Diskrete Mathematik für Informatiker und Informations- und Medientechniker Vorlesung WS 2002/2003

zum großen Teil nach Matoušek, Nešetřil, „Diskrete Mathematik“

Winfried Hochstättler
BTU Cottbus

7. Februar 2005

Inhaltsverzeichnis

Kapitel 1

Notation und Grundstrukturen

1.1 Gliederung und Motivation

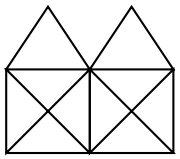
Mit dieser Vorlesung wollen wir Ihnen einerseits Rüstzeug für viele Fragestellungen in der (theoretischen) Informatik in die Hand geben. Andererseits wollen wir Ihnen formale und korrekte logische Schlussweisen näherbringen. Diese Denkweise wird Ihnen bei der Analyse von Programm- oder Netzwerkstrukturen wiederbegegnen.

Nachdem wir in diesem Kapitel einige abstrakte Grundstrukturen kennengelernt haben, werden wir uns zunächst mit Zählproblemen beschäftigen. Dort werden Sie z.B. lernen, folgendes Problem zu lösen:

Problem 1.1.1 *Wie groß ist die Chance mit einem Lotto-Tip fünf Richtige mit Zusatzzahl zu bekommen?*

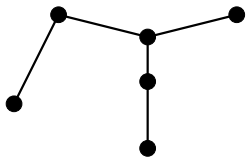
Im dritten Kapitel werden wir Graphen kennenlernen und das Haus vom Nikolaus ohne abzusetzen zeichnen. Ferner werden wir ein Kriterium kennenlernen, das es uns erlaubt, auch das Doppelhaus vom Nikolaus zu betrachten.

Problem 1.1.2 *Kann man nebenstehende Figur ohne abzusetzen zeichnen?*



Im vierten Kapitel lernen wir Bäume kennen und lösen algorithmisch effizient folgendes Problem.

Problem 1.1.3 *Gegeben sind n Stationen und Kosten für eine paarweise Verbindung von je zwei Stationen. Installiere möglichst kostengünstig Verbindungen so, dass jede Station von jeder anderen Station aus (evtl. über Zwischenstationen) erreichbar ist.*



In den darauffolgenden Kapiteln intensivieren wir unsere Betonung der methodische Vorgehensweise und lernen verschiedene Beweise für den gleichen Satz kennen.

Im zweiten Teil der Vorlesung wenden wir uns der mathematischen Logik zu. Hier werden mathematische Sprache und Beweise so stark formalisiert, dass man sie mechanisch behandeln kann. Wir werden dies exemplarisch an der Aussagenlogik durchexerzieren und die Grenzen dieser Vorgehensweise diskutieren.

1.2 Notation

Zunächst wiederholen wir Symbole aus der Mengenlehre, die aus der Schule bekannt sein sollten:

Wir bezeichnen mit

$\mathbb{N}$ die Menge der *natürlichen Zahlen* $\mathbb{N} = \{0, 1, 2, 3, 4, 5, 6, \ldots\}$.

$\mathbb{Z}$ die Menge der *ganzen Zahlen* $\mathbb{Z} = \{\ldots, -4, -3, -2, -1, 0, 1, 2, 3, 4, \ldots\}$.

$\mathbb{Q}$ die Menge der *rationalen Zahlen* $\mathbb{Q} = \{\frac{p}{q} \mid p \in \mathbb{Z}, q \in \mathbb{N} \setminus \{0\}\}$.

$\mathbb{R}$ die Menge der *reellen Zahlen*, dies sind alle Zahlen, die sich als nicht nowendig abbrechende Dezimalbrüche darstellen lassen. Dazu gehören *irrationale, aglebraische Zahlen* wie etwa $\sqrt{2}$, das die Nullstelle von $x^2 - 2$ ist, aber auch *irrationale, transzendente Zahlen*, die nicht Nullstelle eines Polynoms mit rationalen Koeffizienten sind, wie etwa π. Anstatt eines Dezimalkommas, benutzen wir die internationale Schreibweise mit Dezimalpunkt.

Die meisten Operationen, die wir mit Zahlen durchführen, wie Summe, Produkt, Differenz, Quotient, Potenz etc. setzen wir als allgemein bekannt voraus. Ist $x \in \mathbb{R}$, so bezeichnen wir mit

$\lfloor x \rfloor$ den ganzzahligen Teil von x, genauer die nächstkleinere ganze Zahl, also etwa $\lfloor 1.99 \rfloor = 1$, $\lfloor 2.01 \rfloor = 2$, $\lfloor 2 \rfloor = 2$, $\lfloor -1.99 \rfloor = -2$.

$\lceil x \rceil$ ist die nächstgrößere ganze Zahl, also etwa $\lceil 1.99 \rceil = 2$, $\lceil 2.01 \rceil = 3$, $\lceil 2 \rceil = 2$, $\lfloor -1.99 \rfloor = -1$.

Summen und Produkte mehrerer Elemente kürzen wir mit dem *Summationszeichen* Σ und dem Produktzeichen Π ab.

$$\sum_{i=1}^{n} a_i := a_1 + a_2 + \ldots + a_n, \quad \prod_{i=1}^{n} a_i := a_1 \cdot a_2 \cdot \ldots \cdot a_n.$$

Also zum Beispiel $\sum_{i=1}^{5} i^2 = 1 + 4 + 9 + 16 + 25 = 55$.

Die leere Summe ist 0 und das leere Produkt 1, also z.B. $\sum_{i=1}^{0} 3 = 0$, $\prod_{i=1}^{0} 3 = 1$.

Allgemeine Mengen bezeichnen wir meist mit Großbuchstaben. Wenn x in M liegt, schreiben wir $x \in M$, ansonsten $x \notin M$.

Sind M, N zwei Mengen, so ist M eine Teilmenge von N, in Zeichen $M \subseteq N$, wenn $x \in M \Rightarrow x \in N$, in Worten, wenn x in M liegt, so liegt es auch in N. Zwei Mengen M, N sind gleich $M = N$, wenn $M \subseteq N$ und $N \subseteq M$ ist.

Vereinigung und *Schnitt* von Mengen sind definiert als

$$M \cup N := \{x \mid x \in M \text{ oder } x \in N\}, \qquad M \cap N := \{x \mid x \in M \text{ und } x \in N\}.$$

Die *Differenzmenge* $M \setminus N$ ist definiert als $M \setminus N := \{x \in M \mid x \notin N\}$.

Das *Cartesische Produkt* zweier Mengen M und N, symbolisch $M \times N$, ist erklärt als die Menge der geordneten Paare (x, y) mit $x \in M$ und $y \in N$. Wir betrachten nun zwei Spezialfälle von Teilmengen des Cartesischen Produktes.

1.3 Abbildungen

Eine Abbildung ordnet jedem Element aus einer *Urbildmenge* ein Element aus der *Bildmenge* zu. Formal:

Definition 1.3.1 *Eine* Abbildung $f : M \to N$ *aus einer Menge M in eine Menge N ist eine Menge von geordneten Paaren $(x, y) \in M \times N$ mit der Eigenschaft, dass es für jedes $x \in M$ genau ein Tupel in dieser Menge gibt, das x in der ersten Komponente hat. Wir schreiben dann auch $x \mapsto y$.*

Statt $(x, y) \in f$ schreiben wir üblicherweise $f(x) = y$. Ist $A \subseteq M$, so bezeichnen wir mit $f(A) := \{f(a) \mid a \in A\} \subseteq N$ die Menge aller Bilder von Elementen in A.

Definition 1.3.2 *Sind $f : M \to N$ und $g : Y \to M$ Abbildungen, so definieren wir die* Komposition *oder* Hintereinanderausführung *der Abbildungen $h := f \circ g$ durch $h(x) = f(g(x))$.*

Man überzeugt sich, dass h wieder eine Abbildung ist.

Eine Abbildung $f : M \to N$ heißt

injektiv, wenn verschiedene Urbilder verschieden Bilder haben, also $x \neq y \Rightarrow f(x) \neq f(y)$,

surjektiv, wenn jedes Element in der Bildmenge getroffen wird, also $f(M) = N$,

bijektiv, wenn sie injektiv und surjektiv ist.

Definition 1.3.3 *Zwei Mengen A, B heißen* gleichmächtig*, wenn es eine bijektive Abbildung $f : A \to B$ gibt.*

Proposition 1.3.4 *a) Die Hintereinanderausführung injektiver Abbildungen ist injektiv.*

b) Die Hintereinanderausführung surjektiver Abbildungen ist surjektiv.

c) Die Hintereinanderausführung bijektiver Abbildungen ist bijektiv.

Beweis.

a) Seien also f, g zwei injektive Abbildungen und der Bildbereich von g sei identisch mit dem Definitionsbereich (Urbildbereich) von f. Wir haben zu zeigen, dass $x \neq y \Rightarrow f \circ g(x) \neq f \circ g(y)$. Seien also $x \neq y$ zwei verschieden Elemente aus dem Definitionsbereich von g. Da g injektiv ist, sind $g(x) \neq g(y)$ zwei verschiedene Elemente aus dem Definitionsbereich von f. Da f injkektiv ist, folgt nun $f \circ g(x) = f(g(x)) \neq f(g(y)) = f \circ g(y)$.

b) Hier müssen wir zeigen, dass jedes Element aus dem Bildbereich N von $f \circ g$ als Bild angenommen wird. Sei also x ein solches Element. Da f surjektiv ist, gibt es ein y aus dem Definitionsbereich von f mit $f(y) = x$, analog gibt es ein z mit $g(z) = y$. Also ist $f \circ g(z) = x$.

c) Dies folgt aus den beiden vorhergehenden Aussagen.

□

Bei einer bijektiven Abbildung $f : M \to N$ hat jedes Element in der Zielmenge ein Urbild und dieses ist eindeutig. Also können wir die *Umkehrabbildung* $g : N \to M$ definieren als $g(y) = x \Leftrightarrow f(x) = y$. Wir bezeichnen ein solches g auch mit f^{-1}.

Ist $f : M \to N$ eine Abbildung und $L \subseteq M$, so bezeichnen wir mit $f_{|L} : L \to N$ die *Einschränkung von f auf L.* Damit können wir auch zwei Abbildungen verketten, wenn der Bildbereich von der ersten nur eine Teilmenge des Definitionsbereichs der zweiten Funktion ist.

1.4 Relationen

Eine beliebige Teilmenge $R \subseteq M \times N$ des Cartesischen Produktes nennen wir eine Relation. Ist $(x, y) \in R$, so sagen wir auch x steht in Relation mit y.

Zu jedem x und y können wir bzgl. R die Mengen aussondern.

$$[x]_l := \{y \in N \mid (x, y) \in R\}, \qquad [y]_r := \{x \in M \mid (x, y) \in R\}.$$

Den Index lassen wir weg, wenn die Interpretation eindeutig ist

Definition 1.4.1 *Sind M, N, L Mengen und $R \subseteq M \times N$ sowie $S \subseteq N \times L$ Relationen, so erklären wir die* Komposition $R \circ S \subseteq M \times L$ *durch $(x, z) \in R \circ S \Leftrightarrow$ es gibt $y \in N$, so dass $(x, y) \in R$ und $(y, z) \in S$.*

Vorsicht! Auch Abbildungen kann man als Relationen auffassen. Bei der Verknüpfung ist allerdings (aus historischen Gründen) die Notation vertauscht, also $f \circ g$ müsste als Komposition von Relationen aufgefasst, als $g \circ f$ geschrieben werden.

1.5 Äquivalenzrelationen

Wir betrachten in diesem Abschnitt *Relationen auf einer Menge M*, d.h. $R \subseteq M \times M$.

Definition 1.5.1 *Sei R eine Relation auf einer Menge M. Die Relation ist*

reflexiv, *wenn für alle $x \in M : (x, x) \in R$.*

symmetrisch, *wenn $(x, y) \in R \Rightarrow (y, x) \in R$.*

transitiv, *wenn $((x, y) \in R$ und $(y, z) \in R) \Rightarrow (x, z) \in R$.*

Eine reflexive, symmetrische und transitive Relation nennen wir Äquivalenzrelation.

Äquivalenzrelationen definieren so etwas ähnliches wie Gleichheit. Sie zerlegen die Grundmenge in paarweise disjunkte *Äquivalenzklassen*, die Mengen $[x]$.

Proposition 1.5.2 *Sei R eine Äquivalenzrelation auf M. Dann gilt*

a) $[x] \neq \emptyset$ *für alle $x \in M$.*

b) Für je zwei $x, y \in M$ ist entweder $[x] = [y]$ oder $[x] \cap [y] = \emptyset$. Also bilden die Äquivalenzklassen eine Partition von M.

c) R ist durch ihre Äquivalenzklassen vollständig bestimmt.

Beweis.

a) Da R reflexiv ist, gilt stets $x \in [x]$.

b) Wir zeigen $[x] \cap [y] \neq \emptyset \Rightarrow [x] = [y]$. Seien also $x, y \in M$ und $z \in [x] \cap [y]$. Wir zeigen $[x] \subseteq [y]$. Sei dazu $t \in [x]$. Dann sind zunächst $(x, z), (y, z), (x, t) \in R$ also schließen wir mit Symmetrie und Transitivität, $(t, x), (t, z), (z, y), (t, y), (y, t) \in R$. Also auch $t \in [y]$ und somit $[x] \subseteq [y]$. Die Symmetrie in x und y liefert $[y] \subseteq [x]$.

c) Offensichtlich gilt $(x, y) \in R \Leftrightarrow \{x, y\} \subseteq [x]$.

1.6 Partialordnungen

Definition 1.6.1 *Sei R eine Relation auf einer Menge M. Die Relation ist*

antisymmetrisch, *wenn $((x,y) \in R$ und $(y,x) \in R) \Rightarrow x = y$.*

Eine reflexive, antisymmetrische und transitive Relation heißt Partialordnung.

Ist R eine Partialordnung und $(x,y) \in R$, so schreiben wir auch $x \leq y$. Oft nennen wir die Grundmenge P und notieren die Relation als $(P, \leq)$. Stehen je zwei Elemente in Relation, so sprechen wir von einer *linearen Ordung*, einer *totalen Ordnung* oder einfach von einer *Ordnung*.

Beispiel 1.6.2 *Die bekannten Ordnungen auf $(\mathbb{N}, \leq)$ und $(\mathbb{R}, \leq)$ sind Totalordnungen. Die Inklusionsbeziehung (Teilmengenbeziehung) auf der Potenzmenge 2^M, das ist die Menge aller Teilmengen von M, einer Menge M ist eine Partialordnung.*

Sei $(P, \leq)$ eine Partialordnung und $a \leq b \in P$. Ist $a \neq b$, so schreiben wir $a < b$. Wir sagen b *bedeckt* a, in Zeichen $a \lessdot b$, wenn $a < b$ und $a \leq c \leq b \Rightarrow c \in \{a, b\}$. Endliche Partialordnungen werden durch die Bedeckungsrelationen erzeugt:

Proposition 1.6.3 *Sei $(P, \leq)$ eine endliche Partialordnung und $x, y \in P$. Dann gilt $x < y$ genau dann, wenn es $0 \leq k$ Elemente $x_1, \ldots, x_k$ gibt mit $x \lessdot x_1 \lessdot \ldots \lessdot x_k \lessdot y$.*

Beweis. Gilt $x \lessdot x_1 \lessdot \ldots \lessdot x_k \lessdot y$, so folgt aus der Transitivität $x \leq y$ und, falls $k > 0$, impliziert $x_1 \leq y$, nun $x \neq y$ und damit $x < y$. Die andere Implikation zeigen wir mittels Induktion über die Anzahl n der Elemente $t \in P$ mit $x < t < y$. Ist $n = 0$, so ist nichts zu zeigen. Ist $n \geq 1$, so wählen wir ein festes z mit $x < z < y$. Dann gibt es sowohl zwischen x und z, als auch zwischen z und y weniger als n Elemente. Also gibt es $x \lessdot x_1 \lessdot \ldots \lessdot x_l \lessdot z := x_{l+1}$ und $x_{l+1} \lessdot x_{l+2} \lessdot \ldots \lessdot x_k \lessdot y$. □

Es genügt also zur Beschreibung einer endlichen Partialordnung, nur die Bedeckungsrelationen zu betrachten. Diese werden oft graphisch als Hasse-*Diagramm* dargestellt, wobei die Elemente als Punkte und die Relationen als Verbindungen vom kleineren unteren Element zum größeren oberen Element dargestellt werden.

Beispiel 1.6.4 *In Abbildung 1.1 sehen wir links das* Hasse*-Diagramm der Teilbarkeitsrelation $a \leq b \Leftrightarrow a$ teilt b auf der Menge $\{1, 2, \ldots, 11, 12\}$ und rechts das der Teilmengenrelation der Potenzmenge von $\{1, 2, 3, 4\}$.*

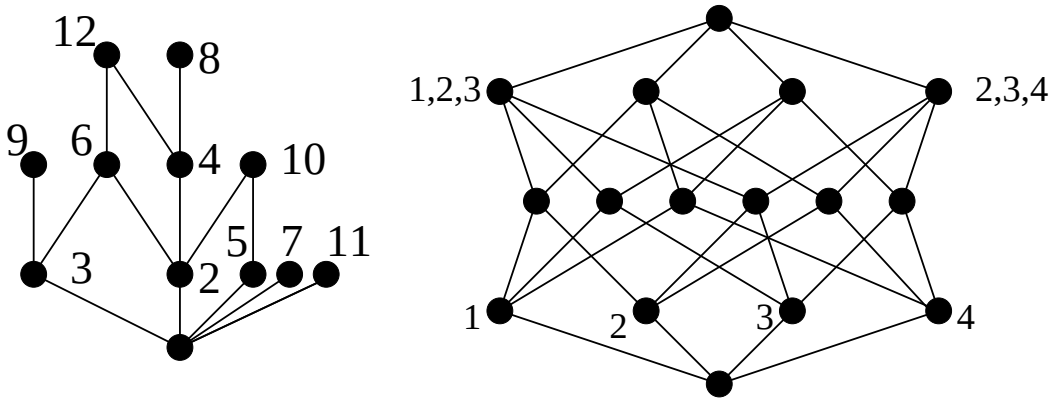


Abbildung 1.1: Zwei Hasse-Diagramme

1.7 Beweismethoden und das Prinzip der vollständigen Induktion

Wie Sie in den bisherigen Abschnitten bereits gesehen haben, besteht ein mathematischer Text zumeist aus Definitionen, Sätzen und Beweisen. Dabei ist eine *Definition* eine sprachliche Vereinbarung, die jeweils einer gewissen Struktur einen Namen gibt. Ein *Satz* besteht zumeist aus einigen Voraussetzungen und einer Behauptung. In dem *Beweis* wird schlüssig Schritt für Schritt dargelegt, warum unter Annahme der Gültigkeit der Voraussetzungen die Behauptung notwendig auch gelten muss. Jeder einzelne Schritt des Beweises sollte logisch nachvollziehbar sein.

Neben solchen direkten Beweisen, wollen wir hier noch drei weitere Vorgehensweisen vorstellen. Zunächst den

1.7.1 Beweis durch Kontraposition

Betrachten wir hierzu den „Satz“:

> Wer einkaufen geht und bar bezahlt hat danach weniger Geld in der Brieftasche.

Diese Aussage hat die Form

$$A \text{ und } B \Rightarrow C.$$

Logisch gleichwertig ist die umgekehrte Implikation der Negationen, nämlich

$$\text{nicht } C \Rightarrow \text{ nicht } (A \text{ und } B),$$

wobei die rechte Seite der Implikation wiederum gleichwertig ist mit

$$\text{nicht } A \text{ oder nicht } B.$$

Insgesamt können wir statt obiger Aussage also genau so gut zeigen:

> Wer danach nicht weniger Geld in der Brieftasche hat war nicht einkaufen oder hat nicht bar bezahlt.

Ähnlich hatten wir beim Beweis, dass eine Äquivalenzrelation eine Menge partitioniert eine logische Äquivalenz ausgenutzt, nämlich anstatt

$$[x] \cap [y] = \emptyset \text{ oder } [x] = [y]$$

hatten wir die logisch äquivalente Aussage

$$[x] \cap [y] \neq \emptyset \Rightarrow [x] = [y]$$

bewiesen.

Wir halten also fest

$$(A \Rightarrow B) \Leftrightarrow (B \text{ oder nicht } A).$$

1.7.2 Widerspruchsbeweis oder reductio ad absurdum

Hier wird eine Behauptung dadurch bewiesen, dass man zeigt, dass die Verneinung der Behauptung etwas Unsinniges impliziert. Betrachten wir hier als Beispiel die Aussage:

> Es gibt unendlich viele Primzahlen $2, 3, 5, 7, 11, 13, 17, 19, 23, \ldots$.

Nehmen wir das Gegenteil an und sei etwa $\{p_1, \ldots, p_n\}$ die endliche Menge der Primzahlen und sei $P = \prod_{i=1}^{n} p_i$. Dann ist $P + 1$ keine Primzahl, wird also von einer Primzahl, etwa p_{i_0} echt geteilt. Also ist

$$\frac{P+1}{p_{i_0}} = \prod_{\substack{i=1 \\ i \neq i_0}}^{n} p_i + \frac{1}{p_{i_0}}$$

eine natürliche Zahl, was offensichtlich Unsinn ist. Also muss obige Aussage richtig sein.

1.7.3 Das Prinzip der vollständigen Induktion

Oft will man Aussagen für endliche Mengen beweisen. Dafür kann man den konstruktiven Aufbau der natürlichen Zahlen ausnutzen. Diese lassen sich nämlich durch einen Anfang, die 0, und eine Nachfolgerfunktion beschreiben. Wenn eine Teilmenge von ganzen Zahlen die 0 und mit jeder Zahl auch ihren Nachfolger enthält, dann enthält sie alle natürlichen Zahlen. (Sie enthält die 0, also die 1, also die 2, also ...). Folglich kann man eine Aussage für eine Zahl $n_0 \in \mathbb{Z}$ beweisen und zeigen, dass sie, wenn sie für eine Zahl n gilt, dann auch für ihren Nachfolger $n + 1$. Diese beiden Fakten zusammengenommen beweisen dann, dass die Aussage für alle ganzen Zahlen, die größer oder gleich n_0 sind, gilt.

Wir betrachten als Beispiel ein Arrangement aus endlich vielen, paarweise nicht parallelen, Geraden in der Ebene, von denen keine drei einen Punkt gemeinsam haben. Dabei habe eine Gerade kein Ende und keinen Anfang. Ein solches Arrangement unterteilt die Ebene in verschiedene Flächen. Diese können nach unendlich offen sein oder sie bilden ein konvexes Polygon.

> Wir behaupten, dass ein solches Arrangement mit mindestens 3 Geraden stets ein Dreieck unter den konvexen Polygonen hat.

Diese Aussage ist offensichtlich richtig, wenn das Arrangement nur aus drei Geraden besteht. Denn zwei Geraden, die nicht parallel sind, schneiden sich in einem Punkt, etwa p. Sei dann q der Punkt auf der dritten Geraden mit dem geringsten Abstand von p. Dann bildet pq die Höhe eines Dreiecks.

Sei nun ein Geradenarrangement mit $n \geq 4$ Geraden gegeben. Wir zeigen:

> Wenn die Aussage für $n - 1$ Geraden richtig ist, so ist sie auch für n Geraden richtig.

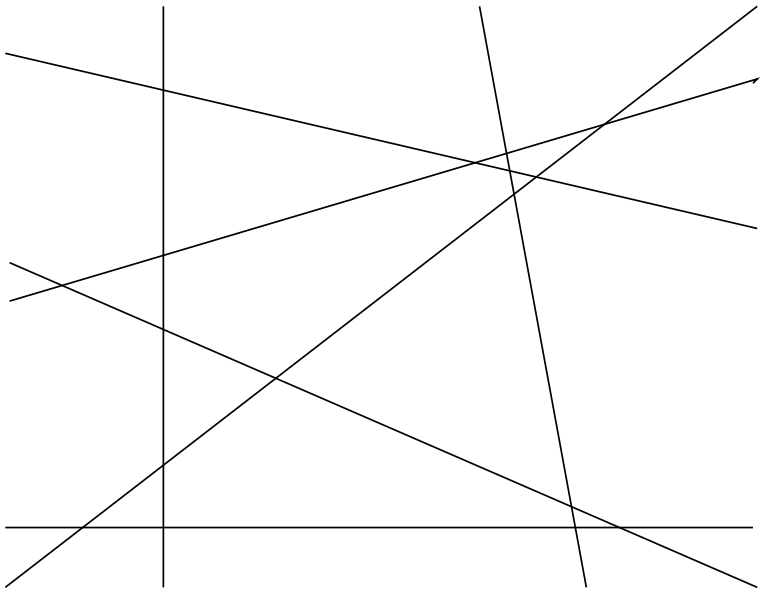


Abbildung 1.2: Ein Geradenarrangement

Wir entfernen dafür eine Gerade g und nehmen an, dass es in dem verbleibenden Arrangement von $n-1$ Geraden ein Dreieck gibt. Nun nehmen wir die Gerade g wieder dazu und unterscheiden zwei Fälle.

a) Die Gerade g schneidet das Dreieck nicht. Offensichtlich bleibt das Dreieck dann erhalten.

b) Die Gerade g schneidet das Dreieck in zwei Kanten. Dann wird das Dreieck in ein Dreieck und ein Viereck zerlegt.

