overline{\exists} = \forall$)

Lemma: Sei $F = Qx\,G$ eine Formel und $Q \in \{\exists, \forall\}$. Ist y eine Variable, die in G nicht vorkommt, dann gilt $F \equiv Qy\,G[x/y]$.

Durch wiederholte Anwendung dieses Lemmas kann man jede Formel F äuivalent in eine *bereinigte* Formel G umwandeln, wobei der Begriff *bereinigt* bedeutet, dass es in G keine Variable gibt, die sowohl gebunden als auch frei vorkommt, und hinter allen vorkommenden Quantoren verschiedene Variablen stehen.

Definition: Eine Formel F ist *pränex* (oder in *Pränexform*), wenn sie die Gestalt $Q_1 y_1\, Q_2 y_2 \ldots Q_n y_n\, G$ hat , wobei $n \geq 0$ und in G keine weiteren Quantoren vorkommen. Wir verwenden die Abkürzung **BPF** für bereinigte Formeln in Pränexform.

Satz: Jede Formel F kann äquivalent in eine Formel G in **BPF** umgewandelt werden.

Der Beweis erfolgt durch Induktion über den Formelaufbau, wobei beim Induktionsanfang für Primformeln keine Umwandlungen notwendig sind. Beim Induktionsschritt muss man drei Fälle unterscheiden.

- Hat F die Form $\neg F_1$, dann kann nach Induktionsvoraussetzung F_1 in eine **BPF** $Q_1 y_1\, Q_2 y_2 \ldots Q_n y_n\, G_1$ umgewandelt werden und mit den Regeln aus Punkt (1) im obigen Satz gilt $F \equiv \overline{Q_1} y_1\, \overline{Q_2} y_2 \ldots \overline{Q_n} y_n\, \neg G_1$.

- Hat F die Form $F_1 \wedge F_2$ (oder analog die Form $F_1 \vee F_2$), dann verwendet man wieder nach Induktionsvoraussetzung die äquivalenten **BPF** $Q_1 y_1\, Q_2 y_2 \ldots Q_n y_n\, G_1$ und $Q'_1 z_1\, Q'_2 z_2 \ldots Q'_m z_m\, G_2$, macht durch gebundene Umbenennung deren Variablenmengen disjunkt (aus z_i wird z'_i und aus G_2 wird G'_2) und erhält somit $F \equiv Q_1 y_1\, Q_2 y_2 \ldots Q_n y_n\, Q'_1 z'_1\, Q'_2 z'_2 \ldots Q'_m z'_m\, G_1 \wedge G'_2$ in **BPF**.

- Hat F die Form QxF_1 und ist F_1 nach Induktionsvoraussetzung äquivalent zu $Q_1 y_1\, Q_2 y_2 \ldots Q_n y_n\, G_1$ (wobei x nach einer eventuellen Umbenennung nicht in $\{y_1, y_2, \ldots, y_n\}$ vorkommt), dann ist F äquivalent zu $Qx\, Q_1 y_1\, Q_2 y_2 \ldots Q_n y_n\, G_1$. Damit ist der Beweis abgeschlossen.

Der nächste Schritt hat das Ziel, alle Existenzquantoren aus einer bereinigten Pränexform zu eliminieren. Wir haben die Idee dafür bereits bei der Übersetzung der Gruppenaxiome vorweggenommen:

- Ursprünglich gab es in Gruppen nur eine binäre Operation $*$, welche durch ein zweistelliges Funktionsymbol f realisiert wird. Bei $(G1)$ ist nichts zu tun, denn dieses Axiom enthält nur Allquantoren: $\forall x \forall y \forall z\, f(f(x,y),z) \equiv f(x,f(y,z))$.

- $(G2)$ wurde ursprünglich durch $\exists e \forall x (f(e,x) \equiv x \wedge x \equiv f(x,e))$ beschrieben. Hier ist e ein Variablensymbol (und sollte wohl besser y oder z heißen). Die Tatsache, dass solch ein e existiert, kann aber auch durch ein zusätzliches Konstantensymbol e beschrieben werden, was aber die nicht-logische Signatur der Sprache erweitert.

- $(G3)$ sagt aus, dass es für jedes Gruppenelement x ein inverses Element $y = \bar{x}$ gibt, so dass $f(x,y) \equiv e \wedge f(y,x) \equiv e$. In diesem Fall kann man y nicht durch ein Konstantensymbol beschreiben, wohl aber durch eine einstellige Funktion g, deren Funktionswert für jedes x das inverse Element $g(x) = \bar{x}$ ist. Auch hier kann also der Existenzquantor durch ein zusätzliches Funktionssymbol eliminiert werden.

Definition: Eine Formel in **BPF** ist in *Skolemform* (oder eine *Skolemformel*), wenn sie keine Existenzquantoren enthält.

Die Verallgemeinerung der obigen Betrachtungen für die Gruppenaxiome auf eine beliebige **BPF**-Formel F führt zu einem Algorithmus, der für F eine passende Skolemformel konstruiert. Der Algorithmus besteht im Wesentlichen nur aus einer while-Schleife, die dann abgebrochen wird, wenn F keine Existenzquantoren enthält. So lange Existenzquantoren auftreten wird jeweils der am weitesten links stehende wie folgt eliminiert:

1. Habe F die Form $\forall y_1 \forall y_2 \ldots \forall y_n \exists z\, G$ (wobei die Anzahl n der Allquantoren vor dem zu elimierenden Existenzquantor auch 0 sein kann).

2. Wir fügen neues n-stelliges Funktionsymbol f in die nichtlogische Signatur ein (0-stellige Funktionssymbole sind Konstantensymbole).

3. Der Existenzquantor wird durch die folgende Substitution der Variable z eliminiert:
$$F := \forall y_1 \forall y_2 \ldots \forall y_n\, G[z/f(y_1,y_2,\ldots,y_n)]$$

Im Allgemeinen wird die ursprüngliche Formel F nicht äquivalent zu der durch den Algorithmus konstruierten Skolemformel sein, denn die Auswertung der Skolemformel hängt wesentlich von der Realisierung der zusätzlichen Funktionen f in der erweiterten Struktur ab. Wenn F aber erfüllbar ist (also ein Modell $\mathcal{M}$ mit $\mathcal{M} \models F$ existiert), dann gibt es auch eine erfüllende Realisierungen der zusätzlichen Funktionen in der Skolemformel, nämlich eine solche, die für $f(\omega(y_1),\omega(y_2),\ldots,\omega(y_n))$ eine passende Belegung der Variable z vom Existenzquantor zurück gibt. Das ist die Grundidee für den Beweis des folgenden, abschließenden Satzes.

Satz: Eine Formel F in **BPF** ist genau dann erfüllbar, wenn die durch den obigen Algorithmus zugeordnete Skolemformel erfüllbar ist.