In jedem Fall enthält unser Arrangement wieder ein Dreieck. Also erhalten wir:

Weil die Aussage für 3 Geraden richtig ist, ist sie für 4 richtig, ist sie für 5 richtig usf. Also ist sie für allen Geradenarrangements richtig. □

Als Zeichen, dass der Beweis fertig ist haben wir rechts ein offenes Quadrat gesetzt

In Proposition 1.6.3 hatten wir nicht nur vom individuellen Vorgänger sondern von allen Vorgängern einer Zahl auf diese geschlossen, also gezeigt:

> Wenn die Aussage für $\{1, \ldots, n-1\}$ wahr ist, so ist sie auch für n wahr.

Dort hatten wir nämlich geschlossen, dass zwischen x und z und auch zwischen z und y weniger als n Elemente liegen. Diese Elementanzahlen sind also Zahlen in $\{1, \ldots, n-1\}$. Offensichtlich geht das Induktionsprinzip auch dafür durch.

Kapitel 2

Elementare Abzählprobleme

In diesem Abschnitt betrachten wir Zählprobleme wie etwa

- Auf wieviele Arten kann ich m Postkarten an n Freunde verschicken?
- Wieviel Tischordnungen sind möglich?
- Wieviele verschiedene Lotto-Tips sind möglich?

2.1 Abbildungen und Mengen

Wir beginnen mit einem Beispiel.

Beispiel 2.1.1 *An unserem Urlaubsort entdecken wir nach intensiver Suche 7 Ansichtskarten, die unseren Ansprüchen genügen und jeweils in hinreichend großer Menge verfügbar sind. Auf wieviele Arten können wir die 12 Onkel, Tanten und Freunde damit beglücken?*

Für den ersten Empfänger haben wir 7 Postkarten zur Auswahl, für den zweiten wieder 7. Die Auswahlen hängen nicht voneinander ab, also ergeben sich insgesamt 49 Möglichkeiten. Iterieren wir diese Argumentation ergeben sich 7^{12} *Möglichkeiten.*

Abstrakt betrachten wir die Menge aller Abbildungen von einer n-elementigen Menge F (von Freunden) in eine m-elementige Menge P (von Postkarten).

Man klassifiziert die unterschiedlichen Objekte, die wir in diesem und dem folgenden Abschnitt beschreiben auch häufig als *Urnenexperimente*. Wir wollen hier die Möglichkeiten zählen, eine Sequenz von nummerierten Kugeln aus

einer Urne zu ziehen, wobei wir gezogene Kugeln wieder zurücklegen und sie uns merken. Man spricht auch von einer *Variation mit Wiederholung.*

Proposition 2.1.2 *Seien $n, m \in \mathbb{N}, m \geq 1$ und F eine n-elementige Menge und P eine m-elementige Menge. Dann ist die Anzahl aller Abbildungen $f : F \to P$ gerade m^n.*

Beweis. Wir führen Induktion über n. Die Anzahl der Abbildungen von der leeren Menge F nach P ist gerade $1 = m^0$. Sei also $n \geq 1$ und $g \in F$ fest gewählt. Nach Induktionsvoraussetzung gibt es m^{n-1} Abbildungen von $F \setminus \{g\}$ nach P. Außerdem gibt es m Abbildungen von $\{g\}$ nach P. Nun können wir jede Abbildung $f : F \to P$ zerlegen in zwei Abbildungen $f_1 : F \setminus \{g\} \to P$ und $f_2 : \{g\} \to P$. Umgekehrt definiert jedes solche Paar (f_1, f_2) ein $f : F \to P$ und diese sind für verschiedene Tupel verschieden. Also gibt es davon $m * m^{n-1} = m^n$ Stück. □

Als Konsequenz erhalten wir

Korollar 2.1.3 *Sei X eine n-elementige Menge. Dann hat X genau 2^n Teilmengen.*

Beweis. Zu einer gegebenen Teilmenge $Y \subseteq X$ definieren wir die *charakteristische Funktion χ_Y von Y* als $\chi_Y : X \to \{0, 1\}$

$$\chi_Y(x) := \begin{cases} 1 & \text{falls} \quad x \in Y \\ 0 & \text{sonst.} \end{cases} \tag{2.1}$$

Offensichtlich sind die charakteristischen Funktionen verschiedener Teilmengen voneinander verschieden. Umgekehrt erhält man aber jede Funktion $f : X \to \{0, 1\}$ als charakteristische Funktion einer Teilmenge. Also ist die Anzahl der Teilmengen von X gerade gleich der Anzahl der Abbildungen $f : X \to \{0, 1\}$ also 2^n nach Proposition 2.1.2. □

Die Hälfte dieser Teilmengen ist gerade und die andere ungerade:

Proposition 2.1.4 *Sei $n \geq 1$. Jede n-elementige Menge hat genau 2^{n-1} Teilmengen mit ungerade vielen Elementen und ebenso viele mit gerade vielen Elementen.*

Beweis. Sei X eine n-elementige Teilmenge und $a \in X$ ein festes Element. Dann hat $X \setminus \{a\}$ nach Korollar 2.1.3 2^{n-1} Teilmengen. Jede solche Teilmenge T hat entweder ungerade viele Elemente oder dies gilt für $T \cup \{a\}$. Umgekehrt enthält jede ungerade Teilmenge S entweder das Element a nicht, oder $S \setminus \{a\}$ ist gerade. Also hat X genau 2^{n-1} ungerade Teilmenge und $2^n - 2^{n-1} = 2^{n-1}$ gerade Teilmengen. □

2.2 Injektive Abbildungen, Permutationen und Fakultät

Das Ende des Urlaubs naht, noch keine Postkarte ist geschrieben, außerdem widerstrebt es uns auf einmal, zweimal die gleiche Postkarte zu versenden. Pragmatisch reduzieren wir die Anzahl der Empfänger auf 5. Auf wieviel Arten können wir unsere 5 Freunde mit den sieben Postkarten beglücken?

In diesem Falle zählen wir also die injektiven Abbildungen in eine endliche Menge.

Das zugehörige Urnenexperiment lautet wie oben aber ohne Zurücklegen. Wir sprechen von einer *Variation ohne Wiederholung*.

Proposition 2.2.1 *Seien $m, n \in \mathbb{N}$. Dann gibt es genau*

$$m(m-1)\dots(m-n+1) = \prod_{i=0}^{n-1}(m-i) \tag{2.2}$$

injektive Abbildungen einer gegebenen n-elementigen Menge A in eine gegebene m-elementige Menge B.

Beweis. Wir führen wieder Induktion über n. Ist $n = 0$, so gibt es genau eine solche Abbildung. Das leere Produkt ist per definitionem 1. Sei also nun $n > 0$ und $a \in A$ ein festes Element. Es gibt m mögliche Bilder $f(a) \in B$ für a. Jede dieser Möglichkeiten wird durch jede injektive Abbildung von $A \setminus \{a\}$ nach $B \setminus \{f(b)\}$ zu einer injektiven Abbildung von A nach B ergänzt. Von letzteren gibt es nach Induktionsvoraussetzung aber genau $\prod_{i=0}^{n-2}(m-1-i) = \prod_{i=1}^{n-1}(m-i)$ Stück, woraus die Behauptung folgt. □

Eine bijektive Abbildung $\sigma : M \to M$ einer Menge in sich selbst hatten wir *Permutation der Menge M* genannt. Ist $|M|$ endlich, so gibt es nach Proposition 2.2.1 $n(n-1)\cdot\ldots 2\cdot 1$ Permutationen. Diese Zahl nennen wir Fakultät.

Definition 2.2.2 *Sei $n \in \mathbb{N}$. Die Zahl*

$$n! := \prod_{i=1}^{n} i = 1\cdot 2\cdot \ldots \cdot (n-1)n \tag{2.3}$$

nennen wir die Fakultät von n.

Durch Abzählen der Elemente können wir jede Permutation einer endlichen Menge der Kardinalität n als Permutation von $N := \{1, 2, \ldots, n\}$ auffassen. Manchmal ist es nützlich, eine Permuation als lineare Anordnung von N zu betrachten. Wir wollen noch ihre Zerlegung in *Zyklen* diskutieren.

Ein Zyklus (Zykel) $(a_1 a_2 \ldots a_k)$ ist eine wiederholungsfreie Folge von Zahlen in N mit $\sigma(a_k) = a_1$ und $\sigma(a_i) = a_{i+1}$ für $i = 1, \ldots, k-1$. Jeder Zyklus ist selbst eine Permutation, die alle Elemente, die nicht vorkommen fest lässt. Wir können nun mit folgendem Algorithmus σ in disjunkte Zyklen zerlegen:

```
for a in N:
   print
   b=a
   print b,
   while σ[b] !=a:
      b=σ[b]
      print b,
      N.remove(b)
```

Man wählt ein noch nicht erledigtes Element, verfolgt sein Bild unter iterierter Anwendung von σ bis es wiederkehrt. Ein Zyklus wurde gefunden und aus der Grundmenge entfernt. Dies iteriert man, bis alle Elemente abgearbeitet sind.

Beispiel 2.2.3 *Wir zerlegen*

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 4 & 7 & 6 & 1 & 2 & 3 & 5 \end{pmatrix}$$

in die Zyklen (14)(275)(36)*.*

2.3 Binomialkoeffizienten

Definition 2.3.1 *Seien $n, k \in \mathbb{N}, n \geq k$. Der* Binomialkoeffizient n über k *ist definiert vermöge*

$$\binom{n}{k} := \frac{n(n-1)(n-2)\cdot \ldots \cdot (n-k+1)}{1 \cdots 2 \cdot \ldots \cdot (k-1)k} = \frac{\prod_{i=0}^{k-1}(n-i)}{k!}. \qquad (2.4)$$

Diese Definition hat gegenüber der verbreiteten Formel $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ den Vorteil, dass sie sich auf den Fall $n \in \mathbb{R}$ verallgemeinern lässt. Insbesondere wollen wir zulassen, dass $k > n$ ist mit $k \in \mathbb{N}$. In diesem Falle ist $\binom{n}{k} = 0$.

Die Zahl n über k gibt nun die Anzahl der Möglichkeiten an, aus einer n-elementigen Menge eine k-elementige Teilmenge auszuwählen.

Beim zugehörigen Urnenexperiment ziehen wir Kugeln ohne Zurücklegen und ignorieren im Ergebnis die Reihenfolge der gezogenen Zahlen. Wir sprechen von einer *Kombination ohne Wiederholung.*

Bevor wir dies beweisen, definieren wir:

Definition 2.3.2 *Sei X eine Menge und $k \in \mathbb{N}$. Dann bezeichne das Symbol*

$$\binom{X}{k}$$

die Menge aller k-elementigen Teilmengen von X.

Proposition 2.3.3 *Sei X eine endliche Menge und $k \in \mathbb{N}$. Dann hat X genau $\binom{|X|}{k}$ k-elementige Teilmengen oder als Formel*

$$\left|\binom{X}{k}\right| = \binom{|X|}{k}. \tag{2.5}$$

Beweis. Offensichtlich ist die Behauptung richtig für $k > |X|$, sei also $k \leq |X|$. Wir betrachten die k-elementigen Teilmengen als Bildmengen $f(\{1, \ldots, k\})$ injektiver Abbildungen von $f : \{1, \ldots k\} \to X$. Davon gibt es zunächst nach Proposition 2.2.1 $\frac{n!}{(n-k)!}$. Ist nun σ eine Permutation von $\{1, \ldots, k\}$, so ist $f \circ \sigma$ eine echte weitere injektive Abbildung mit gleicher Bildmenge. Ist umgekehrt $g : \{1, \ldots k\} \to X$ eine injektive Abbildung, so definiert $\sigma(j) = f^{-1}(g(j))$ eine Permutation $\sigma : \{1, \ldots, k\} \to \{1, \ldots, k\}$. Also haben wir oben jede k elementige Menge genau $k!$ mal gezählt, woraus die Behauptung folgt. □

Beispiel 2.3.4 *Sei $X = \{1, 2, \ldots, 49\}$ und $k = 6$. Dann gibt es $\binom{49}{6} = 13,983,816$ mögliche Lottotips.*

Mit Hilfe der Binomialkoeffizienten können wir auch die Anzahl der Partitionen einer natürlichen Zahl in k Summanden zählen, also z.B. kann man 4 schreiben als $0 + 4$, $1 + 3$, $2 + 2$, $3 + 1$ und $4 + 0$, also gibt es 5 Partitionen

von 4 in 2 Summanden. Zur Bestimmung dieser Zahl betrachten wir zunächst eine feste Partition

$$n = a_1 + \ldots + a_k$$

und stellen uns vor, dass wir die a_i in *unärer Notation* geschrieben hätten. Zusammen mit den Pluszeichen haben wir dann eine Zeichenkette aus n + k - 1 Zeichen. Betrachten wir also die Pluszeichen als Trennsymbole, so entsprechen die Partitionen eineindeutig den Möglichkeiten $k-1$ Pluszeichen in einer Zeichenkette der Länge $n + k - 1$ zu platzieren. Also haben wir

Korollar 2.3.5 *Die Anzahl der Partitionen einer n-elementigen Menge in k unterschiedliche (nummerierte) Klassen ist*

$$\binom{n+k-1}{k-1}. \tag{2.6}$$

Folgende Eigenschaften von Binomialkoeffizienten sollten Sie kennen

Proposition 2.3.6 *Seien $n, k \in \mathbb{N}$, $n \geq k$. Dann gilt*

a) $\binom{n}{k} = \binom{n}{n-k}$,

b) Seien zusätzlich $n \geq k \geq 1$. Dann ist

$$\binom{n-1}{k-1} + \binom{n-1}{k} = \binom{n}{k}. \tag{2.7}$$

Beweis. Die erste Aussage kann man unmittelbar aus der Formel ablesen oder man stellt fest, dass das Komplement einer k-elementigen Teilmenge in einer n-elementigen Menge eine $n-k$-elementige Menge ist und folglich die Anzahl der $n-k$-elementigen Teilmengen einer n-elementigen Menge gleich der Anzahl der k-elementigen Teilmengen ist.

Bei der zweiten Aussage wählen wir wieder eine n-elementige Menge X und $a \in X$. Dann gibt es $\binom{n-1}{k-1}$ k-elementige Teilmengen von X, die a enthalten und $\binom{n-1}{k}$, die a nicht enthalten. □

Die letzte der beiden Gleichungen führt zur Konstruktion des sogenannten *Pascalschen Dreiecks*.

$$
\begin{array}{ccccccccccc}
 & & & & 1 & & & & & & \\
 & & & 1 & & 1 & & & & & \\
 & & 1 & & 2 & & 1 & & & & \\
 & 1 & & 3 & & 3 & & 1 & & & \\
1 & & 4 & & 6 & & 4 & & 1 & & \\
1 & & 5 & & 10 & & 10 & & 5 & & 1 \\
 & & \vdots & & & & \vdots & & & &
\end{array}
$$

Dabei schreibt man an den linken und rechten Rand des Dreiecks lauter Einsen und ein Eintrag entsteht, indem man die Summe der links und rechts darüberstehenden Zahlen bildet. In der n-ten Zeile stehen dann aufgrund der letzten Proposition und dem Fakt, dass $\binom{n}{n} = \binom{n}{0} = 1$ für beliebiges n ist, gerade die Zahlen $\binom{n}{k}$ für $k = 0, \ldots, n$. □

Der Name der Binomialkoeffizienten kommt von folgendem Zusammenhang

Satz 2.3.7 (Binomischer Satz) *Sei $n \in \mathbb{N}$. Dann ist*

$$(1+x)^n = \sum_{k=0}^{n} \binom{n}{k} x^k. \tag{2.8}$$

Beweis. Wir führen Induktion über n. Die Aussage ist richtig für $n = 0$.

$$
\begin{aligned}
(1+x)^n &= (1+x)(1+x)^{n-1} \\
&\overset{IV}{=} (1+x) \sum_{k=0}^{n-1} \binom{n-1}{k} x^k \\
&= \sum_{k=0}^{n-1} \binom{n-1}{k} x^k + \sum_{k=1}^{n} \binom{n-1}{k-1} x^k \\
&= \binom{n}{0} + \sum_{k=1}^{n-1} \left(\binom{n-1}{k-1} + \binom{n-1}{k-1} \right) x^k + \binom{n-1}{n-1} x^n \\
&\overset{(2.7)}{=} 1 + \sum_{k=1}^{n-1} \binom{n}{k} x^k + x^n.
\end{aligned}
$$

□

Korollar 2.3.8

$$\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{n} = 2^n.$$

Beweis. Diese Gleichung erhalten wir, wenn wir im binomischen Satz $x = 1$ wählen. □

Das letzte Korollar gibt einen alternativen Beweis dafür, dass 2^n die Anzahl der Teilmengen einer n-elementigen Menge ist. Wir können ähnlich die Anzahl der ungeraden Teilmengen herleiten, da

$$\binom{n}{0} - \binom{n}{1} + \binom{n}{2} - \binom{n}{3} + \cdots + (-1)^n \binom{n}{n} = (1-1)^n = 0$$

ist, gibt es genauso viele ungerade wie gerade Teilmengen.

Formelsammlungen sind voll von Gleichungen mit Binomialkoeffizienten. Hier eine weitere:

Proposition 2.3.9

$$\sum_{i=0}^{n} \binom{n}{i}^2 = \binom{2n}{n}.$$

Beweis. Wir betrachten eine $2n$-elementige Menge X. Bei dieser färben wir n Elemente rot und die übrigen blau. Jede n-elementige Teilmenge von X setzt sich dann aus i roten Elementen und $n-i$ blauen Elementen zusammen für ein $i \in \{1, \ldots, n\}$. Umgekehrt ergibt jede Menge aus i roten und $n-i$ blauen Elementen genau eine n-elementige Teilmenge von X. Somit haben wir

$$\binom{2n}{n} = \sum_{i=0}^{n} \binom{n}{i}\binom{n}{n-i} = \sum_{i=0}^{n} \binom{n}{i}^2.$$

□

Zum Ende dieses Abschnitts wollen wir noch eine Verallgemeinerung der Binomialkoeffizienten kennenlernen. Dafür betrachten wir zunächst die Fragestellung, wieviele verschiedene Zeichenketten man aus den Buchstaben des Wortes BANANE bilden kann. Nach dem bisher Gelernten können wir die 6 Buchstaben auf 6! Arten anordnen. Dabei erhalten wir allerdings jedes Wort viermal, da N und A je zweimal vorkommen. Die Anzahl der Möglichkeiten ist also $\frac{6!}{1!2!2!1!} = 180$. Allgemein definieren wir

Definition 2.3.10 *Sei $k_1 + \ldots + k_m = n$. Der* Multinomialkoeffizient *ist definiert als*

$$\binom{n}{k_1, k_2, \ldots, k_m} := \frac{n!}{k_1! k_2! \ldots k_m!}. \tag{2.9}$$

Der Multinomialkoeffizient beschreibt also die Anzahl der Möglichkeiten, n Objekte, von denen jeweils k_i nicht unterscheidbar sind, anzuordnen. Im Falle $m = 2$ erhalten wir wieder den Binomialkoeffizienten.

Satz 2.3.11 (Multinomialsatz) *Sei $n \in \mathbb{N}$. Dann ist*

$$(x_1 + x_2 + \ldots + x_m)^n = \sum_{\substack{k_1+\ldots+k_m=n \\ k_1,\ldots,k_m \geq 0}} \binom{n}{k_1, k_2, \ldots, k_m} x_1^{k_1} x_2^{k_2} \ldots x_m^{k_m}. \tag{2.10}$$

Beweis. Übung analog zum Binomialsatz. □

2.4 Abschätzungen

Nachdem wir kurzentschlossen 5 verschiedene Postkarten gekauft haben, sind wir immer noch unschlüssig, wie wir diese auf die Freunde verteilen wollen. Als Notmaßnahme rufen wir jeden an und bitten ihn, eine Zahl zwischen 1 und 5 zu nennen. Wie groß ist die Wahrscheinlichkeit, dass alle 5 Zahlen genannt werden?

Wie wir gelernt haben, geht es um die Wahrscheinlichkeit, dass eine zufällige Abbildung zwischen zwei gleichmächtigen Mengen ein Permutation ist. Diese Wahrscheinlichkeit ist also

$$\frac{n!}{n^n}.$$

Für den Fall $n = 5$ können wir die Zahl noch zu 0.0384 berechnen, wir haben also eine 4 prozentige Chance. Wie ist es aber im Allgemeinen?

Binomialkoeffizienten und Fakultäten wachsen sehr schnell. Manchmal ist es zu aufwändig oder schwierig, solche oder andere Größen exakt zu bestimmen. Oftmals ist aber auch schon mit Abschätzungen geholfen. In diesem und dem nächsten Abschnitt benutzen wir Resultate aus der Analysis, die aus der Schule bekannt sein sollten. Ansonsten verweisen wir dafür auf den Analysiskurs im dritten Semester.

Als Beispiel betrachten wir die Teilsummen der *harmonischen Reihe*.

$$H_n := 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n} = \sum_{i=1}^{n} \frac{1}{i}. \tag{2.11}$$

H_n heißt auch n-te harmonische Zahl und es ist für diese Summe keine geschlossenen Form bekannt, die sie vereinfacht. Wir schätzen nun H_n gegen den Logarithmus ab. Dafür teilen wir die Summanden in Päckchen und zwar setzen wir

$$G_k := \{\frac{1}{2^{k-1}}, \frac{1}{2^{k-1}+1}, \frac{1}{2^{k-1}+2}, \ldots, \frac{1}{2^k - 1}\}.$$

Die größte Zahl in G_k ist $\frac{1}{2^{k-1}}$, die kleinste ist $\frac{1}{2^k-1}$ und $|G_k| = 2^{k-1}$. Hieraus schließen wir

$$\frac{1}{2} = |G_k|\frac{1}{2^k} < \sum_{x \in G_k} x \leq |G_k|\frac{1}{2^{k-1}} = 1.$$

Aufsummiert erhalten wir

$$\frac{1}{2}\lfloor \log_2 n \rfloor = \sum_{k=1}^{\lfloor \log_2 n \rfloor} \frac{1}{2} < H_n \leq \sum_{k=1}^{\lfloor \log_2 n \rfloor + 1} 1 = \log_2 n + 1. \tag{2.12}$$

Genauer kann man sogar zeigen, dass $\ln n < H_n \leq \ln n + 1$. In gewissem Sinne ist diese Abschätzung nicht wesentlich schärfer als die eben angegebene. Der natürliche Logarithmus ist ein konstantes Vielfaches des Zweierlogarithmus und beide Abschätzungen sagen aus, dass die Teilsummen der harmonischen Reihe asymptotisch wie der Logarithmus wachsen. Dieses wollen wir jetzt formalisieren.

Definition 2.4.1 *Seien $f, g : \mathbb{N} \to \mathbb{R}$ Abbildungen. Dann schreiben wir*

$$f = O(g),$$

wenn es eine Konstante C und einen Startpunkt $n_1 \in \mathbb{N}$ gibt, so dass für alle $n \in \mathbb{N}, n \geq n_1$ gilt $|f(n)| \leq Cg(n)$.

Vorsicht! Die „Big-Oh“Notation liefert nur eine Abschätzung nach oben, nicht nach unten. Zum Beispiel ist $n = O(n^5)$.

Folgende Zusammenhänge sind nützlich bei der Abschätzung (z.B. auch von Laufzeiten von Algorithmen).

Proposition 2.4.2 *Seien $C, a, \alpha, \beta > 0$ feste reelle, positive Zahlen unabhängig von n. Dann gilt*

a) $\alpha \leq \beta \Rightarrow n^\alpha = O(n^\beta)$,

b) $a > 1 \Rightarrow n^C = O(a^n)$,

c) $\alpha > 0 \Rightarrow (\ln n)^C = O(n^\alpha)$.

Beweis.

a) $n^\beta = n^\alpha \overbrace{n^{\beta-\alpha}}^{\geq 1}$.

b) Wir betrachten die Folge $a_n := \left(\frac{n}{n-1}\right)^C$. Aus der Schule wissen wir, dass $\lim_{n\to\infty} a_n = 1$ ist. Da $a > 1$ ist, gibt es ein $n_1 \in \mathbb{N}$, so dass für alle $n \geq n_1$ gilt $a_n \leq a$. Nun setzen wir $C_1 := \frac{n_1^C}{a^{n_1}}$ und zeigen $n^C \leq C_1 a^n$ mittels vollständiger Induktion für $n \geq n_1$. Zu Anfang haben wir $n_1^C = C_1 a^{n_1}$. Sei also $n > n_1$. Dann ist nach Induktionsvoraussetzung

$$n^C = a_n (n-1)^C \leq a_n C_1 a^{n-1} \leq a C_1 a^{n-1} = C_1 a^n.$$

c) Wir setzen $a := e^\alpha$. Dann ist $a > 1$ und wir wählen n_1 und C_1 wie eben. Ferner wählen wir n_2 mit $\ln(n_2) \geq n_1$. Indem wir die Monotonie des Logarithmus und ein Verstetigungsargument (vgl. Analysis) ausnutzen, erhalten wir für $n \geq n_2$

$$\begin{aligned} (\ln n)^C &\leq C_1 a^{\ln n} \\ \Leftrightarrow (\ln n)^C &\leq C_1 (e^{\ln a})^{\ln n} = C_1 (e^{\ln n})^{\ln a} = C_1 n^{\ln a} \\ \Leftrightarrow (\ln n)^C &\leq C_1 n^\alpha. \end{aligned}$$

□

Beispiel 2.4.3 *Wenn man eine Formelsammlung zur Hand hat, schlägt man nach (und beweist mittels vollständiger Induktion), dass*

$$\sum_{i=1}^{n} i^3 = \frac{n^2(n+1)^2}{4}. \tag{2.13}$$

Hat man keine Formelsammlung zur Hand, ist die Herleitung dieser Formel recht mühselig. Darum schätzen wir ab: Zunächst ist $\sum_{i=1}^n i^3 \leq \sum_{i=1}^n n^3 = n^4$. *Außerdem ist* $\sum_{i=1}^n i^3 \geq \sum_{i=\lfloor \frac{n}{2} \rfloor}^n \left(\frac{n}{2}\right)^3 \geq \frac{n^4}{16}$. *Also verhält sich die Summe „bis auf einen konstanten Faktor“wie* n^4. *Genau genommen ist dieser „Faktor“ ein Intervall.*

Im Falle des Beispiels ist n^4 nicht nur eine obere, sondern auch eine untere Schranke. Auch dafür gibt es Symbole wie z.B.

$f(n) = o(g(n)) :\Leftrightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$, also wächst f echt langsamer als g,

$f(n) = \Omega(g(n)) \Leftrightarrow g(n) = O(f(n))$, $g(n)$ ist eine untere Schranke für f,

$f(n) = \Theta(g(n)) \Leftrightarrow f(n) = O(g(n))$ und $f(n) = \Omega(g(n))$, also verhalten sich f und g „bis auf einen konstanten Faktor“ gleich,

$f(n) \sim g(n) :\Leftrightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = 1$,wie eben aber „exakt“ mit Faktor 1.

2.5 Abschätzung für Fakultäten und Binomialkoeffizienten

Taschenrechner mit zweistelligem Exponenten versagen bei 70! Das Xwindows-Programm xcalc berechnet immerhin noch $170! = 7.25741 * 10^{306}$, 171! bis 500! sind infinity und für größere Zahlen erhält man nur noch error.

Zunächst haben wir die folgenden offensichtlichen Abschätzungen

Proposition 2.5.1

$$2^{n-1} \leq n! \leq n^n.$$

Beweis. Einerseits ist $2^{n-1} \leq \prod_{i=1}^{n} i = n!$ und andererseits kann man jeden der Faktoren nach oben gegen n abschätzen. □

Die Abschätzung ist recht grob und es drängt sich die Frage auf, ob die Fakultät näher bei der linken oder der rechten Seite liegt.

Die folgende, bessere Abschätzung geht auf Carl-Friedrich Gauß zurück, dessen Gesicht Ihnen noch vom 10 DM Schein bekannt ist.

Satz 2.5.2 *Für alle $n \geq 1$ ist*

$$n^{\frac{n}{2}} \leq n! \leq \left(\frac{n+1}{2}\right)^n. \tag{2.14}$$

Beweis. Der Beweis dieses Satzes benutzt eine Beziehung zwischen dem *arithmetischen Mittel* $\frac{a+b}{2}$ und dem *geometrischen Mittel* $\sqrt{ab}$ zweier positiver reeller Zahlen.

Lemma 2.5.3 (Ungleichung arithmetisches-geometrisches Mittel) *Seien $a, b > 0$ zwei reelle Zahlen. Dann ist*

$$\sqrt{ab} \leq \frac{a+b}{2}. \tag{2.15}$$

Beweis. Aus $0 \leq (\sqrt{a} - \sqrt{b})^2 = a - 2\sqrt{ab} + b$ folgt sofort die Behauptung. □

Beweis von Satz 2.5.2: Wir betrachten $(n!)^2 = (\prod_{i=1}^n i)\,(\prod_{i=1}^n (n+1-i)) = \prod_{i=1}^n i(n+1-i)$. Also gilt mit (2.15)

$$\begin{aligned} n! &= \prod_{i=1}^n \sqrt{i(n+1-i)} \\ &\leq \prod_{i=1}^n \frac{i+(n+1-i)}{2} \\ &= \left(\frac{n+1}{2}\right)^n, \end{aligned}$$

womit die obere Schranke bewiesen ist.

Für die untere genügt es zu beobachten, dass für $i = 1, \ldots, n$ stets $i(n+1-i) \geq n$. Dies ist sofort klar für $i = 1$ und $i = n$. Ansonsten haben wir das Produkt zweier Zahlen, bei dem die kleinere Zahl mindestens zwei und die größere mindestens $\frac{n}{2}$ ist. □

Die wichtigsten, weil genauesten Abschätzungen für die Fakultät erhalten wir mit Hilfe der *Eulerschen Zahl* $e = 2.718...$, der Basis des natürlichen Logarithmus. Wir setzen als (aus der Schule oder Analysis) bekannt voraus, dass für alle $x \in \mathbb{R}$

$$1 + x \leq e^x. \tag{2.16}$$

Satz 2.5.4 *Für alle $n \in \mathbb{N} \setminus \{0\}$ ist*

$$e\left(\frac{n}{e}\right)^n \leq n! \leq en\left(\frac{n}{e}\right)^n. \tag{2.17}$$

Beweis. Wir führen vollständige Induktion über n. Für $n = 1$ haben wir $1 \leq 1! \leq 1$. Sei also $n \geq 2$. Dann ist nach Induktionsvoraussetzung

$$\begin{aligned} e\left(\frac{n}{e}\right)^n &= e\left(\frac{n-1}{e}\right)^{n-1}\left(\frac{n}{e}\right)\left(\frac{n}{n-1}\right)^{n-1} \\ &\leq (n-1)!n\left(\frac{n}{n-1}\right)^{n-1}\frac{1}{e}, \end{aligned}$$

und analog

$$\begin{aligned} en\left(\frac{n}{e}\right)^n &= e(n-1)\left(\frac{n-1}{e}\right)^{n-1}\left(\frac{n}{e}\right)\left(\frac{n}{n-1}\right)^n \\ &\geq (n-1)!n\left(\frac{n}{n-1}\right)^n\frac{1}{e}. \end{aligned}$$

Für die Behauptung genügt es nun, noch zu zeigen , dass $\left(\frac{n}{n-1}\right)^{n-1}\frac{1}{e} \leq 1 \leq \left(\frac{n}{n-1}\right)^n\frac{1}{e}$ oder äquivalent

$$\left(\frac{n}{n-1}\right)^{n-1} \leq e \leq \left(\frac{n}{n-1}\right)^n. \tag{2.18}$$

Nach 2.16 ist nun $\frac{n}{n-1} = 1+\frac{1}{n-1} \leq e^{\frac{1}{n-1}}$ und andererseits $\frac{n-1}{n} = 1-\frac{1}{n} \leq e^{\frac{-1}{n}}$. Aus der ersten Ungleichung erhalten wir durch Exponation sofort die linke Ungleichung von (2.18) und aus der zweiten zunächst $\frac{n}{n-1} \geq e^{\frac{1}{n}}$ und dann die rechte. □

Ohne Beweis geben wir eine noch bessere Abschätzung an, die *Stirlingsche Formel*. Einen Beweis findet man z.B. in [4].

$$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n. \tag{2.19}$$

Wir erinnern daran, dass dies bedeutet, dass der Quotient der beiden Funktionen gegen 1 geht, also der *relative Fehler* gegen 0.

Aus den bewiesenen Formeln für die Fakultät leiten wir nun her den

Satz 2.5.5 *Seien* $1 \leq k \leq n \in \mathbb{N}$. *Dann ist*

$$\binom{n}{k} \leq \left(\frac{en}{k}\right)^k. \tag{2.20}$$

Beweis. Wir zeigen die stärkere Abschätzung

$$\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \ldots + \binom{n}{k} \leq \left(\frac{en}{k}\right)^k.$$

Zunächst einmal ist nach dem Binomischen Satz

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

Dies gilt inbesondere auch für positive x, also schließen wir, dass für $0 < x \leq 1$

$$\binom{n}{0} + \binom{n}{1}x + \binom{n}{2}x^2 + \ldots + \binom{n}{k}x^k \leq (1+x)^n$$

und somit auch

$$\frac{1}{x^k}\binom{n}{0} + \frac{1}{x^{k-1}}\binom{n}{1} + \frac{1}{x^{k-2}}\binom{n}{2} + \ldots + \binom{n}{k} \leq \frac{(1+x)^n}{x^k}.$$

Da

$$0 < x \leq 1$$

können wir die Terme $\frac{1}{x^l}$ nach unten gegen 1 abschätzen. Fixieren wir nun noch $x = \frac{k}{n}$, ergibt sich

$$\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \ldots + \binom{n}{k} \leq \left(1 + \frac{k}{n}\right)^n \left(\frac{n}{k}\right)^k.$$

Benutzen wir nun wieder (2.16), so erhalten wir

$$\left(1 + \frac{k}{n}\right)^n \leq \left(e^{\frac{k}{n}}\right)^n = e^k.$$

□

Aus der Definition der Binomialkoeffizienten folgt sofort die Beziehung

$$\binom{n}{k} = \frac{n-k+1}{k}\binom{n}{k-1}. \tag{2.21}$$

Aus dieser liest man ab, dass die Folge der Binomialkoeffizienten für wachsendes k bis $k = \lfloor \frac{n}{2} \rfloor$ wächst und hinter $k = \lceil \frac{n}{2} \rceil$ wieder fällt. Die größten Binomialkoeffizienten sind also von der Gestalt $\binom{2m}{m}$. Diesen Ausdruck wollen wir nun noch abschätzen.

Proposition 2.5.6 *Für alle $m \geq 1$ ist*

$$\frac{2^{2m}}{2\sqrt{m}} \leq \binom{2m}{m} \leq \frac{2^{2m}}{\sqrt{2m}}. \tag{2.22}$$

Beweis. Wir betrachten die Zahl

$$P = \frac{1 \cdot 3 \cdot 5 \cdot \ldots \cdot (2m-1)}{2 \cdot 4 \cdot 6 \cdot \ldots \cdot 2m}.$$

Dann ist

$$P = \frac{1 \cdot 3 \cdot 5 \cdot \ldots \cdot (2m-1)}{2 \cdot 4 \cdot 6 \cdot \ldots \cdot 2m} \frac{2 \cdot 4 \cdot 6 \cdot \ldots \cdot 2m}{2 \cdot 4 \cdot 6 \cdot \ldots \cdot 2m} = \frac{(2m)!}{2^{2m} m! m!} = \frac{\binom{2m}{m}}{2^{2m}}.$$

Also ist die Behauptung der Proposition äquivalent zu

$$\frac{1}{2\sqrt{m}} \leq P \leq \frac{1}{\sqrt{2m}}. \tag{2.23}$$

Für die obere Schranke von (2.23) betrachten wir das Produkt von

$$\left(\frac{1 \cdot 3}{2^2}\right) \left(\frac{3 \cdot 5}{4^2}\right) \ldots \left(\frac{(2m-1)(2m+1)}{(2m)^2}\right) = (2m+1)P^2.$$

Jeder der geklammerten Ausdrücke ist aber von der Gestalt $1 - \frac{1}{k^2}$ und somit ist das Produkt kleiner als 1. Also ist

$$P < \sqrt{\frac{1}{2m+1}} \leq \sqrt{\frac{1}{2m}}.$$

Für die untere Schranke benutzen wir analog

$$\left(\frac{2 \cdot 4}{3^2}\right) \left(\frac{4 \cdot 6}{5^2}\right) \ldots \left(\frac{(2m-2)2m}{(2m-1)^2}\right) = \frac{1}{2(2m)P^2} < 1.$$

□

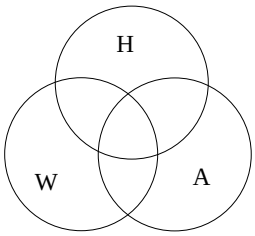
2.6 Das Prinzip von Inklusion und Exklusion

Wir erläutern das Zählprinzip dieses Abschnitts an einem einfachen Beispiel.

Beispiel 2.6.1 *In einer Gruppe Studenten besitzen 20 ein Handy, 15 ein Auto und 8 eine eigene Wohnung. Es gibt 2 Handybesitzer und 3 Autofahrer unter den Wohnungsinhabern, 6 handybesitzende Autofahrer und einen mit Wohnung, Auto und Handy. Aus wie vielen Studenten besteht die Gruppe, wenn jeder Student mindestens Auto, Handy oder Wohnung hat?*

Zählen wir zunächst die Gruppe von Studenten, die ein Handy oder eine Wohnung haben. Zählen wir Handybesitzer und Wohnungsinhaber zusammen, so haben wir zwei Studenten doppelt gezählt, also kommen wir auf $|H \cup W| = |H| + |W| - |H \cap W| = 28 - 2 = 26$.

Betrachten wir das Venn-Diagramm *der Situation.*



Wenn wir die Größen der einzelnen Mengen addieren, so haben wir die paarweisen Schnitte doppelt und den Schnitt aller Mengen dreifach gezählt. Ziehen wir die paarweisen Schnitte ab, so haben wir alle die Elemente genau einmal gezählt, die nicht in allen Mengen liegen. Also erhalten wir die Formel

$$|H \cup A \cup W| = |H| + |A| + |W| - |H \cap A| - |H \cap W| - |A \cap W| + |H \cap A \cap W|, \quad (2.24)$$

was in unserer Situation auf 33 Studenten schließen lässt.

Wenn wir diesen Ansatz verallgemeinern, kommen wir auf eine Formel wie

$$\begin{aligned} |A_1 \cup A_2 \cup \cdots \cup A_n| &= |A_1| + |A_2| + \cdots + |A_n| - |A_1 \cap A_2| - |A_1 \cap A_3| \\ &\quad - \cdots - |A_1 \cap A_n| - |A_2 \cap A_3| - \cdots + (-1)^{n-1} |A_1 \cap \ldots \cap A_{n-1} \cap A_n|. \end{aligned}$$

Diese Schreibweise ist sehr unübersichtlich und wir wollen Alternativen diskutieren. Eine Möglichkeit ist

$$\begin{aligned} |A_1 \cup A_2 \cup \cdots \cup A_n| &= \sum_{i=1}^{n} |A_i| - \sum_{1 \le i_1 < i_2 \le n} |A_{i_1} \cap A_{i_2}| \\ &+ \sum_{1 \le i_1 < i_2 < i_3 \le n} |A_{i_1} \cap A_{i_2} \cap A_{i_2}| - \cdots + (-1)^{n-1} |A_1 \cap \ldots \cap A_{n-1} \cap A_n|. \end{aligned}$$

Kürzer und (fast) ohne Punkte ist die folgende Schreibweise, die die Notation $\binom{X}{k}$ für die Menge aller k-elementigen Teilmengen benutzt.

Satz 2.6.2 (Prinzip von Inklusion und Exklusion, Siebformel) *Seien $A_1, \ldots, A_n$ endliche Teilmengen eines gemeinsamen Universums. Dann ist*

$$\left| \bigcup_{i=1}^{n} A_i \right| = \sum_{k=1}^{n} (-1)^{k-1} \sum_{I \in \binom{\{1,2,\ldots,n\}}{k}} \left| \bigcap_{i \in I} A_i \right|. \quad (2.25)$$

Den Beweis dieses wichtigen Satzes wollen wir auf zwei Arten führen. Einmal mittels vollständiger Induktion und dann mittels elementarem Abzählen.

Erster Beweis (vollständige Induktion über $n \geq 1$): Im Falle $n = 1$ ist die Aussage trivial und für $n = 2$ haben wir uns davon überzeugt, dass die Formel gilt. Sei also $n \geq 3$. Dann ist

$$\left|\bigcup_{i=1}^{n} A_i\right| = \left|A_n \cup \bigcup_{i=1}^{n-1} A_i\right| = \left|\bigcup_{i=1}^{n-1} A_i\right| + |A_n| - \left|\left(\bigcup_{i=1}^{n-1} A_i\right) \cap A_n\right|.$$

In der letzten Gleichung haben wir die Gültigkeit der Formel für $n = 2$ ausgenutzt. Nun wenden wir die Induktionsvoraussetzung an und erhalten:

$$\begin{aligned}
\left|\bigcup_{i=1}^{n} A_i\right| &= \left|\bigcup_{i=1}^{n-1} A_i\right| + |A_n| - \left|\bigcup_{i=1}^{n-1} (A_i \cap A_n)\right| \\
&= \left(\sum_{k=1}^{n-1} (-1)^{k-1} \sum_{I \in \binom{\{1,2,\ldots,n-1\}}{k}} \left|\bigcap_{i \in I} A_i\right|\right) + |A_n| \\
&\quad - \sum_{k=1}^{n-1} (-1)^{k-1} \sum_{I \in \binom{\{1,2,\ldots,n-1\}}{k}} \left|\bigcap_{i \in I \cup \{n\}} A_i\right| \\
&= \left(\sum_{k=1}^{n-1} (-1)^{k-1} \sum_{I \in \binom{\{1,2,\ldots,n-1\}}{k}} \left|\bigcap_{i \in I} A_i\right|\right) + |A_n| \\
&\quad + \sum_{k=2}^{n} (-1)^{k-1} \sum_{n \in I \in \binom{\{1,2,\ldots,n-1,n\}}{k}} \left|\bigcap_{i \in I} A_i\right|.
\end{aligned}$$

In der ersten Summe treten alle Teilmengen von $\{1, \ldots, n\}$ auf, die n nicht enthalten, dahinter alle, die n enthalten. Die Vorzeichen sind richtig, also die Behauptung bewiesen. □

Zweiter Beweis (mittels Abzählen): Wir untersuchen, wie oft ein festes Element $x \in A_1 \cup \cdots \cup A_n$ auf der rechten Seite gezählt wird. Sei $J \in \{1, \ldots, n\}$ die Menge der Indizes $l \in J$ mit $x \in A_l$ und $|J| = j$. Dann trägt x auf der rechten Seite zu jedem Summanden genau eins bei, für dessen Indexmenge I gilt $I \subseteq J$. Nun gibt es $\binom{j}{k}$ k-elementige Teilmengen von $\{1, \ldots, n\}$, die in J enthalten sind, nämlich genau dessen Teilmengen. Das Element x wird also auf der rechten Seite genau

$$j - \binom{j}{2} + \binom{j}{3} - \cdots + (-1)^{j-1} \binom{j}{j} = \binom{j}{0} - (1-1)^j = 1$$

mal gezählt. □

Beispiel 2.6.3 *Wir haben von n Freunden je einen Witz aufgeschnappt und uns zwar die Pointe aber nicht den Erzähler gemerkt. Als kommunikative Menschen erzählen wir jedem der Freunde genau einen zufälligen dieser n Witze, aber jedem einen anderen. Wie groß ist die Wahrscheinlichkeit, dass wir keinem Freund seinen eigenen Witz erzählen.*

Abstrakt suchen wir nach der Wahrscheinlichkeit, dass eine zufällige Permutation keinen Fixpunkt hat. Betrachten wir nämlich die Abbildung, die jedem Witze erzählenden Freund den Empfänger des Witzes zuordnet, so erhalten wir eine Permutation $\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$. Wir erzählen niemandem seinen eigenen Witz, wenn $\sigma(i) \neq i$ für alle $1 \leq i \leq n$, also σ keinen Fixpunkt, d.i. ein i mit $\sigma(i) = i$ hat. Sei $D(n)$ die Anzahl der fixpunktfreien Permutationen.

Da jede Permutation gleichwahrscheinlich ist, ist die gesuchte Wahrscheinlichkeit $\frac{D(n)}{n!}$.

Wir zählen dafür die Permutationen mit Fixpunkt. Wir können nämlich sehr leicht die Permutationen zählen, die mindestens eine gegebene Menge von k Elementen festlassen. Dies sind ja genau die Permutationen der übrigen $n-k$ Elemente, also $(n-k)!$ Stück.

Die Menge aller Permutationen der Menge $\{1, \ldots, n\}$ kürzen wir mit S_n ab.

Sei für $i = 1, \ldots, n : A_i := \{\sigma \in S_n \mid \sigma(i) = i\}$. Dann ist $D(n) = n! - |A_1 \cup A_2 \cdots \cup A_n|$. Ferner haben wir $|A_i| = (n-1)!$ und ist $I \subseteq \{1, \ldots, n\}$, dann ist

$$\left| \bigcap_{i \in I} A_i \right| = (n - |I|)!$$

Setzen wir dies in das Prinzip von Inklusion und Exklusion ein, so erhalten wir, da es jeweils $\binom{n}{k}$ k-elementige Indexmengen gibt und die zugehörigen Schnitte alle die gleiche Kardinalität haben:

$$|A_1 \cup \ldots \cup A_n| = \sum_{k=1}^{n} (-1)^{k-1} \binom{n}{k} (n-k)! = \sum_{k=1}^{n} (-1)^{k-1} \frac{n!}{k!}.$$

Wir halten fest

Satz 2.6.4 *Die Anzahl der fixpunktfreien Permutationen einer n-elementigen Menge ist*

$$D(n) = \sum_{i=0}^{n} (-1)^i \frac{n!}{i!}.$$

□

Kommen wir zurück zu der Fragestellung, so sehen wir, dass wir die Wahrscheinlichkeit als $\sum_{i=0}^{n}(-1)^i\frac{1}{i!}$ erhalten. Diese Zahl konvergiert mit $n \to \infty$ gegen $e^{-1} = 0.36787\ldots$ (vgl. Schule oder Analysis) und zwar sehr schnell. Für $n = 5$ hat man schon $0.36666666\ldots$. Die Wahrscheinlichkeit hängt hier also fast nicht von n ab.

Als letzte Anwendung in diesem Kapitel zählen wir zu einer Zahl n die teilerfremden kleineren Zahlen. Diese Anzahl heißt Eulerfunktion und spielt in der Zahlentheorie und in der Kryptographie eine wichtige Rolle. Bezeichne für zwei Zahlen $a, b, ggT(a, b)$ den größten gemeinsamen Teiler dieser beiden Zahlen. Dann ist die *Eulersche φ-Funktion* definiert durch

$$\varphi(n) = |\{m \in \{1, 2, \ldots, n\} \mid ggT(n, m) = 1\}|\,.$$

Ist n eine Primzahl $n = p$, so ist offensichtlich $\varphi(p) = p - 1$. Als nächstes untersuchen wir Primzahlpotenzen $n = p^k$ mit $k \in \mathbb{N}, k \geq 2$. Wir zählen dann alle Zahlen $\leq p^k$, die keine Vielfachen von p sind, das sind $p^k - p^{k-1} = p^k(1-\frac{1}{p})$ Stück.

Sei nun n eine beliebige natürliche Zahl. Dann kann man es in seine Primfaktoren zerlegen:

$$n = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_r^{\alpha_r},$$

wobei $p_1, \ldots, p_r$ die verschiedenen Primteiler von n sind, also $\alpha_i \geq 1$. Dann setzen wir

$$A_i := \{m \in \{1, 2, \ldots, n\} \mid p_i \text{ teilt } m\}.$$

Dann ist

$$\varphi(n) = n - |A_1 \cup A_2 \cup \cdots \cup A_r|.$$

Die Menge $\bigcap_{i\in I} A_i$ für $I \subseteq \{1, \ldots, r\}$ besteht aus allen Zahlen $\leq n$, die durch $\prod_{i\in I} p_i$ teilbar sind, also $n/\prod_{i\in I} p_i$ vielen. Nun können wir mit dem Prinzip von Inklusion und Exklusion zeigen

Satz 2.6.5 *Sei $n = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_r^{\alpha_r}$. Dann ist*

$$\varphi(n) = n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\cdots\left(1 - \frac{1}{p_r}\right). \tag{2.26}$$

Beweis. Das Prinzip von Inklusion und Exklusion liefert

$$\varphi(n) = n - \sum_{\emptyset\neq I\subseteq\{1,2,\ldots,r\}} (-1)^{|I|-1}\frac{n}{\prod_{i\in I} p_i} = n \sum_{I\subseteq\{1,2,\ldots,r\}} (-1)^{|I|}\frac{1}{\prod_{i\in I} p_i}.$$

Dass diese Formel mit der behaupteten übereinstimmt, zeigen wir mittels vollständiger Induktion über r, den Fall $r = 1$ haben wir oben schon diskutiert. Sei also $r \geq 2$. Dann ist

$$\begin{aligned}
\prod_{i=1}^{r}\left(1-\frac{1}{p_i}\right) &= \left(1-\frac{1}{p_r}\right)\prod_{i=1}^{r-1}\left(1-\frac{1}{p_i}\right) \\
&= \left(1-\frac{1}{p_r}\right)\sum_{I\subseteq\{1,2,\ldots,r-1\}}(-1)^{|I|}\frac{1}{\prod_{i\in I}p_i} \\
&= \sum_{I\subseteq\{1,2,\ldots,r-1\}}(-1)^{|I|}\frac{1}{\prod_{i\in I}p_i} - \sum_{r\in I\subseteq\{1,2,\ldots,r\}}(-1)^{|I|-1}\frac{1}{\prod_{i\in I}p_i} \\
&= \sum_{I\subseteq\{1,2,\ldots,r\}}(-1)^{|I|}\frac{1}{\prod_{i\in I}p_i}.
\end{aligned}$$

□

Kapitel 3

Einführung in Graphen

Wir haben einen Spezialfall von Graphen bereits bei den HASSE-Diagrammen kennengelernt. Graphen bestehen aus einer endlichen Menge von Knoten (Objekten, Punkten) und Kanten, die je zwei dieser Knoten verbinden. Sie sind also recht nützlich, um Straßennetzwerke oder Relationen zu kodieren und werden Sie ihr ganzes Studium über begleiten.

3.1 Definition eines Graphen, Isomorphismus

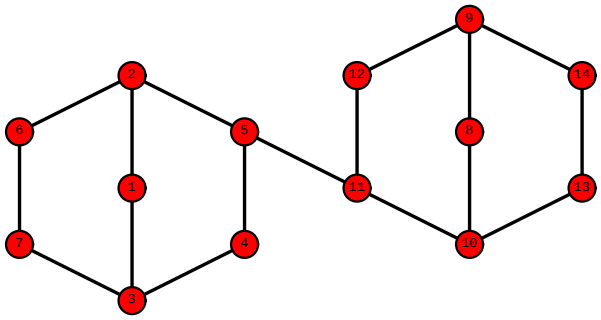
Wir geben hier zwei Definitionen eines Graphen, eine einfache, nützliche, die aber leider nicht in allen Situationen ausreichend ist und eine allgemeine, umständliche. Fangen wir mit der einfachen an.

Definition 3.1.1 *Sei V eine endliche Menge (von* Knoten*) und $E \subseteq \binom{V}{2}$ eine Teilmenge der zweielementigen Teilmengen von V. Dann nennen wir das geordnete Paar (V, E) einen* Graph *(genauer einen ungerichteten, einfachen Graphen).*

Haben wir einen Graphen G gegeben, dann bezeichnen wir seine Knotenmenge auch mit $V(G)$ und seine Kanten mit $E(G)$. Ist $\{u, v\}$ eine Kante eines Graphen G, sagen wir, u und v sind *adjazent* oder *Nachbarn*. Manchmal schreiben wir auch einfach (u, v) für eine Kante $\{u, v\}$.

Beispiel 3.1.2 *Wir können Graphen zeichnen, indem wir für jeden Knoten einen Punkt in die Ebene zeichnen und die Punkte durch eine Linie verbinden, wenn es die entsprechende Kante gibt. In der Abbildung sehen Sie einen*

Graphen mit 14 Knoten und 17 Kanten.

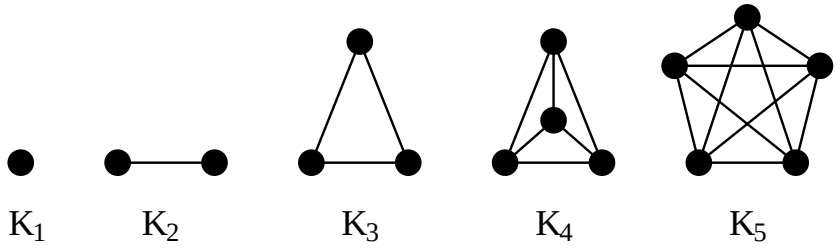


Man beachte aber, dass diese Skizze nur eine Visualisierung des abstrakten Objekts ist. Z.B. für die algorithmische Behandlung speichern wir den Graphen als Listen $V = \{1,2,3,4,5,6,7,8,9,10,11,12,13,14\}$,
$E = \{\{1,2\},\{1,3\},\{2,5\},\{2,6\},\{3,4\},\{3,7\},\{4,5\},\{6,7\},\{5,11\},\{8,9\},$
$\{8,10\},\{9,12\},\{9,14\},\{10,11\},\{10,13\},\{11,12\},\{13,14\}\}$.

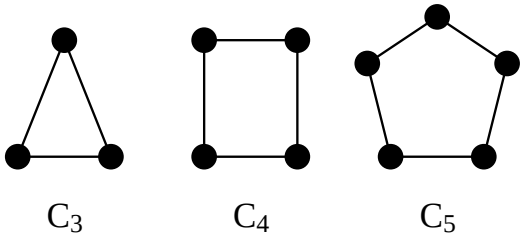
Wir führen nun einige wichtige Graphenklassen ein.

Beispiel 3.1.3 *Sei* $n \in \mathbb{N}$ *und* $V = \{1,2,\ldots,n\}$.

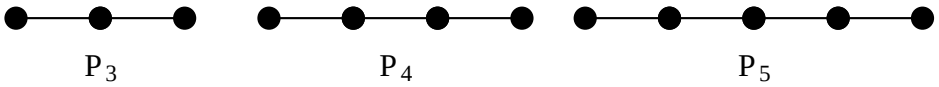
- K_n, *der* vollständige Graph mit n Knoten. *Dann ist* K_n *der Graph mit Knotenmenge und Kantenmenge* $\binom{V}{2}$.



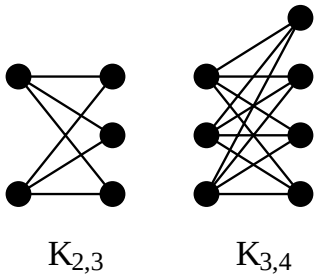
- *Sei* $n \geq 3$. *Der* Kreis mit n Knoten C_n *hat die Kantenmenge* $\{1,n\}$ *und* $\{i,i+1\}$ *für* $i = 1,\ldots,n-1$.



- P_n, *der* Weg mit n Knoten *hat Kantenmenge* $\{i, i+1\}$ *für* $i = 1, \ldots, n-1$.



- $K_{m,n}$ *der vollständige, bipartite Graph mit* $m + n$ *Knoten hat Knotenmenge* $V \cup W$ *mit* $W = \{1, \ldots, m\}$ *und Kantenmenge* $V \times W$.

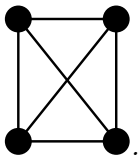


Wir sehen zwei Graphen als gleich an, wenn sie im Wesentlichen aus der selben Knoten- und Kantenmenge bestehen. Genauer definieren wir:

Definition 3.1.4 *Seien* $G = (V, E)$ *und* $G' = (V', E')$ *zwei Graphen. Dann heißen* G *und* G' isomorph*, wenn es eine Bijektion* $f : V \to V'$ *gibt mit*

$$\text{Für alle } u, v \in V : \{u, v\} \in E \Leftrightarrow \{f(u), f(v)\} \in E'.$$

Die Abbildung f *heißt dann ein* Isomorphismus *und wir schreiben* $G \cong G'$.

Beispiel 3.1.5 *Der* K_4 *ist isomorph zu dem Graphen*  .

Bei kleinen Graphen kann man noch alle möglichen Permutationen der Knoten enummerieren, um zu entscheiden, ob zwei Graphen isomorph sind. Für den allgemeinen Fall ist jedoch kein effizienter Algorithmus bekannt (die Anzahl der Permutationen wächst zu stark, um effizient enummeriert werden zu können). Man vermutet, dass es keinen effizienten Algorithmus gibt.

Wir wollen im Folgenden die Anzahl nicht isomorpher Graphen mit n Knoten abschätzen. Sei also $V = \{1, \ldots, n\}$. Jede Teilmenge von $\binom{V}{2}$ definiert

zunächst einmal einen Graphen. Allerdings haben wir hierbei isomorphe Graphen. Z.B. gibt es drei isomorphe Graphem auf $\{1,2,3\}$ mit einer Kante. Isomorphe Graphen werden aber durch eine Permutation der Knoten ineinander überführt. Also haben wir von jedem Graphen höchstens $n!$ isomorphe Kopien gezählt. Einige Graphen (wie den Graphen ohne Kanten) haben wir zwar nur einmal gezählt, aber dennoch bewiesen: Es gibt mindestens

$$\frac{2^{\binom{n}{2}}}{n!}$$

paarweise nicht isomorphe Graphen mit n Knoten. Wir schätzen die Größenordnung dieser Zahl ab. Dafür genügt die grobe Schranke $n! \leq n^n$. Diese impliziert.

$$\begin{aligned}\log_2\left(\frac{2^{\binom{n}{2}}}{n!}\right) &= \binom{n}{2} - \log_2(n!) \\ &\geq \frac{n^2}{2} - \frac{n}{2} - n\log_2(n) \\ &= \frac{n^2}{2}\left(1 - \frac{1}{n} - \frac{2\log_2(n)}{n}\right).\end{aligned}$$

Der Ausdruck in der letzten Klammer verhält sich für große n etwa wie $\frac{n^2}{2}$. Also ist die Anzahl paarweise nicht isomorpher Graphen mit n Knoten für wachsendes n deutlich größer als die Anzahl der Teilmengen einer n-elementigen Menge.

Wir hatten zu Anfang dieses Abschnitts zwei Definitionen für Graphen versprochen. In der bisherigen Definition kann es zwischen je zwei Knoten höchstens eine Kante geben. Manchmal kann es notwendig und sinnvoll sein, solche *parallelen Kanten* zuzulassen. Auch *Schleifen*, das sind Kanten, bei denen die Endknoten übereinstimmen, können auftreten. In diesem Falle ist unsere bisherige Definition unzureichend. Da für diese Vorlesung unser Graphenbegriff aber ausreicht, wollen wir die andere Definition als *Multigraphen* bezeichnen.

Definition 3.1.6 *Ein* Multigraph $G = (V, E, ad)$ *ist ein Tripel bestehend aus einer endlichen Menge V (von Knoten), einer endlichen Menge E (von Kanten) und einer Adjazenzfunktion $ad : E \to \binom{V}{2} \cup V$, die jeder Kante einen oder zwei Endknoten zuordnet. Haben $e, e' \in E$ die gleichen Endknoten, so heißen sie* parallel. *Eine Kante mit nur einem Endknoten heißt Schleife.*

3.2 Teilgraphen, Komponenten, Adjazenzmatrix

Wir wollen zunächst eine Enthaltenseinsbeziehung für Graphen definieren.

Definition 3.2.1 *Seien $G = (V, E)$ und $H = (W, F)$ zwei Graphen. Dann heißt H ein* Teilgraph *von G, wenn $W \subseteq V$ und $F \subseteq E$. Darüberhinaus sagen wir H ist ein* induzierter Teilgraph*, wenn $F = E \cap \binom{W}{2}$.*

Ein induzierter Teilgraph besteht also aus einer Teilmenge der Knoten und allen Kanten, die im Ausgangsgraphen zwischen diesen Knoten existieren.

Beispiel 3.2.2 *Der P_4 ist ein induzierter Teilgraph des C_5 und ein (nicht induzierter Teilgraph des C_4.*

Ein Teilgraph, der isomorph zu einem Weg P_t ist, heißt *Weg* oder *Pfad* im Graphen. Einen Weg kann man als alternierende Sequenz von paarweise verschiedenen Knoten und Kanten

$$(v_0, e_1, v_1, e_2, \ldots, e_k, v_k)$$

mit $e_i = (v_{i-1}, v_i)$ darstellen. Oft notieren wir Wege auch nur als Knotensequenz $(v_0, v_1, \ldots, v_k)$ oder Kantensequenz $(e_1, e_2, \ldots, e_k)$. Wir nennen einen solchen Weg auch *einen v_0-v_k-Weg der Länge k.*

Analog nennen wir einen Teilgraphen, der isomorph zu einem Kreis ist, einen Kreis in G. Auch Kreise kann man als Knoten-Kantenfolge oder auch als Knotenfolge bzw. Kantenfolge notieren. Diese müssen jeweils (bis auf Anfangs- und Endknoten) paarweise verschieden sein. Die Anzahl der Kanten oder Knoten eines Kreises heißt die *Länge* des Kreises.

Definition 3.2.3 *Ein Graph $G = (V, E)$ heißt* zusammenhängend*, wenn es zu je zwei Knoten u, v einen u-v-Weg gibt. Ein mengentheoretisch maximaler zusammenhängender Teilgraph eines Graphen heißt* Komponente*.*

Beispiel 3.2.4 *Den Zusammenhang sieht man der Zeichnung nicht immer sofort an. Der Davidsstern in Abbildung 3.1 ist unzusammenhängend und hat zwei Komponenten.*

Der Nachteil bei der Definition des Zusammenhangs über Wege ist, dass wir stets darauf achten müssen, dass diese Wege keine Wiederholungen von Knoten oder Kanten haben.

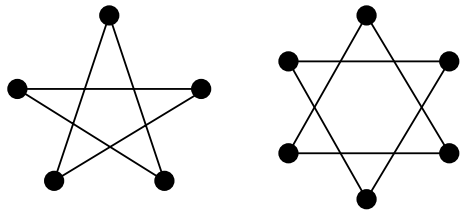


Abbildung 3.1: Das Pentagramm ist zusammenhängend, der Davidsstern nicht.

Definition 3.2.5 *Eine alternierende Folge von Knoten und Kanten* $(v_0, e_1, v_1, e_2, \ldots, e_k, v_k = v_t)$ *mit* $e_i = (v_{i-1}, v_i)$ *heißt* Spaziergang der Länge k *von* v_0 *nach* v_k.

Proposition 3.2.6 *Sei* $G = (V, E)$ *ein Graph und* $v_0, v_t \in V$. *Es gibt genau dann einen* v_0-v_t-*Weg, wenn es einen Spaziergang von* v_0 *nach* v_t *gibt.*

Beweis. Jeder Weg ist auch ein Spaziergang. Gibt es nun einen Spaziergang $(v_0, e_1, v_1, e_2, \ldots, e_k, v_k)$ von v_0 nach v_t, so gibt es auch einen darunter, der die kürzeste Länge hat. Dieser muss ein Weg sein. Denn angenommen $v_i = v_j$ mit $i < j$, so wäre $(v_0, e_1, v_1, \ldots, e_{i-1}, v_i, e_j, v_{j+1}, \ldots, e_k, v_k)$ ein kürzerer Spaziergang von v_0 nach v_t im Widerspruch zur Annahme. □

Auf Grund dieser Tatsache identifizieren wir die Knoten der Komponenten als Äquivalenzklassen der Äquivalenzrelation (!)

$$a \sim b \Leftrightarrow \text{ es gibt einen Spaziergang von } a \text{ nach } b.$$

Die Zusammenhangskomponenten bestimmt man algorithmisch mit Suchverfahren, z.B. mit Breitensuche oder Tiefensuche, die wir später kennenlernen werden.

In einem zusammenhängenden Graphen können wir endliche Distanzen definieren.

Definition 3.2.7 *Sei* $G = (V, E)$ *ein zusammenhängender Graph und* $u, v \in V$. *Die Länge eines kürzesten* u, v-*Weges nennen wir den* Abstand $dist(u, v)$ *von* u *und* v *in* G.

Die *Abstandsfunktion* oder *Metrik* ist also eine Abbildung $dist_G : V \times V \to \mathbb{N}$.

Proposition 3.2.8 *Die Metrik eines Graphen erfüllt*

Nichtnegativität: $dist_G(u,v) \geq 0$ *und* $dist_G(u,v) = 0 \Leftrightarrow u = v$.

Symmetrie: *Für alle* $u, v \in V : dist_G(u,v) = dist_G(v,u)$.

Dreiecksungleichung: *Für alle* $u, v, w \in V : dist_G(u,w) \leq dist_G(u,v) + dist_G(v,w)$.

Beweis. Die ersten beiden Eigenschaften sind trivial. Im dritten Fall erhält man durch Verkettung eines kürzesten u-v-Weges mit einem kürzesten v-w-Weg einen Spaziergang von u nach w. □

Graphen spielen in der Datenverarbeitung eine große Rolle. Üblicherweise werden sie als Adjazenzlisten abgespeichert. Manchmal sind aber auch zwei Darstellungen mit *Matrizen* sinnvoll.

Definition 3.2.9 *Sei M eine Menge und $m, n \in \mathbb{N}$. Eine* $(m \times n)$-Matrix A *(über M) ist ein mn-Tupel von Elementen von M, die in einem rechteckigen Schema in m* Zeilen *und n* Spalten *angeordnet werden. Mit a_{ij} bezeichnen wir den Eintrag in der i-ten Zeile und j-ten Spalte.*

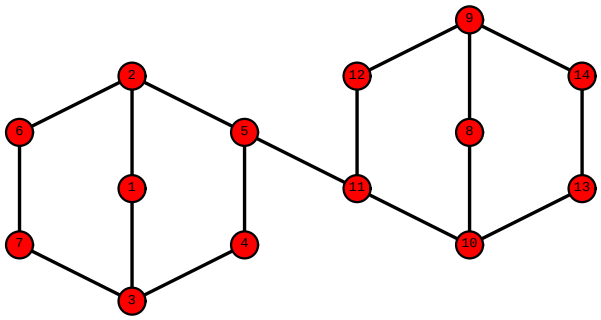
Definition 3.2.10 *Sei $G = (V, E)$ ein Graph mit Knotenmenge $V = \{v_1, \ldots, v_n\}$ und Kantenmenge $E = \{e_1, \ldots, e_m\}$. Die* Adjazenzmatrix $A_G = (a_{ij})_{i,j=1}^{n}$ *ist dann eine $n \times n$-Matrix definiert vermöge*

$$a_{ij} = \begin{cases} 1 & \text{wenn } (v_i, v_j) \in E \\ 0 & \text{sonst.} \end{cases}$$

Die Knoten-Kanten Inzidenzmatrix B_G *ist eine $n \times m$-Matrix $B_G = (b_{ij})_{(i,j)=1}^{(n,m)}$ definiert vermöge*

$$b_{ij} = \begin{cases} 1 & \text{wenn } v_i \text{ Endknoten von } e_j \text{ ist.} \\ 0 & \text{sonst.} \end{cases}$$

Beispiel 3.2.11 *Wir betrachten den Graphen aus Beispiel 3.1.2.*



Dieser hat die Adjazenzmatrix

$$A_G = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \end{array} \begin{array}{c} \begin{array}{cccccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \end{array} \\ \left(\begin{array}{cccccccccccccc} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{array} \right) \end{array}$$

und die Knoten-Kanten-Inzidenzmatrix

$$B_G = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \end{array} \begin{array}{c} \begin{array}{ccccccccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 \end{array} \\ \left(\begin{array}{ccccccccccccccccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{array} \right) \end{array}.$$

Die Matrizenschreibweisen sind oftmals von eher theoretischem Interesse, als dass sie eine sinnvolle Kodierung für die Datenverarbeitung wären.

3.3 Breadth First Search

In Computeranwendungen wird man in der Regel, wie gesagt, Adjazenzlisten verwenden. Eine mögliche Kodierung ist eine (doppelt) verkettete Liste von Knoten. Jeder Knoten hat einen Zeiger auf den Anfang seiner Adjazenzliste, die eine (doppelt) verkettete Liste von Zeigern auf Kanten ist. Die Kanten wiederum haben Zeiger auf Anfangs- und Endknoten. Dadurch ist sichergestellt, dass man an einem Knoten in konstanter Zeit eine nächste Kante erhält und von einer Kante in konstanter Zeit Anfangs- und Endknoten erfährt. In Abbildung 3.2 haben wir einen Graphen mit zugehöriger Datenstruktur skizziert.

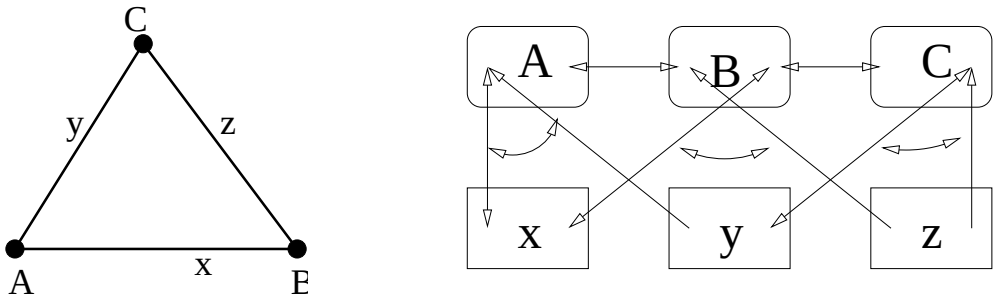


Abbildung 3.2: Ein Graph mit Adjazenzliste

Wir wollen am Beispiel der Breitensuche nun studieren, wie man diese Datenstruktur in einem Algorithmus anspricht. Wir betrachten die Breitensuche zur Berechnung der Komponenten eines Graphen. Dafür iterieren wir wie folgt:

- solange es einen unbearbeiteten Knoten w gibt, markiere diesen Knoten als zur w-Komponente gehörend und stelle ihn in eine neue Warteschlange.
- solange die Warteschlange nicht leer ist, nehmen wir den Kopf v der Schlange und markieren seine Nachbarn, die weder abgearbeitet noch in der Schlange sind, als zur gleichen Komponente gehörend wie v und stellen sie ans Ende der Schlange.

Wir erhalten folgenden Code

```
for v in Vertices:
  if pred[v] == None:                                    (1)
    pred[v]= v
```

```
    component[v]=v                                    (2)
    Q.Append(v)
  while Q.IsNotEmpty():
    v = Q.Top()
    for w in Neighborhood(v):
      if pred[w] == None:
        pred[w] = v
        component[w] = component[v]                   (3)
        Q.Append(w)
```

In dem if-Block von (1) bis (2) initialisieren wir eine neue Komponente und bestimmen diese in der while-Schleife. Zusätzlich merken wir uns für jeden Knoten noch seinen Vorgänger, von dem aus er markiert wurde. Der durch diese Vorgängerrelation definierte Graph ist ein Wald (vgl. nächstes Kapitel), da er keine Kreise enthält. Die einzelnen Komponenten heißen Breitensuchbaum oder BFS-tree.

Wir wollen noch den Rechenaufwand zur Bestimmung der Komponenten verifizieren. Die äußere for-Schleife betreten wir $O(|V|)$ mal, müssen eventuell $O(|V|)$ mal die while-Schleife ausführen und darin bis zu $O(|V|)$ mal Nachbarn abarbeiten. Fassen wir dies zusammen, so kommen wir zur einer Abschätzung von $O(|V|^3)$.Wie wir sehen werden, ist diese Abschätzung extrem schlecht und ungeschickt. Günstiger ist es, wenn wir zunächst einmal festhalten, dass der if-Block für jeden Knoten höchstens einmal ausgeführt wird. Die Gesamtarbeit im if-Block ist also $O(|V|)$.

Die Gesamtarbeit in der while-Schleife können wir ermitteln, wenn wir zählen, wie oft die innere for-Schleife betreten wird. Zunächst einmal halten wir dafür fest, dass jeder Knoten genau einmal in der Schlange steht, da im Folgenden sein Vorgänger gesetzt ist. Also werden in der for-Schleife alle Nachbarschaftsbeziehungen für jeden Knoten abgearbeitet. Jede Kante vermittelt genau zweimal, nämlich für beide Endknoten, eine solche Relation. Also wird die for-Schleife $O(|E|)$-mal abgearbeitet und wir erhalten als Gesamtlaufzeit

Satz 3.3.1 *BFS berechnet die Komponenten eines Graphen in* $O(|V|+|E|)$. □

Wir haben zwar die Aussage über die Laufzeit dieser Aussage bewiesen, aber die Aussage über die Komponenten noch nicht. Wir werden auf dieses Thema zurückkommen, wenn wir aufspannende Bäume von Graphen im nächsten Kapitel behandeln.

In den Listenstrukturen kann man auch die allgemeinere Definition eines Multigraphen kodieren, was bei Adjazenzmaztrizen nur bedingt möglich ist.

3.4 Valenzsequenzen

Sei $G = (V, E)$ ein Graph und $v \in V$. Der *Knotengrad* oder *die Valenz* $deg_G(v)$ von v ist dann die Anzahl Kanten, deren Endknoten v ist. (In Multigraphen werden Schleifen dabei doppelt gezählt.)

Ist $v_1, \ldots, v_n$ (irgend) eine lineare Anordnung der Knoten, so nennen wir

$$(\ \deg(v_1),\ \deg(v_2), \ldots,\ \deg(v_n))$$

die *Gradsequenz* oder *Valenzsequenz* des Graphen. Wir sehen zwei Valenzsequenzen als gleich an, wenn sie durch Umordnen auseinander hervorgehen. Deswegen gehen wir im Folgenden davon aus, dass die Zahlen der Größe nach, und zwar nicht aufsteigend sortiert sind. Der Graph aus Beispiel 3.1.2 hat dann die Valenzsequenz

$$(3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2).$$

Kann man einem Zahlentupel ansehen, ob es eine Valenzsequenz eines Graphen ist? Zunächst einmal können wir Folgen wie $(3, 3, 3, 2, 2, 2)$ ausschließen:

Proposition 3.4.1 (Handshake Lemma) *In jedem Graphen $G = (V, E)$ ist die Summe der Knotengrade gerade, genauer gilt*

$$\sum_{v \in V} deg(v) = 2|E|. \tag{3.1}$$

Beweis. Links wird jede Kante für jeden Endknoten genau einmal, also insgesamt doppelt, gezählt. □

Als direkte Konsequenz erhalten wir:

Korollar 3.4.2 *In jedem Graphen ist die Anzahl der Knoten mit ungeradem Knotengrad gerade.*

□

Dies charakterisiert aber Gradfolgen noch nicht, denn z.B. ist $(4, 3, 1, 1, 1)$ keine Gradfolge, da aus den ersten zwei Knoten noch mindestens jeweils 3 bzw. 2 Kanten in die anderen drei führen müssten, die aber keine Chance haben, anzukommen. Im Allgemeinen gilt der folgende Satz, dessen Beweis etwas den Rahmen diese Vorlesung sprengt. Wir zeigen nur die Notwendigkeit.

Satz 3.4.3 (Erdös und Gallai 1963) *Sei $d_1 \geq d_2 \geq \ldots \geq d_n \geq 0$ eine Folge natürlicher Zahlen. Dann ist $(d_1, \ldots, d_n)$ genau dann die Gradsequenz eines einfachen Graphen, wenn $\sum_{i=1} d_i$ gerade ist und*

$$\forall i = 1, \ldots, n : \sum_{j=1}^{i} d_j \leq i(i-1) + \sum_{j=i+1}^{n} \min\{i, d_j\}. \tag{3.2}$$

Beweis. Einen vollständigen Beweis findet man z.B. in [6]. Wir zeigen die Notwendigkeit. Dass die Valenzsumme gerade sein muss, sagt das Handshake Lemma. Ist G ein Graph mit der angegebenen Valenzsequenz und ist $I = \{1, \ldots, i\}$ die Menge der Knoten mit Knotengrad $d_1, \ldots, d_i$, dann kann jeder der i Knoten in I höchstens alle anderen $i-1$ Knoten in I kennen. Also müssen alle Knoten in I insgesamt mindesten $\sum_{j=1}^{i} d_j - i(i-1)$ Knoten außerhalb von I kennen. Jeder Knoten $v \in V \setminus I$ kann aber höchstens $\min\{i, deg(v)\}$ Knoten in I kennen. □

Die folgende rekursive Charakterisierung führt auf einen Algorithmus zum Erkennen von Valenzsequenzen.

Satz 3.4.4 *Sei $D = (d_1, d_2, \ldots, d_n)$ eine Folge natürlicher Zahlen, $n > 1$ und $d_1 \geq d_2 \geq \ldots, d_n \geq 0$. Dann ist D genau dann die Valenzsequenz eines einfachen Graphen, wenn die Folge $D' = (d'_2, d'_3, \ldots, d'_n)$ definiert durch*

$$d'_i := \begin{cases} d_i - 1 & \text{für } i = 2, d_1 + 1 \\ d_i & \text{für } i = d_1 + 2, \ldots, n \end{cases}$$

die Valenzsequenz eines Graphen ist.

Beweis. Ist D' die Valenzsequenz eines Graphen G', so fügen wir zu G' einen neuen Knoten hinzu, der genau die Knoten mit Valenz $d'_2, \ldots, d'_{d_1+1}$ kennt und erhalten so einen Graphen G mit Valenzsequenz D. Die andere Richtung ist etwas schwerer. Die Aussage bedeutet gerade

Behauptung: Wenn es einen Graphen mit der Sequenz D gibt, so gibt es auch einen solchen, bei dem der Knoten v mit der größten Valenz genau zu den $\deg(v)$ Knoten mit den nächsthöheren Valenzen adjazent ist.

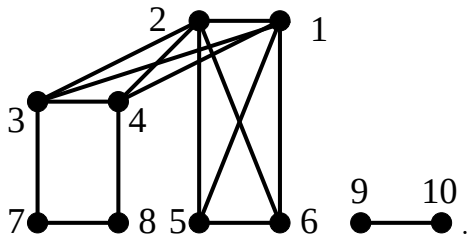
Setzen wir also voraus, es gäbe einen Graphen G mit Sequenz D. Wir wählen nun einen Graphen mit Knotenmenge $\{v_1, \ldots, v_n\}$, bei dem stets $\deg(v_i) = d_i$ ist und der maximale Index j eines zu v_1 benachbarten Knoten minimal ist. Ist $j = d_1 + 1$, so können wir v_1 mit allen Kanten entfernen und erhalten einen Graphen mit Valenzsequenz D'. Sei also $j > d_1 + 1$. Dann gibt es ein $1 < i < j$, so dass v_i den Knoten v_1 nicht kennt. Da $d_i \geq d_j$ ist und v_j v_1 kennt, v_i aber nicht, muss es auch einen Knoten v_k geben, den v_i kennt, aber v_j nicht. Wir entfernen nun aus G die Kanten (v_1v_j) und (v_i, v_k) und fügen die Kanten (v_1v_i) und (v_jv_k) hinzu und erhalten einen einfachen Graphen $\tilde{G}$ mit Valenzsequenz D, bei dem der größte Index eines Knoten, der v_1 kennt, kleiner ist als bei G, im Widerspruch zur Wahl von G. Also muss für G schon $j = d_1 + 1$ gegolten haben. □

Aus diesem Satz erhält man sofort ein Verfahren, das entscheidet, ob eine gegebene Sequenz die Valenzsequenz eines Graphen ist.

Beispiel 3.4.5 *Wir betrachten die Sequenz* $(5, 5, 4, 4, 3, 3, 2, 2, 1, 1)$*. Durch Anwenden des Satzes erhalten wir die Sequenz* $(4, 3, 3, 2, 2, 2, 2, 1, 1)$ *und dann* $(2, 2, 1, 1, 2, 2, 1, 1)$*. Diese sortieren wir wieder und erhalten*

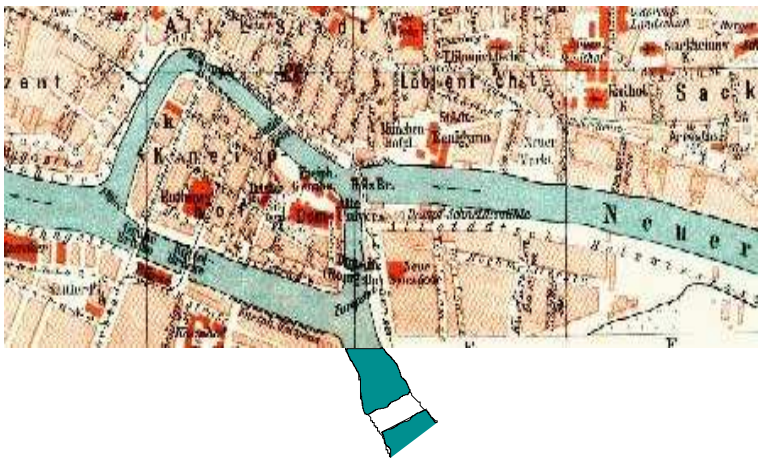
$\begin{pmatrix} 3 & 4 & 7 & 8 & 5 & 6 & 9 & 10 \\ 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 \end{pmatrix}$ *und* $\begin{pmatrix} 4 & 7 & 8 & 5 & 6 & 9 & 10 \\ 1 & 1 & 2 & 1 & 1 & 1 & 1 \end{pmatrix}$.

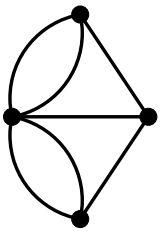
$\begin{pmatrix} 8 & 4 & 7 & 5 & 6 & 9 & 10 \\ 2 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$ *wird zu* $\begin{pmatrix} 4 & 7 & 5 & 6 & 9 & 10 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$*. Letztere Folge können wir in wenigen Schritten auf* (0)*, den* K_1 *reduzieren. Als Graphen konstruieren wir*



3.5 Eulertouren

Leonhard Euler löste 1736 folgendes Problem. Über die neue Pregel führten sieben Brücken, die die Ufer der neuen Pregel und die Inseln verbanden:





Ist es möglich bei einem sonntäglichen Spaziergang über alle sieben Brücken zu gehen, jede Brücke nur einmal zu betreten und zum Ausgangspunkt zurückzukehren, ohne zu schwimmen?

Die Fragestellung lässt sich durch den links daneben gezeichneten Multigraphen modellieren, bei dem die Fragestellung dann lautet, ob es möglich ist, den Graphen in einem Zug ohne abzusetzen zu zeichnen, wobei man wieder am Anfangspunkt endet.

Definition 3.5.1 *Sei $G = (V, E)$ ein Multigraph ohne isolierte Knoten, d.h. $G = K_1$ oder $deg(v) > 0$ für alle $v \in V$ und $v_0 \in V$. Ein Spaziergang $v_0e_1v_1e_2 \dots e_mv_0$ von v_0 nach v_0 heißt* Eulertour, *wenn er jede Kante genau einmal benutzt. Der Graph G heißt* eulersch, *wenn er eine Eulertour ausgehend von einem (und damit von jedem) Knoten $v_0 \in V$ hat.*

Die Existenz einer Eulertour lässt sich leicht erkennen.

Satz 3.5.2 *Sei $G = (V, E)$ ein Multigraph. Dann sind paarweise äquivalent.*

a) G ist eulersch.

b) G ist zusammenhängend und alle Knoten haben geraden Knotengrad.

c) G ist zusammenhängend und E ist disjunkte Vereinigung von Kreisen.

Beweis.

$a) \Rightarrow b)$ Die erste Implikation ist offensichtlich, da eine Eulertour alle Kanten genau einmal benutzt und geschlossen ist, man also in jeden Knoten genausooft ein- wie auslaufen muss.

$b) \Rightarrow c)$ Die zweite Implikation zeigen wir mittels vollständiger Induktion über die Kantenmenge. Sei also G ein zusammenhängender Graph, bei dem alle Knoten einen geraden Knotengrad haben. Ist $|E| = 0$, so ist $G = K_1$ und die leere Menge ist die disjunkte Vereinigung von Null Kreisen. Sei also $|E| > 0$. Wir starten bei einem beliebigen Knoten v_0 und wählen eine Kante $e = (v_0, v_1)$. Da $\deg(v_1)$ gerade ist, gibt es eine Kante (v_1, v_2) mit $e \neq (v_2, v_1)$. Wir fahren so fort. Da V endlich ist, muss sich irgenwann ein Knoten w zum ersten Mal wiederholen. Der Teil des Spaziergangs von w nach w ist dann geschlossen und wiederholt weder Kanten nach Knoten, bildet also einen Kreis C_1. Diesen entfernen wir. Jede Zusammenhangskomponente des resultierenden Graphen hat nur Knoten mit geradem Knotengrad, ist also nach Induktionsvoraussetzung disjunkte Vereinigung von Kreisen.

$c) \Rightarrow a)$ Sei schließlich G zusammenhängend und $E = C_1 \dot\cup \ldots \dot\cup C_k$ disjunkte Vereinigung von Kreisen. Wir gehen wieder mit Induktion, diesmal über k vor, ist $k = 0$, so ist nichts zu zeigen. Andernfalls ist jede Komponente von $G \backslash C_1$ eulersch nach Induktionsvoraussetzung. Seien die Knoten von $C_1 = v_1, \ldots, v_l$ durchnummeriert. Dann enthält jede Komponente von $G \setminus C_1$ auf Grund des Zusammenhangs von G einen Knoten von C_1 mit kleinstem Index und diese Kontaktknoten sind paarweise verschieden. Wir durchlaufen nun C_1 und, wenn wir an einen solchen Kontaktknoten kommen, durchlaufen wir die Eulertour seiner Komponente, bevor wir auf C_1 fortfahren.

□

In Königsberg ist jeder Knoten ungerade, der Multigraph also deutlich nicht eulersch.

Der Beweis von Satz 3.5.2 enthält im Prinzip einen rekursiven Algorithmus, mit dem man eine Eulertour bestimmen kann. Etwas direkter geht der Algorithmus nach Fleury vor:

Sei $v_0 \in V$. Starte in $v = v_0$ einen Kantenzug, iteriere

- So lange $E \neq \emptyset$.
- Wähle in v eine auslaufende Kante $e = (v, w)$, so dass die Kantenmenge $E \setminus e$, falls sie nicht leer ist, einen zusammenhängenden Graphen induziert, der v_0 enthält.
- Setze $E = E \setminus e$ und $v = w$.

Satz 3.5.3 *Der Algorithmus von Fleury entfernt die Kanten in der Reihenfolge einer Eulertour.*

Beweis. Wir haben zu zeigen, dass eine Kantenwahl wie angegeben stets möglich ist. Dazu verfahren wir per Induktion über den Lauf des Algorithmus. Im ersten Schritt ist eine solche Wahl offensichtlich möglich, man kann e beliebig wählen. Sei also im letzten Schritt eine Kante entfernt worden und E und v aufdatiert worden. Dann induziert E einen Graphen, dessen Kantenmenge zusammenhängend ist und bei dem höchstens v und v_0 ungeraden und alle übrigen Kanten geraden Knotengrad haben. Die restliche Kantenmenge evtl. mit der Zusatzkante vv_0 induziert einen Eulerschen Graphen. Sei $e = (v, w)$ eine Kante, die in einer Eulertour in diesem Graphen nach (bzw. vor) v_0v durchlaufen wird. Dann ist die Kantenmenge $E \setminus e$ offensichtlich zusammenhängend und enthält v_0. □

Bei der Implementierung des Fleury ist die effiziente Implementierung des Zusammenhangschecks der Flaschenhals.

Einen rekursiven Algorithmus, der die Idee aus dem Beweis von Satz 3.5.2 umsetzt, kann man leicht in $O(|E|)$ implementieren (s.z.B. [2]).

3.6 Gerichtete Graphen und Eulertouren

Wir hatten Graphen als ungerichtete einfache Graphen eingeführt. In *gerichteten Graphen* ist die Adjazenrelation nicht mehr notwendig symmetrisch. Dieses Phänomen taucht bei Einbahnstraßen oder etwa Ordnungsrelationen auf.

Definition 3.6.1 *Sei V eine endliche Menge (von* Knoten*) und $A \subseteq V \times V \setminus \Delta$ eine Teilmenge der (geordneten) Tupel über V ohne die Diagonale, d.h. ohne die Elemente der Form (v, v). Dann nennen wir das geordnete Paar (V, A) einen* gerichteten Graph *oder einen* Digraph *(genauer, einen einfachen gerichteten Graphen). Die Kanten $(v, w) \in A$ nennen wir auch* Bögen *und v den Anfang (tail) und w das Ende (head).*

Die Definitionen für Graphen lassen sich in der Regel auf Digraphen übertragen. Wir erhalten so gerichtete Pfade, Kreise oder Spaziergänge, auch sprechen wir von Multidigraphen, wenn gleichgerichtete Kanten mehrfach vorkommen dürfen. Ein gerichteter Spaziergang ist z.B. eine alternierende Folge

aus Knoten und Bögen $(v_0, a_1, v_1, a_2, \ldots, a_k, v_k)$ mit $a_i = (v_{i-1}, v_i)$. Bei Knotengraden unterscheiden wir zwischen dem *Innengrad* $\deg^+_G(v)$, der Anzahl der einlaufenden Kanten, deren Ende v ist und dem *Außengrad* $\deg^-_G(v)$. Der zugrundeliegende Multigraph eines (Multi-)Digraphen ist der Multigraph, der entsteht, wenn man die Orientierung der Bögen vergisst. Ist der zugrundeliegende Multigraph ein Graph, so heisst der Digraph eine *Orientierung* des zugrundeliegenden Graphen.

Auch den Begriff der Eulertour übernehmen wir.

Definition 3.6.2 *Sei $D = (V, A)$ ein (Multi-)Digraph. Ein geschlossener Spaziergang, der jeden Bogen genau einmal benutzt, heißt* Eulertour*. Ein (Multi-)Digraph heißt* eulersch*, wenn er eine Eulertour hat.*

Und wie im ungerichteten Fall zeigt man:

Satz 3.6.3 *Sei $D = (V, A)$ ein (Multi-)Digraph. Dann sind paarweise äquivalent.*

a) D ist eulersch.

b) D ist zusammenhängend und alle Knoten haben gleichen Innen- wie Außengrad.

c) D ist zusammenhängend und A ist disjunkte Vereinigung von gerichteten Kreisen.

Beweis. Selber analog zu Satz 3.5.2. □

Beispiel 3.6.4 (Das Rotating Drum Problem nach Good 1946) *In einer rotierenden Trommel wird die Position durch jeweils einen String aus k Nullen und Einsen bestimmt. Wieviele Stellungen kann man auf diese Art und Weise unterscheiden? Genauer: Wie lang kann ein binärer (aus Nullen und Einsen) String sein, bei dem alle Teilstrings der Länge k paarweise verschieden sind.*

Wir betrachten den Digraphen, bei dem die Knoten alle 01-Strings der Länge $k-1$ sind, $V = \{0,1\}^{k-1}$. Wir haben einen Bogen a von dem Knoten $v = b_1 b_2 \ldots b_{k-1}$ zum Knoten $w = a_1 a_2 \ldots a_{k-1}$, wenn $b_i = a_{i-1}$ für $k = 2, \ldots, k-1$, also w aus v durch Streichen des ersten Bits und Anhängen eines weiteren entsteht. Wir können a mit der Bitfolge $b_1 b_2 \ldots b_{k-1} a_{k-1}$ identifizieren. Die Kantenmenge entspricht dann genau den binären Wörtern der Länge k. Dieser Multidigraph heißt deBruijn Graph*.*

Die obige Aufgabenstellung ist dann gleichbedeutend damit, dass wir in diesem Graphen einen möglichst langen, kantenwiederholungsfreien, geschlossenen Spaziergang suchen. Wie sieht dieser Graph aus? Er hat 2^{k-1} Knoten und in jeden Knoten führen genau zwei Kanten hinein und genau zwei wieder heraus. Also ist der Digraph eulersch und aus einer Eulertour konstruieren wir ein zyklisches Wort der Länge 2^k.

Betrachten wir den Fall $k = 4$, erhalten wir den Graphen in Abbildung 3.3 und als zyklischen String z.B.

0000111101100101.

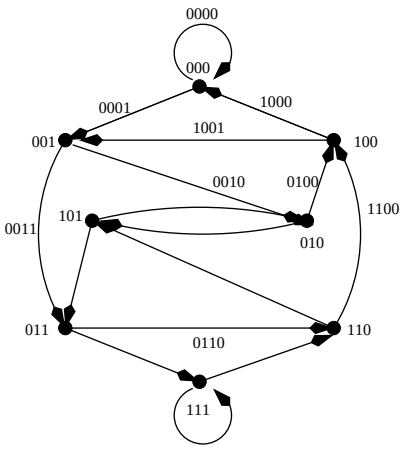


Abbildung 3.3: Der deBruijn Graph für $k = 4$

3.7 Zweizusammenhang

Definition 3.7.1 *Sei $G = (V, E)$ ein Graph. Wir sagen G ist k-fach* knotenzusammenhängend*, wenn $|V| \geq k + 1$ ist und der Graph nach Entfernen beliebiger $k - 1$ Knoten immer noch zusammenhängend ist. Wir sagen G ist k-fach* kantenzusammenhängend*, wenn er nach Entfernen beliebiger $k - 1$ Kanten immer noch zusammenhängend ist. Die größte natürliche Zahl, für die G knoten- bzw. kantenzusammenhängend ist, heißt* Knoten- *bzw.* Kantenzusammenhangszahl *$\kappa(G)$ bzw. $\kappa'(G)$.*

Für diese Definition haben wir Operationen auf Graphen benutzt, die noch nicht definiert sind. Das wollen wir nun nachholen.

Definition 3.7.2 *Sei $G = (V, E)$ ein Graph. Wir definieren folgende Graphen, die durch Operationen auf G entstehen.*

Entfernen einer Kante $e \in E$**:** *Der Graph $G \setminus e$ ist der Graph $G \setminus e := (V, E \setminus \{e\})$.*

Einfügen einer Kante $\bar{e} \in \binom{V}{2} \setminus E$**:** *Der Graph $G + \bar{e}$ ist der Graph $G + \bar{e} := (V, E \cup \{\bar{e}\})$.*

Entfernen eines Knotens $v \in V$**:** *Der Graph $G \setminus v$ ist der Graph $G \setminus v = (V \setminus v, \tilde{E})$ mit $\tilde{E} := \{e \in E \mid v \notin e\}$.*

Unterteilen einer Kante $e \in E$**:** *Die Unterteilung $G\%e$ mit $e = (v, w)$ ist der Graph $G\%e := (V \cup u, \hat{E})$, wobei $u \notin V$ ein neuer Knoten sei und $\hat{E} := (E \setminus \{e\}) \cup \{(v, u), (u, w)\}$.*

Kontraktion einer Kante $e \in E$**:** *Die Kontraktion von e G/e mit $e = (v, w)$ ist der Multigraph $G/e = (\tilde{V}, \tilde{E})$ mit $\tilde{V} := V \cup \{u\} \setminus \{v, w\}$, wobei $u \notin V$ ein neuer Knoten sei und $\tilde{E} := \{e \in E \mid e \cap \{v, w\} = \emptyset\} \cup \{(u, x) \mid (v, x) \in E\} \cup \{(y, u) \mid (y, w) \in E\}$.*

Alle diese Operationen sind assoziativ und kommutativ (sofern sie miteinander verträglich sind). Also kann man auch Knotenmengen löschen oder Kantenmengen kontrahieren oder löschen. Ein Graph der durch sukzessives Unterteilen von Kanten ausgehend von G entsteht, heißt Unterteilung von *G.*

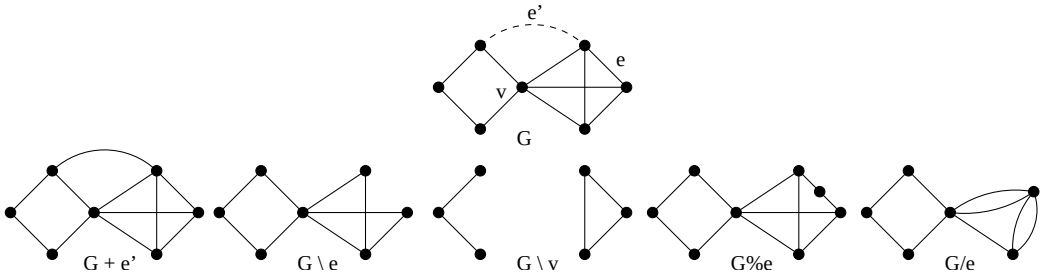
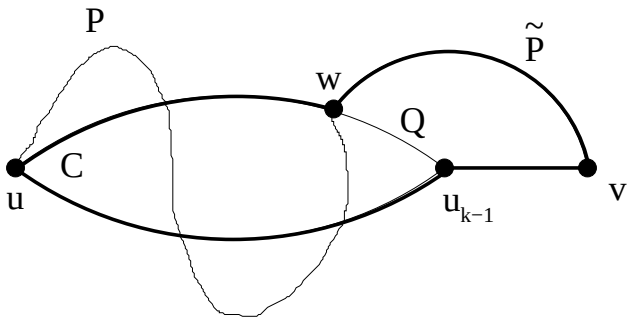


Abbildung 3.4: Operationen auf Graphen

Spezialisieren wir nun den Zusammenhangsbegriff für $k = 2$, so ist ein Graph 2-knotenzusammenhängend (oder kurz zweizusammenhängend), wenn man seinen Zusammenhang nicht durch Entfernen eines Knotens zerstören kann. Dafür können wir zeigen:

Satz 3.7.3 *Ein Graph ist genau dann 2-knotenzusammenhängend, wenn je zwei Knoten $u \neq v$ auf einem gemeinsamen Kreis liegen.*

Beweis. Liegen je zwei Knoten auf einem Kreis, so kann man den Zusammenhang des Graphen sicherlich nicht durch Entfernen eines einzelnen Knoten zerstören. Die andere Implikation zeigen wir mittels vollständiger Induktion über $dist(u, v)$. Ist $dist(u, v) = 1$, so gibt es eine Kante $e = (u, v) \in E$. In dem Graphen $G \setminus e$ muss es nun immer noch einen Weg P von u nach v geben, da ansonsten mindestens einer von $G \setminus v$ und $G \setminus u$ unzusammenhängend wäre. Dieser Weg zusammen mit e bildet den gesuchten Kreis. Sei also nun $dist(u, v) \geq 2$ und $u = u_0 u_1 \dots u_{k-1} u_k = v$ ein kürzester Weg von u nach v. Dann liegen nach Induktionsvoraussetzung u und u_{k-1} auf einem gemeinsamen Kreis C. Liegt v auch auf diesem Kreis, so sind wir fertig. Sei also $v \notin C$. Da $G \setminus u_{k-1}$ zusammenhängend ist, gibt es darin immer noch einen Weg von u nach v. Sei $w \notin \{u_{k-1}, v\}$ der letzte Knoten auf diesem Weg, der zu C gehört und sei $\tilde{P}$ der Teilweg von P von w nach v. Sei Q der Weg von w nach u_{k-1} auf C, der nicht über u führt. Dann ist $C \setminus Q \cup \tilde{P} \cup \{(v, u_{k-1})\}$ ein Kreis, der u und v enthält. □



Bemerkung 3.7.4 *Satz 3.7.3 ist ein Spezialfall des* Satzes von Menger*, der besagt, dass ein Graph genau dann k-knotenzusammenhängend ist, wenn es zu je zwei Knoten u, v k uv-Wege gibt, die paarweise nur die Endknoten gemeinsam haben.*

Aus dieser Charakterisierung schließen wir

Korollar 3.7.5 *Ein Graph $G = (V, E)$ ist genau dann 2-zusammenhängend, wenn jede Unterteilung von G 2-zusammenhängend ist.*

Beweis. Es genügt, die Behauptung für $G\%e$ und eine Kante $e = (v, w) \in E$ zu beweisen. Wenn je zwei Knoten von $G\%e$ auf einem gemeinsamen Kreis

liegen, gilt dies sicherlich auch für G. Für die andere Implikation gehen wir mit der Definition vor. Ist $x \in V$ ein „Originalknoten“ von G verschieden von v und w, so ist $G\%e \setminus x = (G \setminus x)\%e$ zusammenhängend. Gleiches gilt auch, wenn wir v oder w entfernen, dann haben wir zusätzlich an den anderen Knoten eine Kante angehängt. Ist u der Unterteilungsknoten, so ist $G\%e \setminus u = G \setminus e$. Von letzterem haben wir im Beweis von Satz 3.7.3 gezeigt, dass er zusammenhängend ist. □

Oft wird von zweizusammenhängenden Graphen eine konstruktive Eigenschaft genutzt. Sie haben eine *Ohrenzerlegung*.

Definition 3.7.6 *Sei $G = (V, E)$ ein Graph. Eine Folge $(C_0, P_1, P_2, \ldots, P_k)$ heißt* Ohrenzerlegung von G*, wenn*

- *C_0 ein Kreis ist,*
- *für alle $i = 1, \ldots, k$ P_i ein Pfad ist, der mit $V(C_0) \cup \bigcup_{j=1}^{i-1} V(P_j)$ genau seinen Anfangs- und Endknoten gemeinsam hat,*
- *$E(C_0), E(P_1), \ldots, E(P_k)$ eine Partition der Kantenmenge E bildet.*

Satz 3.7.7 *Ein Graph ist genau dann 2-zusammenhängend, wenn er eine Ohrenzerlegung hat.*

Beweis. Habe der Graph zunächst eine Ohrenzerlegung $C_0, P_1, \ldots, P_k$. Wir zeigen per Induktion über k, dass G 2-zusammenhängend ist. Ist $k = 0$, so liegen offensichtlich je zwei Knoten auf einem gemeinsamen Kreis, sei also $k > 0$. Nach Induktionsvoraussetzung ist dann der Graph $\tilde{G}$, der aus $C_0, P_1, \ldots, P_{k-1}$ gebildet wird, 2-sammenhängend. Sei P_k ein vw-Weg. Ist $G = \tilde{G} + (vw)$, so ist G sicherlich auch 2-zusammenhängend. Andernfalls entsteht der Graph G aus $\tilde{G}$, indem entweder zunächst die Kante (v, w) hinzugefügt und dann (evtl. mehrfach) unterteilt wird oder, weil $(vw) \in \tilde{G}$ zunächst diese Kante unterteilt wird und dann (vw) wieder hinzugefügt wird. Beim Addieren einer Kante bleibt der 2-Zusammenhang erhalten und beim Unterteilen nach Korollar 3.7.5.

Sei nun G 2-zusammenhängend. Wir definieren die Ohrenzerlegung induktiv. Sei zunächst C_0 ein beliebiger Kreis in G. Wir nehmen nun an, es seien die Ohren $C_0, P_1, \ldots, P_i$ definiert. Ist $E = E(C_0) \cup \bigcup_{j=1}^{i} E(P_j)$, so sind wir fertig. Andernfalls gibt es, da G zusammenhängend ist, eine Kante $e \in E \setminus \left(E(C_0) \cup \bigcup_{j=1}^{i} E(P_j)\right)$ mit $e = (v, w) \cap V_i \neq \emptyset$, wobei $V_i :=$

$V(C_0) \cup \bigcup_{j=1}^{i} V(P_j)$. Sei v ein Knoten aus diesem Schnitt. Liegt auch w im Schnitt, so setzen wir $P_{i+1} = e$, andernfalls gibt es, da $G \setminus v$ zusammenhängend ist, zu jedem Knoten $x \in V_i \setminus \{v\}$ einen Weg von w nach x. Sei ein solcher Weg P so gewählt, dass er außer x keinen weiteren Knoten in V_i enthält. Wir verlängern P um e zu einem vx-Weg, der unser neues Ohr P_{i+1} ist. □

Der Beweis verdeutlicht auch, dass man jeden 2-zusammenhängenden Graphen aus dem C_3 durch sukzessive Hinzunahme von Kanten zwischen existierenden Knoten und Unterteilung von Kanten erhalten kann.

Kapitel 4

Bäume

4.1 Definition und Charakterisierungen

Definition 4.1.1 *Ein zusammenhängender Graph $T = (V, E)$, der keinen Kreis enthält, heißt* Baum.

Bäume werden durch die folgenden Eigenschaften charakterisiert.

Satz 4.1.2 *Sei $T = (V, E)$ ein Graph und $|V| \geq 2$.Dann sind paarweise äquivalent:*

a) T ist ein Baum.

b) Zwischen je zwei Knoten $v, w \in V$ gibt es genau einen Weg von v nach w.

c) T ist zusammenhängend und für alle $e \in E$ ist $T \backslash e$ unzusammenhängend.

d) T ist kreisfrei und für alle $\bar{e} \in \binom{V}{2} \setminus E$ enthält $T + \bar{e}$ einen Kreis.

e) T ist zusammenhängend und $|E| = |V| - 1$.

f) T ist kreisfrei und $|E| = |V| - 1$.

Um den Beweis mittels Induktion über die Knotenzahl führen zu können, zeigen wir zunächst, dass jeder Baum ein Blatt, d.i. ein Knoten v mit $\deg(v) = 1$, hat. Genauer gilt:

Lemma 4.1.3 *Jeder Baum mit mindestens zwei Knoten hat mindestens zwei Blätter.*

Beweis. Da der Baum zusammenhängend ist und mindestens zwei Knoten hat, enthält er Wege der Länge mindestens 1. Sei $P = (v_1, \ldots, v_k)$ ein möglichst langer Weg in T. Da T kreisfrei ist, ist v_1 zu keinem von $v_3, \ldots, v_k$ adjazent. Dann muss $\deg(v_1)$ aber schon 1 sein, da man ansonsten P_k verlängern könnte. Die gleiche Argumentation gilt für v_k. □

Lemma 4.1.4 *Sei $G = (V, E)$ ein Graph und v ein Blatt in G. Dann ist G ein Baum genau dann, wenn $G \setminus v$ ein Baum ist.*

Beweis. Sei G ein Baum und v ein Blatt von G. Dann enthält kein Weg in G den Knoten v als inneren Knoten. Also ist $G \setminus v$ immer noch zusammenhängend und gewiss weiterhin kreisfrei.

Sei umgekehrt nun vorausgesetzt, dass $G \setminus v$ ein Baum ist. Da v ein Blatt ist, hat es einen Nachbarn u, von dem aus man in $G \setminus v$ alle Knoten erreichen kann, also ist G zusammenhängend. Offensichtlich kann v auf keinem Kreis liegen. □

So gerüstet schreiten wir zum Beweis des Satzes.

Beweis von Satz 4.1.2. Die Äquivalenz gilt offensichtlich für alle Graphen mit genau zwei Knoten. In diesem Falle sind alle Graphen, die eine der äquivalenten Bedingungen erfüllen, isomorph zum K_2, wobei Punkt d) gilt, da keine solche Kante existiert. Wir fahren fort per Induktion und nehmen an, dass $|V| \geq 3$ und die Gültigkeit der Äquivalenz für Graphen mit höchsten $|V| - 1$ Knoten bewiesen sei.

a) $\Rightarrow$ *b)* Seien $x, y \in V$. Ist x oder y ein Blatt, o.E. x, so sei w der Nachbar von x. Nach Induktionsvoraussetzung und Lemma 4.1.4 gibt es in $T \setminus x$ genau einen Weg von y nach w. Diesen können wir zu einem Weg von y nach x verlängern. Umgekehrt setzt sich jeder Weg von x nach y aus der Kante (xw) und einem wy-Weg in $T \setminus x$ zusammen. Also gibt es auch höchstens einen xy-Weg in T. Sind beide kein Blatt, so folgt die Behauptung per Induktion, wenn wir ein beliebiges Blatt entfernen.

b) $\Rightarrow$ *c)* Wenn es zwischen je zwei Knoten einen Weg gibt, ist der Graph zusammenhängend. Sei $e = (v, w) \in E$. Gäbe es in $T \setminus e$ einen (vw)-Weg, dann gäbe es in T deren zwei, da e schon einen vw-Weg bildet. Also muss $T \setminus e$ unzusammenhängend sein.

$c) \Rightarrow d)$ Wenn es in T einen Kreis gibt, so kann man jede beliebige Kante dieses Kreises entfernen, ohne den Zusammenhang zu zerstören, da diese Kante in jedem Spaziergang durch den Rest des Kreises ersetzt werden kann. Die Aussage in $c)$ verbietet also die Existenz eines Kreises. Sei $\bar{e} = (v, w) \in \binom{V}{2} \setminus E$. Da T zusammenhängend ist, gibt es in T einen vw-Weg, der mit der Kante $\bar{e}$ einen Kreis in $T + \bar{e}$ bildet.

$d) \Rightarrow a)$ Wenn T nicht zusammenhängend ist, so kann man eine Kante zwischen zwei Komponenten einfügen, ohne einen Kreis zu erzeugen. Also muss T zusammenhängend sein. Die Kreisfreiheit ist sowieso voraussgesetzt.

$a) \Leftrightarrow e)$ Sei v ein Blatt im Baum T. Nach Induktionsvoraussetzung und Lemma 4.1.4 ist $|E| = |E(T \setminus v)| + 1 = |V(T \setminus v)| - 1 + 1 = |V| - 1$. Ist umgekehrt vorausgesetzt, dass $|E| = |V| - 1$ und T zusammenhängend ist, so haben wir $\deg(v) \geq 1$ für alle $v \in V$ und nach dem Handshake Lemma $\sum_{v \in V} \deg(v) = 2|V| - 2$. Also muss es einen Knoten mit $\deg(v) = 1$, also ein Blatt in T geben. Entfernen wir dieses Blatt, so folgt die Behauptung aus der Induktionsvoraussetzung und Lemma 4.1.4.

$a) \Leftrightarrow f)$ Der Beweis verläuft analog zu dem soeben Gezeigten, wenn man ausnutzt, dass ein Graph mit mindestens einer Kante aber ohne Blatt stets einen Kreis hat.

□

4.2 Isomorphismen von Bäumen

Im Gegensatz zu der Situation bei allgemeinen Graphen kann man bei Bäumen (und einigen anderen speziellen Graphenklassen) die Isomorphie zweier Graphen effizient testen. Wir stellen einen Algorithmus vor, der zu jedem Baum mit n Knoten eine 2n-stellige Binärzahl berechnet, die wir als den *Code* des Graphen bezeichnen. Dieser Code zweier Bäume ist gleich genau dann, wenn die Bäume isomorph sind.

Zunächst ist folgendes Konzept hilfreich, das wir implizit schon bei der Breitensuche kennengelernt haben.

Definition 4.2.1 *Ein* Wurzelbaum *oder eine* Arboreszenz *ist ein Paar* (T, r) *bestehend aus einem Baum* T *und einem* Wurzelknoten r. *Wir denken uns*

nun alle Kanten des Baumes so orientiert, dass die Wege von r zu allen anderen Knoten v gerichtete Wege sind. Ist dann (v, w) ein Bogen, so sagen wir v ist Vater *von w und w ist* Sohn *oder* direkter Nachfahre *von v.*

Ein gepflanzter Baum *(T, r, ν) ist ein Wurzelbaum, bei dem an jedem Knoten eine Reihenfolge $\nu(v)$ der direkten Nachkommen vorgegeben ist. Dadurch ist eine „Zeichenvorschrift" definiert, wie wir den Graphen in die Ebene einzubetten haben.*

Für jeden dieser Bäume mit zusätzlicher Struktur kann man Isomorphismen definieren. Ein Isomorphismus zweier Wurzelbäume $(T, r), (T', r')$ ist ein Isomorphismus von T und T', bei dem r auf r' abgebildet wird. Ein Isomorphismus gepflanzter Bäume ist ein Isomorphismus der Wurzelbäume, bei dem zusätzlich die Reihenfolge der Söhne berücksichtigt wird. Die Bäume in Abbildung 4.1 sind isomorph als Bäume, die beiden rechten sind isomorph als Wurzelbäume und keine zwei sind isomorph als gepflanzte Bäume.

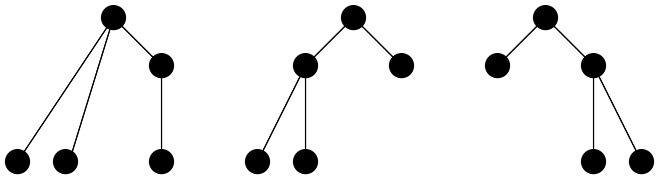


Abbildung 4.1: Gepflanzte Bäume

Wir gehen nun in drei Schritten vor.

a) Zu einem gegebenen Baum bestimmen wir zunächst eine Wurzel.

b) Zu einem Wurzelbaum bestimmen wir eine kanonische Pflanzung.

c) Zu einem gepflanzten Baum bestimmen wir einen eindeutigen Code.

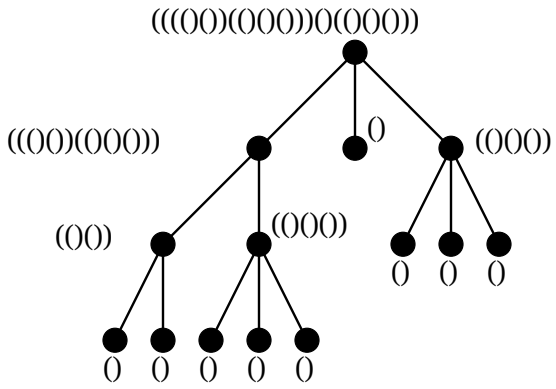
Da sich der erste und der zweite Schritt leichter darstellen lassen, wenn der dritte bekannt ist, stellen wir dieses Verfahren von hinten nach vorne vor.

Sei also (T, r, ν) ein gepflanzter Baum. Wir definieren den Code „Bottom-Up" für jeden Knoten, indem wir ihn zunächst für Blätter erklären und dann für gepflanzte Bäume, bei denen alle Knoten außer der Wurzel schon einen Code haben. Dabei identifizieren wir den Code eines Knoten x mit dem Code des gepflanzten Baumes, der durch den Ausgangsbaum auf x und allen seinen Nachfolgern induziert wird. Um die Idee des Codes besser zu verstehen, verwenden wir statt 01 die Symbole ().

- Alle Blätter haben den Code ().
- Ist x ein Knoten mit Söhnen $y_1, \ldots y_k$, deren Codes $C_1, \ldots, C_k$ sind, so erhält x den Code $(C_1 C_2 \ldots C_k)$.

Dann entsprechen die von Knoten auf ihrer Nachfolgermenge induzierten Graphen genau den wohlgeklammerten Ausdrücken, also den Ausdrücken, die gleich viele öffnende wie schließende Klammern haben, die mit einer öffnenden Klammer beginnen, welche erst mit der letzten Klammer geschlossen wird. Die Codes der Söhne von x findet man also wie folgt:

- Streiche die erste (öffnende) Klammer.
- Solange das nächste Zeichen eine öffnende Klammer ist
  - Suche die entsprechende schließende Klammer, schreibe die Zeichenkette bis hierhin raus und lösche sie.



In dem Beispiel in der Abbildung erhalten wir als Codes der Söhne der Wurzel auf diese Weise völlig zu Recht $C_1 = (((0)(0))((0)(0)(0)))$, $C_2 = ()$, $C_3 = ((0)(0))$.

Eine anschauliche Interpretation des Codes eines gepflanzten Baumes erhält man, wenn man den geschlossenen Weg betrachtet, der an der Wurzel mit der Kante nach links unten beginnt und dann außen um den Baum herumfährt. Jedesmal, wenn wir eine Kante abwärts fahren, schreiben wir eine öffnende Klammer und eine schließende Klammer und wenn wir eine Kante aufwärts fahren. Schließlich machen wir um den ganzen Ausdruck noch ein Klammerpaar für die Wurzel.

Offensichtlich können wir so den gepflanzten Baum (bis auf Isomorphie) rekonstruieren. Nicht isomorphe gepflanzte Bäume haben also verschiedene Codes. Umgekehrt bleibt der Code eines gepflanzten Baumes unter einem Isomorphismus offensichtlich invariant, also haben isomorphe gepflanzte Bäume den gleichen Code.

Wir übertragen diesen Code nun auf Wurzelbäume, indem wir die Vorschrift wie folgt modifizieren

- Alle Blätter haben den Code ().
- Ist x ein Knoten mit Söhnen, deren Codes bekannt sind, so sortiere die Söhne so zu $y_1, \ldots y_k$, dass für die zugehörigen Codes gilt $C_1 \geq C_2 \geq \ldots \geq C_k$, und x erhält dann den Code $(C_1 C_2 \ldots C_k)$.

Dabei bedeutet $A \leq B$ irgendeine Totalordnung auf den endlichen Klammerstrings. Üblich ist die lexikographische Ordnung (vgl. Telephonbücher und Lexika). Hierbei gehen wir davon aus, dass die öffnende Klammer kleiner als die schließende Klammer ist, und wenn ein String A ein echtes Anfangsstück von B ist, so ist $A < B$, also z.B. $(()()) < (()())($, dieser Fall kann aber bei uns nicht auftreten, da die Ausdrücke wohlgeklammert sind. Ansonsten

- sei j der kleinste Index mit $a_j \neq b_j$. Dann ist $A < B \Leftrightarrow a_j < b_j$, ansonsten $B < A$.

Diese Vereinbarung definiert auf den Knoten eine Reihenfolge der Söhne, macht also auf eindeutige Weise aus einem Wurzelbaum einen gepflanzten Baum. Offensichtlich bekommen so isomorphe Wurzelbäume den gleichen Code.

Kommen wir nun zu den Bäumen. Wir versuchen zunächst von einem gegebenen Baum einen Knoten zu finden, der sich als Wurzel aufdrängt und unter Isomorphismen fix bleibt. Ein solcher Knoten soll in der Mitte des Baumes liegen. Das zugehörige Konzept ist auch auf allgemeinen Graphen sinnvoll.

Definition 4.2.2 *Sei $G = (V, E)$ ein Graph und $v \in V$. Als* Exzentrizität *$ex_G(v)$ bezeichnen wir die Zahl*

$$ex_G(v) = \max\{dist_G(v, w) \mid w \in V\} \tag{4.1}$$

also den größten Abstand zu einem anderen Knoten.

Das Zentrum $Z(G)$ ist die Menge der Knoten minimaler Exzentrizität

$$Z(G) = \{v \in V \mid ex_G(v) = \min\{ex_G(w) \mid w \in V\}\}\,. \tag{4.2}$$

Ist das Zentrum unseres Baumes ein Knoten, so wählen wir diesen als Wurzel. Ansonsten nutzen wir aus, dass

Lemma 4.2.3 *Sei $T = (V, E)$ ein Baum. Dann ist $|Z(G)| \leq 2$. Ist $Z(G) = \{x, y\}$, so ist $(x, y) \in E$.*

Beweis. Wir beweisen dies mittels vollständiger Induktion über $|V|$. Die Aussage ist sicherlich richtig für Bäume mit einem oder zwei Knoten. Ansonsten entfernen wir alle Knoten, die in T ein Blatt sind, und erhalten einen nicht leeren Baum T' auf einer Knotenmenge $V' \subset V$, die echt kleiner geworden ist. Offensichtlich ist $Z(G) \subseteq V'$, und für alle Knoten $w \in V'$ gilt: $ex_{T'}(w) = ex_T(w) - 1$, insbesondere also $Z(T) = Z(T')$ und dieses hat nach Induktionsvoraussetzung höchstens zwei Elemente, die im Falle gleich zwei adjazent sind. □

Besteht das Zentrum aus einer Kante (x, y), entfernen wir diese, bestimmen die Codes der in x bzw. y gewurzelten Teilbäume und wählen den Knoten als Wurzel des Teilbaums, der einen kleineren String liefert. Wir fassen zusammen:

- Ist $Z(G) = \{v\}$, so ist ist der Code von T der Code von (T, v).
- Ist $Z(G) = \{x_1, x_2\}$, so seien die Komponenten von $T \setminus e$ $x_1 \in T_1$ und $x_2 \in T_2$. Sei C_i der Code des Wurzelbaumes (T_i, x_i) und die Namen so gewählt, dass $C_1 \leq C_2$. Dann ist der Code von T der Code des Wurzelbaumes (T, x_1).

Satz 4.2.4 *Zwei Bäume haben genau dann den gleichen Code, wenn sie isomorph sind.*

Beweis. Sind zwei Bäume nicht isomorph, so sind auch alle zugehörigen gepflanzten Bäume nicht isomorph, also die Codes verschieden. Sind die Bäume isomorph, so wird unter jedem Isomorphismus das Zentrum auf das Zentrum abgebildet. Ist das Zentrum ein Knoten, so folgt die Behauptung aus der Eindeutigkeit des Codes für gewurzelte Bäume. Ansonsten ist die Wahl der

Wurzel nur dann nicht eindeutig, wenn die Teilbäume (T_1, x_1) und (T_2, x_2) den gleichen Code haben. Dann sind diese beiden Wurzelbäume aber isomorph. □

Wir haben hier implizit ein Induktionsargument benutzt. Sie sollten in der Lage sein, dieses explizit zu formulieren und die Verankerung zu verifizieren.

4.3 Aufspannende Bäume

In diesem Abschnitt werden wir unter anderem den fehlenden Teil des Beweises, dass der BFS die Komponenten eines Graphen berechnet, nachholen.

Definition 4.3.1 *Ein kreisfreier Graph heißt* Wald. *Sei $G = (V, E)$ ein Graph und $T = (V, F)$ ein kreisfreier Teilgraph, der die gleichen Zusammenhangskomponenten wie V hat. Dann heißt T ein G* aufspannender Wald. *Sind G und damit auch T zusammenhängend, so heißt T ein G* aufspannender Baum.

Wir analysieren nun zwei schnelle Algorithmen, die in einem zusammenhängenden Graphen einen aufspannenden Baum berechnen.

Algorithmus 4.3.2 *Sei E eine (beliebig sortierte) Liste der Kanten des Graphen (V, E) und zu Anfang $T = \emptyset$. Die Funktion* AddEdge(e) *füge zu T die Kante e hinzu.*

for *e* **in** *E:*
  if not *CreatingCycle(e):*
    AddEdge(e)

Lemma 4.3.3 *Algorithmus 4.3.2 berechnet einen G aufspannenden Wald.*

Beweis. Offensichtlich berechnet der Algorithmus eine kreisfreie Menge, also einen Wald T. Wir haben zu zeigen, dass zwischen zwei Knoten u, v genau dann ein Weg in T existiert, wenn er in G existiert. Eine Implikation ist trivial. Sei also $u = v_0, v_1, \ldots, v_k = v$ ein uv-Weg in G. Angenommen u und v lägen in unterschiedlichen Komponenten von T. Sei dann v_i der letzte Knoten auf diesem Weg, der in T in der gleichen Komponente wie u liegt.

Dann ist $e = (v_i v_{i+1}) \in E \setminus T$. Als e im Algorithmus abgearbeitet wurde, schloss e folglich mit T einen Kreis. Folglich gibt es in T einen Weg vo v_i nach v_{i-1} im Widerspruch dazu, dass sie in verschiedenen Komponenten liegen. □

Betrachten wir die Komplexität des Algorithmus, so hängt diese von einer effizienten Implementierung des Kreistests ab. Eine triviale Implementierung dieser Subroutine in $O(|V||)$ ist offensichtlich, führt aber zu einer Gesamtlaufzeit von $O(|V||E|)$. Um effizienter zu werden müssen wir also folgendes Problem schneller lösen.

Problem 4.3.4 (UNION-FIND) *Sei $V = \{1, \ldots, n\}$ und eine initiale Partition in n triviale einelementige Klassen gegeben. Wie sieht eine geeignete Datenstruktur aus, so dass man folgende Operationen effizient auf einer gegebenen Partition ausführen kann?*

UNION *Gegeben seien x, y aus verschiedenen Klassen, vereinige diese Klassen.*

FIND *Gegeben seien $x, y \in V$. Stelle fest, ob x und y in der gleichen Klasse liegen.*

Für unseren Algorithmus zur Berechnung eines aufspannenden Waldes benötigen wir $|E|$ FIND und höchstens $|V| - 1$ UNION Operationen. Wir stellen eine einfache Lösung dieses Problems vor. Jede Klasse hat eine Nummer, jeder Knoten die Nummer seiner Klasse. In einem Array speichern wir einen Zeiger auf die Klasse jedes Knoten, die als Liste organisiert ist. Die Klasse hat zusätzlich ein Feld, in dem die Anzahl der Elemente der Klasse eingetragen ist, bei einer nicht mehr existierenden Nummer ist der Eintrag 0. Für eine FIND-Operation benötigen wir dann nur einen Vergleich der Zahlen, also konstante Zeit. Bei einer UNION-Operation erbt die kleinere Komponente die Nummer der größeren, wir datieren die Nummern der Knoten in der kleineren Komponente auf und verschmelzen die Listen.

Lemma 4.3.5 *Die Gesamtkosten des Kompomonentenverschmelzens betragen $O(|V| \log |V|)$.*

Beweis. Wir beweisen dies mit vollständiger Induktion über $n = |V|$. Für $n = 1$ ist nichts zu zeigen. Verschmelzen wir zwei Komponenten T_1 und T_2 der Größe $n_1 \leq n_2$ mit $n = n_1 + n_2$. Dann ist $n_1 \leq \frac{n}{2}$ und das Update kostet cn_1.

Addieren wir dies zu den Kosten für das Verschmelzen der einzelnen Knoten zu T_1 und T_2, die nach Induktionsvoraussetzung bekannt sind, erhalten wir

$$\begin{aligned} cn_1 + cn_1 \log n_1 + cn_2 \log n_2 &\leq cn_1 + cn_1 \log \frac{n}{2} + cn_2 \log n \\ &= cn_1 + cn_1(\log n - 1) + cn_2 \log n \\ &= cn \log n. \end{aligned}$$

□

Die beste bekannte UNION-FIND Struktur geht auf R. Tarjan zurück und benötigt Gesamtkosten von $O(\alpha(|V|)|V| + |E|)$, wobei α die *Inverse der Ackermann-Funktion* ist, eine Funktion, die zwar gegen Unendlich wächst aber viel langsamer als $\log n, \log \log n$ etc.

Unser zweiter Algorithmus ist ähnlich zu einer allgemeineren Version des Breadth-First-Search Algorithmus. Aufgrund dieser Allgemeinheit beschreiben wir ihn nur verbal.

Algorithmus 4.3.6 *Sei $v \in V$.*

- *Setze $V_0 = \{v\}, T_0 = \emptyset, i = 0$*
- *Solange es geht*
  - *Wähle eine Kante $e = (x, y) \in E$ mit $x \in V_i, \not\ni y$ und setze $V_{i+1} = V_i \cup \{y\}, T_{i+1} = T_i \cup \{e\}, i = i + 1$.*

Lemma 4.3.7 *Wenn der Algorithmus 4.3.6 endet, dann ist $T = T_i$ aufspannender Baum der Komponente von G, die v enthält.*

Beweis. Die Kantenmenge T ist offensichtlich zusammenhängend und kreisfrei und verbindet alle Knoten in V_i. Nach Konstruktion gibt es keine Kante mehr, die einen Knoten aus V_i mit einem weiteren Knoten verbindet. □

Eine Möglichkeit, diesen Algorithmus zu implementieren, haben wir mit dem BFS kennengelernt. Der dort angegebene Algorithmus startet allerdings zusätzlich in jedem Knoten und überprüft, ob dieser in einer neuen Komponente liegt. Damit haben wir also jetzt den fehlenden Teil vom Beweis von Satz 3.3.1 nachgeholt.

4.4 Minimale aufspannende Bäume

Wir wollen nun ein kostengünstigstes, zusammenhängendes Netzwerk bestimmen. Uns sind die Kosten der Verbindung zweier Nachbarn im Netzwerk bekannt, und wir wollen jeden Knoten von jedem aus erreichbar machen und die Gesamtkosten minimieren.

Problem 4.4.1 *Sei $G = (V, E)$ ein zusammenhängender Graph und $w : E \to \mathbb{N}$ eine nichtnegative Kantengewichtsfunktion. Bestimme einen zusammenhängenden, aufspannenden (d.h. der alle Knoten von G enthält) Teilgraphen $T = (V, F)$ so dass*

$$w(F) := \sum_{e \in F} w(e) \tag{4.3}$$

minimal ist.

Bemerkung 4.4.2 *In diesem Falle macht es auch mathematisch keinen wesentlichen Unterschied, ob die Gewichtsfunktion ganzzahlig oder reell ist. Da wir im Computer mit beschränkter Stellenzahl rechnen, können wir im praktischen Betrieb sowieso nur mit rationalen Gewichtsfunktionen umgehen. Multiplizieren wir diese mit dem Hauptnenner, ändern wir nichts am Verhältnis der Kosten, insbesondere bleiben Optimallösungen optimal. Also können wir in der Praxis o.E. stets von ganzzahligen Daten ausgehen.*

Da die Gewichtsfunktion nicht-negativ ist, können wir, falls eine Lösung einen Kreis enthält, aus diesem so lange Kanten entfernen, bis diese kreisfrei ist, ohne höhere Kosten zu verursachen. Also können wir uns auf folgendes Problem zurückziehen:

Problem 4.4.3 (Minimaler aufspannender Baum (MST)) *Sei $G = (V, E)$ ein zusammenhängender Graph und $w : E \to \mathbb{N}$ eine nichtnegative Kantengewichtsfunktion. Bestimme aufspannenden Baum $T = (V, F)$ minimalen Gewichts $w(F)$.*

Ein Graph kann sehr viele aufspannende Bäume haben, wie wir in Kapitel 7 sehen werden. Eine vollständige Enumeration ist also kein effizientes Verfahren. Ein solches gewinnen wir aber sofort aus Algorithmus 4.3.2.

Algorithmus 4.4.4 (Greedy-Algorithmus (Kruskal)) *Sortiere die Kanten so, dass*

$$w(e_1) \leq w(e_2) \leq \ldots \leq w(e_m)$$

und führe Algorithmus 4.3.2 aus.

Satz 4.4.5 *Der Greedy-Algorithmus berechnet einen minimalen aufspannenden Baum.*

Beweis. Als spezielle Implementierung von Algorithmus 4.3.2 berechnet der Greedy einen aufspannenden Baum T. Angenommen es gäbe einen aufspannenden Baum $\tilde{T}$ mit $w(\tilde{T}) < w(T)$. Sei dann ein solches $\tilde{T}$ so gewählt, dass $|T \cap \tilde{T}|$ maximal ist. Sei e die Kante mit kleinstem Index in $T \setminus \tilde{T}$. Dann schließt e in $\tilde{T}$ nach Satz 4.1.2 einen Kreis C_1. Nach Wahl von $\tilde{T}$ muss nun für alle $f \in C_1 \setminus T : w(f) < w(e)$ sein, denn sonst könnte man durch Austausch von e und einem solchen f mit $w(f) \geq w(e)$ einen Baum $\hat{T}$ konstruieren mit $w(\hat{T}) \leq w(\tilde{T}) < w(T)$ und $|T \cap \hat{T}| > |T \cap \tilde{T}|$. Sei nun $f \in C_1 \setminus T$. Da f vom Greedy-Algorithmus verworfen wurde, schließt es mit T einen Kreis C_2 mit für alle $g \in C_2 : w(g) \leq w(f)$. Sei $g \in C_2 \setminus \tilde{T}$. Dann ist $g \in T \setminus \tilde{T}$ und $w(g) \leq w(f) < w(e)$. Also hat g einen kleineren Index als e im Widerspruch zur Wahl von e. □

Bemerkung 4.4.6 *Für das Sortieren der Kanten benötigt man bekanntlich* $O(|E| \log(|E|))$*, also erhalten wir wegen*

$$O(\log(|E|)) = O(\log(|V|^2)) = O(\log(|V|))$$

mit unserer Implementierung von UNION-FIND ein Verfahren der Komplexität $O((|E| + |V|) \log(|V|))$.

Kapitel 5

Graphen in der Ebene

Einige Graphen, die aus Anwendungen motiviert sind, lassen sich kreuzungsfrei in die Ebene zeichnen. Die gilt z.B. für Straßennetze ohne Brücken oder Unterführungen.

5.1 Planare Graphen

Folgendes Problem ist ähnlich populär wie das Königsberger Brückenproblem.

Problem 5.1.1 *Drei Männer Y_1, Y_2, Y_3 besuchen regelmäßig drei Frauen X_1, X_2, X_3. Die Termine sind so koordiniert, dass sich nie zwei Männer bei der gleichen Frau zur gleichen Zeit treffen. Dennoch kommt es unterwegs oft zu hässlichen Szenen.*

Gibt es eine Möglichkeit die Wege P_{ij} von Y_i zu X_j so festzulegen, dass sie intern kreuzungsfrei sind, also die Gefahr einer ungeplanten Begegnung vermieden wird?

Definition 5.1.2 *Ein Multigraph $G = (V, E)$ heißt* planar*, wenn er eine planare Einbettung hat, d.i. eine Abbildung $p : V \to \mathbb{R}^2$ sowie eine Abbildung $l : E \to \mathcal{J}$, wobei $\mathcal{J}$ die Menge aller offenen Jordanbögen (das sind doppelpunktfreie, einfache Kurven in der Ebene ohne ihre Randpunkte) ist, so dass für jede Kante $e = (u, v)$ der topologische Rand von $l(e)$ gerade $p(u)$ und $p(v)$ sind und für je zwei Kanten e, f $l(e) \cap l(f) = \emptyset$, m.a.W. man kann den Graphen kreuzungsfrei in die Ebene zeichnen.*

Problem 5.1.1 ist also gleichbedeutend mit der Fragestellung, ob der $K_{3,3}$ planar ist.

Bei zusammenhängenden, ebenen Multigraphen besteht folgende Beziehung zwischen der Anzahl der Knoten V, Kanten E und Gebiete F. Dabei sind die Gebiete die zusammenhängenden Komponenten von $\mathbb{R}^2 \setminus (L(E) \cup p(V))$.

Lemma 5.1.3 (Eulersche Polyederformel)

$$|V| - |E| + |F| = 2. \tag{5.1}$$

Beweis. Wir führen Induktion über $|V|$. Ist $|V| = 1$, so sind alle Kanten Schleifen. Mit dem Außengebiet zählen wir $|E| + 1$ Gebiete. Also ist $|V| - |E| + |F| = 1 - |E| + |E| + 1 = 2$.

Sei also nun $G(V, E)$ ein zusammenhängender, planarer Multigraph mit $|V| > 1$ Knoten und $e \in E$ eine Kante, die keine Schleife ist. Kontrahieren wir e, so erhalten wir einen zusammenhängenden, planaren Multigraphen, der ebensoviele Gebiete, aber genau einen Knoten und genau eine Kante weniger hat als G. Für diesen gilt nach Induktionsvoraussetzung

$$|V| - 1 - (|E| - 1) + |F| = 2,$$

woraus die Behauptung folgt. □

Bemerkung 5.1.4 *Wenn wir die Formel lesen als*

$$-1 + |V| - |E| + |F| - 1 = 0,$$

wobei die erste -1 die leere Menge und die letzte den ganzen Graphen zählt, können wir sie auf beliebig dimensionale Polyeder (eckige, konvexe Körper) verallgemeinern zu

> *Zählt man die Seitenflächen eines Polyeders nach Dimension geordnet mit alternierendem Vorzeichen, so erhält man stets Null.*

5.2 In planaren Graphen ist $|E| = O(|V|)$

Planare Graphen müssen stets „sparse“ sein, sie dürfen nur relativ wenige Kanten haben.

Lemma 5.2.1 *Sei G ein planarer Graph, in dem alle Gebiete Dreiecke sind, also von genau drei Kanten berandet werden. Dann ist*

$$|E| = 3|V| - 6. \tag{5.2}$$

Beweis. Da alle Gebiete Dreiecke sind, gilt

$$2|E| = \sum_{f \in F} 3 = 3|F|$$

und somit $|F| = \frac{2}{3}|E|$. Setzen wir dies in (5.1) ein, so erhalten wir.

$$-\frac{1}{3}|E| + |V| \quad = \quad 2,$$

woraus die Behauptung folgt. □

Korollar 5.2.2 *Sei $G = (V, E)$ ein (einfacher) planarer Graph. Dann ist*

$$|E| \leq 3|V| - 6.$$

Beweis. Wir betrachten eine Einbettung von G. Durch Hinzufügen von Kanten können wir einen planaren Graphen $\tilde{G} = (V, \tilde{E})$ mit $E \subseteq \tilde{E}$ konstruieren, dessen Gebiete alle Dreiecke sind. Dann ist $|E| \leq |\tilde{E}| = 3|V| - 6$. □

In bipartiten planaren Graphen sind die Gebiete sogar alle mindestens Vierecke. Also erhalten wir analog.

Lemma 5.2.3 *Sei G ein, bipartiter, planarer Graph, in dem alle Gebiete Vierecke sind. Dann ist*

$$|E| = 2|V| - 4. \tag{5.3}$$

Beweis. Wir haben $2|E| = \sum_{f \in F} 4 = 4|F|$ und somit $|F| = \frac{1}{2}|E|$. Setzen wir dies in (5.1) ein, folgt die Behauptung. □

Korollar 5.2.4 *Sei $G = (V, E)$ ein (einfacher) planarer, bipartiter Graph. Dann ist*

$$|E| \leq 2|V| - 4.$$

□

5.3 Der Satz von Kuratowski

Nun können wir zeigen, dass es für Problem 5.1.1 keine Lösung geben kann.

Satz 5.3.1 *Der K_5 und der $K_{3,3}$ sind nicht planar.*

Beweis. Für den K_5 gilt:

$$10 = |E| > 3|V| - 6 = 9$$

und für den $K_{3,3}$:

$$9 = |E| > 2|V| - 4 = 8.$$

□

Diese beiden Graphen sind tatsächlich die „einzigen Obstruktionen“ für Planarität. Genauer gilt der Satz von Kuratowski, den wir hier ohne Beweis angeben.

Satz 5.3.2 (Kuratowski) *Ein Graph ist genau dann planar, wenn er keine Unterteilung des $K_{3,3}$ oder des K_5 als Teilgraphen hat.*

Kapitel 6

Die Methode des doppelten Abzählens

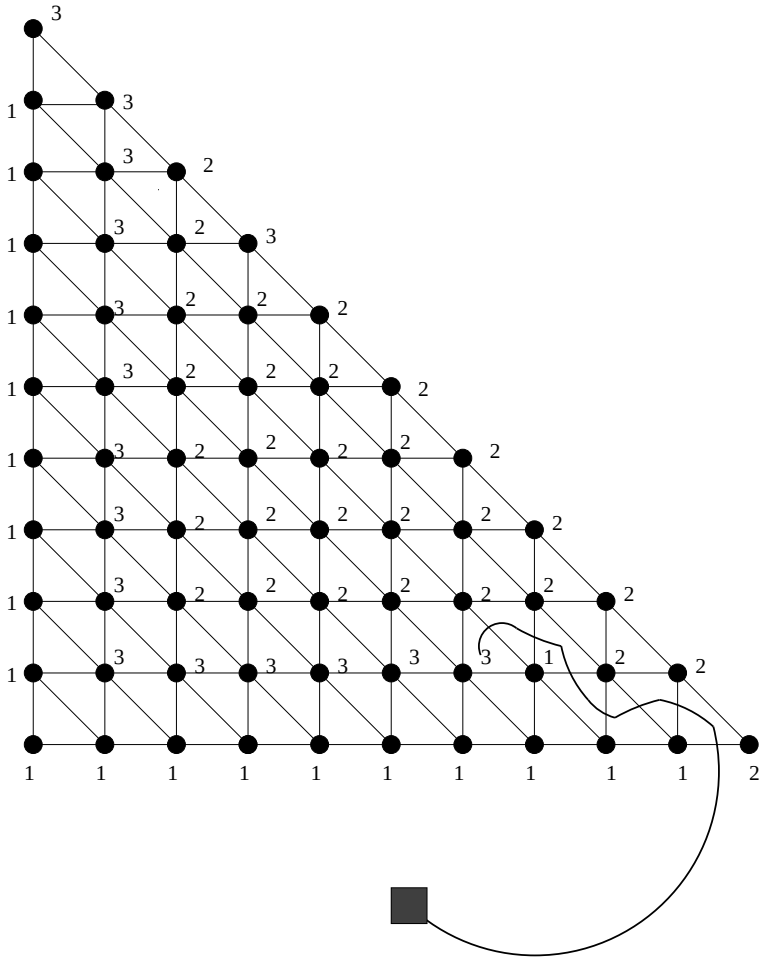
6.1 Paritätsargumente

Wir betrachten folgende Situation. Ein ebenes Dreieck mit den Ecken A_1, A_2, A_3 ist in kleinere Dreiecke unterteilt, wobei keine Ecke eines Dreiecks auf einer Seite (Kante) eines anderen Dreiecks liegen soll. Wir sagen, das Dreieck ist *trianguliert.* Ferner seien alle Ecken eines kleinen Dreiecks mit einer der Zahlen 1,2 oder 3 versehen. Dabei sei A_i mit dem Label i versehen für $i = 1, 2, 3$ und auf der Dreiecksseite, die die Ecken A_i und A_j verbindet, kommen nur die Label i und j vor.

Lemma 6.1.1 (Sperner-Lemma, ebene Version) *In der geschilderten Situation gibt es ein Dreieck, das drei verschiedene Label hat.*

Beweis. Wir definieren folgenden Graphen. Die Knotenmenge sei $V = D \cup v_0$, wobei D die Menge der kleinen Dreiecke sei und v_0 für das Gebiet außerhalb des Dreiecks $A_1A_2A_3$ stehe. Zwei Knoten seien adjazent, wenn die zugehörigen Dreiecke eine gemeinsame Kante haben und diese die Labels 1 und 2 trägt. Wir untersuchen diesen Graphen. Wenn ein kleines Dreieck die Labels 1, 2, 3 hat, dann ist es zu genau einem weiteren Dreieck adjazent, es ist ein Blatt. Anderfalls hat ein Dreieck entweder keinen Nachbarn oder die Labels 1,1,2 oder 1,2,2. In beiden Fällen hat es genau zwei Nachbarn also insbesondere geraden Knotengrad. Nun betrachten wir v_0. Die Adjazenz zu diesem Knoten kann nur durch Kanten vermittelt werden, die auf der Seite A_1A_2 liegen. Laufen wir diese Strecke von A_1 nach A_2, so haben wir jeweils

einen Nachbarn wenn das Label wechselt, beim ersten Nachbarn von 1 auf 2, dann von 2 auf 1, wieder von 1 auf 2 und so weiter. Da aber A_2 das Label 2 hat, muss ein solcher Wechsel ungerade oft stattfinden, der Knoten v hat also ungeraden Knotengrad. Nach dem Handshake Lemma muss es mindestens einen weiteren Knoten mit ungeradem Knotengrad geben. Nach unserer Analyse muss dieser Knoten zu einem vollständig gelabelten Dreieck gehören. □



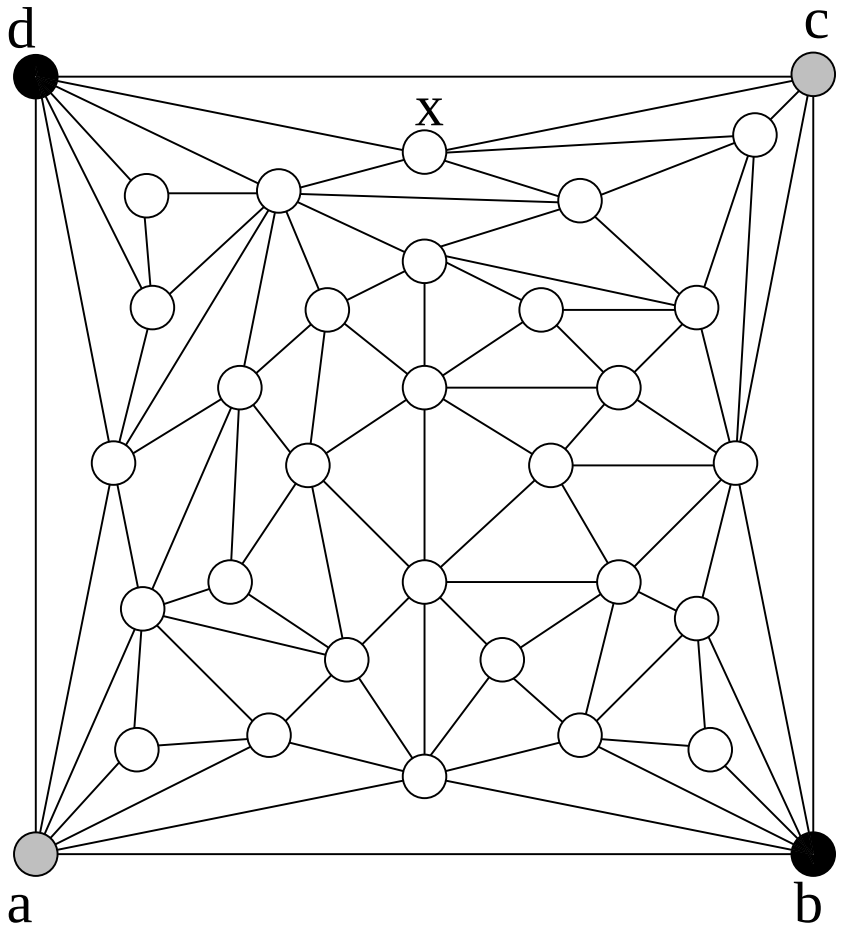
Genauer haben wir sogar gezeigt.

Satz 6.1.2 *In der geschilderten Situation gibt es eine ungerade Anzahl Dreiecke, die drei verschiedene Label haben.*

□

Bemerkung 6.1.3 *Den Beweis kann man als Induktionsschritt betrachten. Nach Induktionsvoraussetzung gibt es zwischen A_1 und A_2 ungerade viele Kanten mit beiden Labels, dies ist auch die Induktionsverankerung. Also muss es auch ungerade viele Dreiecke mit allen Labels geben. Dieser Beweis lässt sich dann auf* Simplizes *beliebiger Dimension verallgemeinern.*

Als nächstes betrachten wir eine Triangulierung eines Quadrates, bei der im Innern der Randkanten keine weiteren Triangulierungsknoten liegen.



Auf diesem wird folgendes Spiel gespielt. Es wird je abwechselnd ein Knoten gefärbt. Alice färbt schwarz und Bob färbt grau. Zu Anfang sind je zwei diagonal gegenüberliegende Knoten mit den jeweiligen Farben gefärbt, a und c in grau und b und d in schwarz. Ein Spieler hat gewonnen, wenn er seine Eckknoten durch einen Weg über Knoten in seiner Farbe gefärbt hat.

Satz 6.1.4 *In dem beschriebenen Spiel endet keine Partie unentschieden.*

Beweis. Wir können davon ausgehen, dass alle Knoten gefärbt sind, denn wenn ein Spieler gewonnen hat, färben wir alle übrigen Knoten in seiner Farbe und ansonsten könnten wir weiter spielen. Wir markieren nun alle grauen Knoten, die auf einem grauen Weg von a aus erreichbar sind mit 1, alle schwarzen Knoten, die auf einem schwarzen Weg von b erreichbar sind mit 2 und alle anderen mit 3. Angenommen nun d und c wären mit 3 gelabelt. Die Knoten d und c haben einen gemeinsamen Nachbarn, mit dem sie ein Dreieck bilden, da im Innern der Strecke cd kein Knoten liegt. Dieser Knoten x muss dann auch mit mit 3 gelabelt sein. Wir entfernen nun die Kante cd und setzen $A_3 = x$. Dadurch erhalten wir eine gelabelte Triangulierung eines Dreiecks, das die Bedingungen des Sperner Lemmas erfüllt. Es gibt also ein Dreieck mit einem schwarzen Knoten, der auf einem schwarzen Weg von b aus erreicht werden kann, mit einem grauen Knoten, der auf einem grauen Weg von a aus erreicht werden kann und einem schwarz oder grau gefärbten Knoten, der weder auf einem schwarzen noch einem grauen Weg erreicht wird. Dies ist ein Widerspruch.

An statt die Strecke cd zu entfernen hätten wir ebenso gut einen Hut aufsetzen können, also ein Dreieck mit einer neuen Ecke y, die mit 3 gelabelt ist, hinzufügen können. □

6.2 Der Satz von Sperner

Der Satz von Sperner hat mit dem Sperner Lemma nur den Namengeber gemeinsam. Sei X eine endliche Menge und $\mathcal{M}$ ein System von Teilmengen von X. Dann nennen wir $\mathcal{M}$ eine *Antikette*, wenn es keine Mengen $A \neq B \in \mathcal{M}$ gibt mit $A \subset B$.

Satz 6.2.1 (Satz von Sperner) *Ist ein System von Teilmengen $\mathcal{M}$ einer endlichen Menge X mit $|X| = n$ eine Antikette, so ist*

$$|\mathcal{M}| \leq \binom{n}{\lfloor \frac{n}{2} \rfloor}. \tag{6.1}$$

Beweis. Außer der Antikette betrachten wir Ketten, das sind Mengen von Teilmengen $\{A_1, A_2, \ldots, A_k\}$ mit $i < j \Rightarrow A_i \subset A_j$. Genauer betrachten wir maximale Ketten $\mathcal{R}$, das sind Ketten, in die man keine weitere Teilmenge aufnehmen kann, ohne die Ketteneigenschaft zu verletzen. Diese haben offensichtlich die Struktur

$$\emptyset \subset \{x_1\} \subset \{x_1, x_2\} \subset \{x_1, x_2, x_3\} \subset \ldots \subset \{x_1, x_2, \ldots, x_n\}. \tag{6.2}$$

Ketten entsprechen also den linearen Anordnungen oder Permutationen, es gibt davon $n!$ Stück. Eine Kette und eine Antikette können nun offensichtlich höchstens ein Element gemeinsam haben.

Sei nun $\mathcal{M}$ eine Antikette. Wir betrachten die Paare aus maximalen Ketten und Mengen in der Antikette $\mathcal{M}$, die in der Kette enthalten sind, also $\{(\mathcal{R}, M) \mid M \in \mathcal{R} \cap \mathcal{M}\}$. Da stets höchstens ein $M \in \mathcal{M}$ in einer Kette liegt, ist $|\{(\mathcal{R}, M) \mid M \in \mathcal{R} \cap \mathcal{M}\}| \leq n!$. Umkehrt betrachten wir ein festes $M = \{x_1, \ldots, x_k\} \in \mathcal{M}$. Dann liegt M in der maximalen Kette $\mathcal{R}$ genau dann, wenn $\mathcal{R}$ mit einer Permutation von $\{x_1, \ldots, x_k\}$ beginnt und danach die Elemente $\{x_{k+1}, \ldots, x_n\}$ permutiert. Also liegt M in $|M|!(n-|M|)!$ Ketten. Also erhalten wir

$$n! \geq \sum_{\{(\mathcal{R},M)|M\in\mathcal{R}\cap\mathcal{M}\}} 1 = \sum_{M\in\mathcal{M}} |M|!(n-|M|)!$$

und somit

$$\sum_{M\in\mathcal{M}} \frac{1}{\binom{n}{|M|}} \leq 1.$$

Nach (2.21) ist $\binom{n}{|M|} \leq \binom{n}{\lfloor \frac{n}{2} \rfloor}$ also

$$\sum_{M\in\mathcal{M}} \frac{1}{\binom{n}{\lfloor \frac{n}{2} \rfloor}} \leq \sum_{M\in\mathcal{M}} \frac{1}{\binom{n}{|M|}} \leq 1$$

und somit $|\mathcal{M}| \leq \binom{n}{\lfloor \frac{n}{2} \rfloor}$. □

6.3 Ein Resultat aus der extremalen Graphentheorie

Die extremale Graphentheorie beschäftigt sich mit Fragestellungen wie

> Wieviel Kanten muss ein Graph haben, damit das Auftreten einer gewissen Struktur garantiert ist.

Z.B. wissen wir nach Satz 4.1.2, dass ein Graph mit mindestens $|V|$ Kanten stets einen Kreis enthalten muss. Wir zeigen folgenden Satz, den Paul Erdös 1938 gefunden hat.

Satz 6.3.1 *Ein Graph mit n Knoten, der kein Viereck enthält, also keinen Teilgraphen isomorph zum $K_{2,2} = C_4$ hat, hat höchstens $\frac{1}{2}(n\sqrt{n}+n)$ Kanten.*

Beweis. Wir zählen die möglichen Teilgraphen $K_{1,2}$, indem wir die Menge M aller Paare $(v, \{u, u'\})$ zählen, bei denen v mit u und u' adjazent ist. Für jedes Paar $\{u, u'\} \in \binom{V}{2}$ kann es höchstens ein solches $v \in V$ geben, da bei einem weiteren solchen, etwa $v' \in V$ die Knoten v, v', u, u' alle Kanten eines $K_{2,2}$ hätten. Also ist $|M| \leq \binom{n}{2}$. Der Knoten v gehört nun zu $\binom{\deg(v)}{2}$ solcher Paare, also haben wir

$$\binom{n}{2} \geq |M| = \sum_{i=1}^{n} \binom{\deg(v_i)}{2}. \tag{6.3}$$

Wir können davon ausgehen, dass unser Graph keine isolierten Knoten hat, also ist $\deg(v_i) \geq 1$ für alle v_i. Damit können wir abschätzen $\binom{\deg(v_i)}{2} \geq \frac{1}{2}(\deg(v_i) - 1)^2$ und somit erhalten wir aus (6.3)

$$\sum_{i=1}^{n} (\deg(v_i) - 1)^2 \leq n^2 - n \leq n^2. \tag{6.4}$$

Für den Rest des Beweises benötigen wir eine bekannte Ungleichung aus der Linearen Algebra, die wir im nächsten Semester beweisen werden.

Satz 6.3.2 (Cauchy-Schwarzsche Ungleichung) *Seien $x_1, \ldots, x_n \in \mathbb{R}$ und $y_1, \ldots, y_n \in \mathbb{R}$. Dann ist*

$$\sum_{i=1}^{n} x_i y_i \leq \sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2} \tag{6.5}$$

□

Wir wenden nun die Cauchy-Schwarzsche Ungleichung auf die Zahlen $x_1 = \deg(v_1) - 1, \ldots, x_n = \deg(v_n) - 1)$ und $y_1 = 1, \ldots, y_n = 1$ an und erhalten

$$\sum_{i=1}^{n} (\deg(v_i) - 1) \leq \sqrt{\sum_{i=1}^{n} (\deg(v_i) - 1)^2} \sqrt{n}.$$

Setzen wir hierein nun (6.4) ein, so folgt

$$2|E(G)| - n = \sum_{i=1}^{n} (\deg(v_i) - 1) \leq n\sqrt{n}. \tag{6.6}$$

Lösen wir dies nach $|E|$ auf, so folgt die Behauptung. □

Kapitel 7

Die Anzahl aufspannender Bäume und vier Beweise

7.1 Die Cayley-Formel

In diesem Kapitel wollen wir die aufspannenden Bäume von (gelabelten) vollständigen Graphen bestimmen. Für einen Graphen G bezeichne $T(G)$ die Anzahl aufspannender Bäume, wobei wir isomorphe, aber verschiedene Bäume als unterschiedlich zählen, z.B. ist $T(K_3) = 3$, obwohl alle drei aufspannenden Bäume isomorph sind. Deswegen sprechen wir von gelabelten Bäumen.

Satz 7.1.1 (Cayley-Formel) *Sei $n \geq 2$. Dann gilt*

$$T(K_n) = n^{n-2}.$$

Einen Beweis, der ähnlich einfach aussieht, lernen wir als letzten kennen. Wir stellen vorher in diesem Kapitel verschiedene Beweise für die Formel vor, da diese unterschiedliche Ideen und Techniken benutzen. Damit wollen wir, zum Abschluss unserer Ausflüge in die Graphentheorie, auch die Reichhaltigkeit der Ideen und Methoden in diesem Gebiet dokumentieren.

7.2 Ein Beweis mit Valenzsequenzen

Zunächst einmal zählen wir Bäume zu gegebenen Valenzsequenzen.

Lemma 7.2.1 *Seien $d_1, \ldots, d_n \in \mathbb{N}$ und $d_i \geq 1$ für alle i, und gelte die Gleichung $\sum_{i=1}^{n} d_i = 2n-2$. Die Anzahl der aufspannenden Bäume von K_n, in denen der Grad des Knoten $v_i \in V$ genau d_i ist, beträgt*

$$\frac{(n-2)!}{(d_1-1)!(d_2-1)!\ldots(d_n-1)!}. \tag{7.1}$$

Beweis. Wir führen Induktion über $n = |V|$. Für $n = 1$ gibt es keine solchen Zahlen d_i, die Aussage ist leer, also wahr. Für $n = 2$ ergibt die Formel 1. Sei also $n > 2$. Wegen $\sum_{i=1}^{n} d_i < 2n$, gibt es ein i mit $d_i = 1$, wir können annehmen $i = n$. Sonst vertauschen wir i und n. Der Knoten v_n ist also in jedem Baum mit dieser Gradsequenz ein Blatt. Der Vater von v_n ist ein Knoten v_j mit $d_j > 1$, also ist $T \setminus v_n$ ein Baum mit Valenzsequenz $(d_1, \ldots, d_{j-1}, d_j - 1, d_{j+1}, \ldots, d_{n-1})$. Umgekehrt erhalten wir aus jedem solchen Baum durch Anhängen des Blattes v_n an v_j einen Baum mit den gewünschten Eigenschaften.

Die Anzahl der aufspannenden Bäume von K_{n-1} mit Valenzsequenz $(d_1, \ldots, d_{j-1}, d_j - 1, d_{j+1}, \ldots, d_{n-1})$ ist aber nach Induktionsvoraussetzung

$$\begin{aligned}&\frac{(n-3)!}{(d_1-1)!\ldots(d_{j-1}-1)!(d_j-2)!(d_{j+1}-1)!\ldots(d_{n-1}-1)!}\\ =\ &\frac{(n-3)!(d_j-1)}{(d_1-1)!\ldots(d_{j-1}-1)!(d_j-1)!(d_{j+1}-1)!\ldots(d_{n-1}-1)!}.\end{aligned}$$

In der Gestalt auf der rechten Seite gilt die Formel auch, wenn $d_j = 1$, also v_j ein Blatt ist, denn in diesem Fall gibt es keinen solchen Baum und die Formel ergibt 0. Also ist die gesuchte Zahl

$$\begin{aligned}&\sum_{j=1}^{n} \frac{(n-3)!(d_j-1)}{(d_1-1)!\ldots(d_{j-1}-1)!(d_j-1)!(d_{j+1}-1)!\ldots(d_{n-1}-1)!}\\ =\ &\left(\sum_{j=1}^{n}(d_j-1)\right)\frac{(n-3)!}{(d_1-1)!\ldots(d_{j-1}-1)!(d_j-1)!(d_{j+1}-1)!\ldots(d_{n-1}-1)!}\\ =\ &(2n-2-n)\frac{(n-3)!}{(d_1-1)!\ldots(d_{j-1}-1)!(d_j-1)!(d_{j+1}-1)!\ldots(d_{n-1}-1)!}\\ =\ &\frac{(n-2)!}{(d_1-1)!\ldots(d_{j-1}-1)!(d_j-1)!(d_{j+1}-1)!\ldots(d_{n-1}-1)!(d_n-1)!}.\end{aligned}$$

Die letzte Gleichung benutzte auch $(d_n - 1)! = 0! = 1$. □

Nutzen wir nun den Multinomialsatz (2.10)aus, so erhalten wir

$$\begin{aligned} n^{n-2} &= \underbrace{(1+1+\ldots+1)}_{n-mal}{}^{n-2} \\ &= \sum_{\substack{k_1+k_2+\ldots+k_n=n-2 \\ k_1,\ldots,k_n\geq 0}} \frac{(n-2)!}{k_1!k_2!\ldots k_n!} \\ &= \sum_{\substack{d_1+d_2+\ldots+d_n=2n-2 \\ d_1,\ldots,d_n\geq 1}} \frac{(n-2)!}{(d_1-1)!(d_2-1)!\ldots(d_n-1)!}. \end{aligned}$$

Das war der erste Beweis der Cayley Formel. □

7.3 Ein Beweis mit Wirbeltieren

Definition 7.3.1 *Ein Wirbeltier ist ein Tripel (T, t, h), wobei T ein Baum ist und t, h zwei ausgezeichnete (eventuell gleiche) Knoten in V sind, der Kopf und der Schwanz.*

Die Cayleysche Formel ist dann offensichtlich äquivalent zu der Aussage, dass es genau n^n Wirbeltiere auf n Knoten gibt.

Um dies zu zeigen konstruieren wir eine Bijektion zwischen den Wirbeltieren auf den Knoten $V = \{1, \ldots, n\}$ und den Abbildungen $\tau : V \to V$ einer n-elementigen Menge in sich selbst. Davon gibt es nach Proposition 2.1.2 n^n Stück.

Lemma 7.3.2 *Es gibt eine Bijektion zwischen der Menge aller Wirbeltiere auf V und der Menge aller Abbildungen von V in sich selbst.*

Beweis. Sei dazu zunächst τ eine solche Abbildung. Wir betrachten den Digraphen $G = (V, A)$, wobei $(i, j) \in A \Leftrightarrow j = f(i)$. Also hat in diesem Digraphen jeder Knoten genau eine ausgehende Kante. Folglich gibt es in jeder Komponente des zu Grunde liegenden Graphen mit $n_1 \leq n$ Knoten genau n Kanten, ein solcher Graph besteht also aus einem Baum plus einer Kante. Diese Kante schließt einen eindeutigen Kreis mit dem Baum. Jeder Kreis einer solchen Komponente definiert einen Zyklus. Alle Zyklen von Kreisen solcher Komponenten definieren somit eine Permutation und diese eine lineare Ordnung auf den Elementen in den Zyklen. Wir ersetzen nun alle gerichteten Kreise durch einen gerichteten Pfad, der die Knoten in dieser

Reihenfolge durchläuft, nennen den Anfang dieses Weges t und das Ende h, vergessen die Orientierung und haben ein Wirbeltier konstruiert.

Durchlaufen wir diese Konstruktion rückwärts, können wir ausgehend von einem beliebigen Wirbeltier eine Abbildung von V in sich selber definieren, wobei wir τ aus seinem Wirbeltier rekonstruieren. Da wir ebenso ausgehend von einem Wirbeltier seine Abbildung konstruieren und daraus das Wirbeltier rekonstruieren können, ist die Zuordnung bijektiv. □

Beispiel 7.3.3 *Wir betrachten die Abbildung*

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
17	13	9	19	3	10	7	14	8	5	3	2	12	13	7	17	9	13	6	14

Zunächst erhalten wir die Zyklen $(2, 13, 12)(7)$, diese ergeben die lineare Ordnung $(13, 7, 2, 12)$ und hieraus das Wirbeltier in Abbildung 7.1.

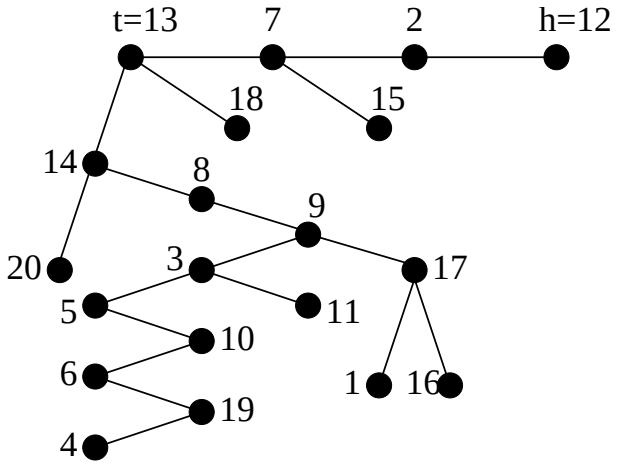


Abbildung 7.1: Ein Wirbeltier

7.4 Der Prüfer-Code

Dies ist ein Klassiker. Die Originalarbeit von H. Prüfer stammt aus dem Jahre 1918.

Wir kodieren die gelabelten aufspannenden Bäume T des K_n, dessen Knoten mit $\{1,\ldots,n\}$ gelabelt sind, eineindeutig durch Folgen von $n-2$ Zahlen in $\{1,..,n\}$, so dass jede solche Zahlenfolge auch wiederum genau einen aufspannenden Baum repräsentiert. Wir nennen diese Folgen auch Wörter $C(T)$ und können diese Wörter auch miteinander verketten. Ist z.B. $C(T) = (c_1,\ldots,c_n)$, so bezeichne $kC(T)$ die Folge $kC(T) = (k, c_1,\ldots,c_n)$. Tatsächlich ist der Prüfer-Code definiert für beliebige Labels, die streng linear geordnet sind.

Wir definieren diese Folgen rekursiv. Die Rekursion verankern wir im K_2, dem einzigen aufspannenden Baum des K_2. Diesem ordnen wir die leere Folge zu $C(K_2) = ()$. Sei nun T_k ein gelabelter Baum mit $k \geq 3$ Knoten. Sei i das Blatt mit dem kleinsten Index. Sei j der Index des eindeutigen zu i adjazenten Baumknoten und $T_{k-1} = T \setminus i$. Dann ist $C(T_k) = jC(T_{k-1})$.

Beispiel 7.4.1 *Wir betrachten den Baum aus Abbildung 7.1. Das Blatt mit kleinstem Index ist die 1, also beginnt unser Wort mit 17. Als nächstes erhalten wir den Vater von 4, also 19. Insgesamt erhalten wir die Folge*

$$(17, 19, 3, 2, 7, 7, 13, 17, 9, 13, 14, 6, 10, 5, 3, 9, 8, 14).$$

Offensichtlich liefert diese Rekursion einen String der Länge $n-2$, der die n Zeichen benutzt, mit denen der Baum gelabelt ist.

Umgekehrt müssen wir zu einer solchen Zeichenfolge einen gelabelten Baum konstruieren. Die erste Zahl einer solchen Folge ist der eindeutige Nachbar des Blattes mit dem kleinsten Label. Welches Label b_1 hatte dieses Blatt? Sicherlich keine Zahl, die in der Folge noch vorkommt, denn darin kommen keine Blätter vor. Umgekehrt treten aber alle Nichtblätter auf, da man um zu einer Kante zu gelangen, von jedem Nichtblatt mindestens einen Nachbarn entfernt haben muss, der also zu diesem Zeitpunkt ein Blatt war. Also ist $b_1 = \min\{1,\ldots,n\} \setminus C(T)$ Dies liefert uns nun wiederum eine Vorschrift, um den Baum rekursiv zu konstruieren. Am Schluss bleiben zwei Knoten übrig, die die Labels der $(n-1)$-te Kante ergeben.

Beispiel 7.4.2 *Wir rekonstruieren aus dem Prüfer-Code den Baum aus Abbildung 7.1. Wir setzen* $N = \{1,\ldots,20\}$ *und*

$$P = \{17, 19, 3, 2, 7, 13, 9, 14, 6, 10, 5, 8\}.$$

Die kleinste Zahl in $N \setminus P$ *ist* 1*, also ist die erste Kante* $(1, 17)$*. Wir streichen 17 aus der Folge, setzen* $N = \{N \setminus 1\}$ *und* $P = P$*, da die 17 noch in der Folge vorkommt.*

Die zweite Kante ist $(4, 19)$. $N = N \setminus \{4\}, P = P \setminus \{19\}$.

Die nächsten Kanten sind $(11, 3), (12, 2), (2, 7)$. *Den bisher konstruierten Teilwald haben wir links in Abbildung 7.2 skizziert, aus* N *haben wir 1,2,3,4,und 11 entfernt und aus* P *19 und 2.*

Die nächsten Kanten sind $(15, 7), (7, 13), (16, 17), (17, 9), (18, 13), (13, 14)$ *und wir haben den Teilwald in der Mitte.*

Schließlich fügen wir noch die Kanten $(19, 6), (6, 10), (10, 5), (5, 3), (3, 9), (9, 8)$ *und* $(8, 14)$ *hinzu. Unsere Menge* P *ist nun leer und* $N = \{14, 20\}$ *und wir erhalten die Darstellung auf der rechten Seite, die, wie man sich überzeugt, isomorph zu dem gelabelten Baum aus Abbildung 7.1 ist.*

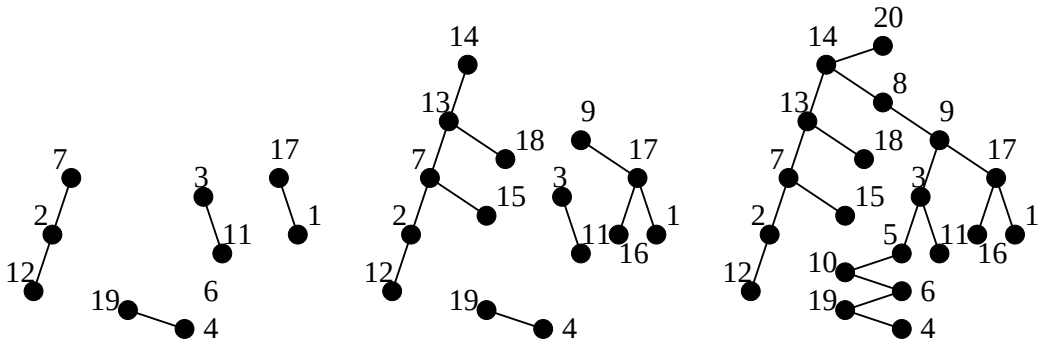


Abbildung 7.2: Dekodierung des Prüfer-Codes

Wir haben somit eine Vorschrift angegeben, die aus einer beliebigen Sequenz in $\{1, \ldots, n\}^{n-2}$ einen Graphen mit $n - 1$ Kanten konstruiert und aus dem Prüfer-Code eines Baumes den Baum rekonstruiert. Wir müssen nun noch zeigen, dass der so enstehende Graph stets ein Baum ist, der diesen Code hat. Wir zeigen sogar:

Proposition 7.4.3 *Die angegebene Konstruktion generiert aus einer beliebigen Sequenz in* $\{1, \ldots, n\}^{n-2}$ *einen Wurzelbaum, dessen zugrundeliegender Baum die vorgegebene Sequenz als Prüfer-Code hat.*

Beweis. Dies zeigen wir durch Induktion über $n \geq 2$. Für den leeren Code ist die Behauptung richtig. Sei also $n \geq 3$, $b_1, \ldots, b_{n-2}$ ein Code und $j = \min \{\{1, \ldots, n\} \setminus \{b_1, \ldots, b_{n-2}\}\}$. Nach Induktionsvoraussetzung ist $b_2, \ldots, b_{n-2}$ der Prüfer-Code des Baumes T', der der Arboreszenz $\vec{T}'$ mit Labels $\{1, \ldots, n\} \setminus \{j\}$ zu Grunde liegt, die durch die Konstruktionvorschrift entsteht. Dann entsteht T aus T' durch Anhängen des Blattes j mit dem Bogen (b_1, j), dies ist

ein Baum nach Lemma 4.1.4 und offensichtlich wieder gewurzelt. Der Index j ist offensichtlich daran das Blatt mit dem kleinsten Index, da alle Knoten mit kleinerem Index noch einen Nachfahren haben. □

7.5 Kantengelabelte Wurzelbäume

Der folgende Beweis ist relativ neu von Jim Pitman (1998) und überraschend einfach. Wir zählen kanten- und knotengelabelte Wurzelbäume, d.h. wir haben knotengelabelte Wurzelbäume, bei denen die Kanten noch mit Labeln in $\{1, 2, \ldots, n-1\}$ versehen sind. Die Cayleyformel ist dann äquivalent dazu, dass wir mit den n Wahlmöglichkeiten für die Wurzel und den $(n-1)!$ möglichen Kantenlabelungen insgesamt $n^{n-1}(n-1)!$ solcher Objekte haben. An und für sich haben wir in dieser Äquivalenz, $T(G)$ durch doppeltes Abzählen bestimmt.

Wir hatten Wurzelbäume als von der Wurzel weg orientierte Graphen betrachtet. Jeder Knoten, abgesehen von der Wurzel ist Spitze genau eines Bogens. Wir interpretieren nun die Labelings der Kanten als Reihenfolge, in der die Kanten zu einem Wurzelbaum zusammengefügt werden. Zu jedem Zeitpunkt bildet dann die Menge der gerichteten Kanten einen Wald, in dem jeder Baum gewurzelt ist. Sind k Kanten eingefügt, so haben wir $n-k$ Komponenten, also ebensoviele Wurzeln. Die neue Kante kann nun aus einem beliebigen Knoten hinausführen, davon gibt es n Stück, aber enden darf sie nur in der Wurzel einer anderen Komponente, dafür gibt es $n-k-1$ Möglichkeiten. Jeder kanten- und knotengelabelte Wurzelbaum wird auf diese Weise konstruiert und jede solche Konstruktion führt zu einem anderen solchen Baum.

Also zählen wir insgesamt

$$\prod_{k'=0}^{n-2} n(n-k'-1) = \prod_{k=1}^{n-1} n(n-k) = (n-1)!n^{n-1}$$

solcher Bäume.

□

Kapitel 8

Einführung in die Logik

8.1 Allgemeine Fragestellungen

Was ist Logik? Zunächst einmal versteht man unter Logik so etwas wie das „folgerichtige Denken“ oder Argumentieren. Wir wollen uns hier dem Begriff der Logik von einer streng formalen Seite her nähern und das kennenlernen, was in der Mathematik unter Logik verstanden wird.

Die Ursprünge dieses Logikbegriffs werden üblicherweise auf Aristoteles (382 –344 v. Chr.) und Leibniz (1646–1716) zurückgeführt. Betrand Russel (1872–1970) zitiert einen (lateinischen) Text von Leibniz, der das Leibniz'sche Ideal der Logik beschreibt, wie folgt: “If controversies were to arise, there would be no more need of disputation between two philosophers than between two accountants. For it would suffice to take their pencils in their hands, to sit down to their slates, and say to each other (with a friend as witness, if they liked): Let us calculate.”([12]). Um uns diesem Ideal zu nähern, müssen wir *logisches Schließen* auf ein *formales Kalkül* reduzieren. Dafür ist es nötig die zu behandelnden Aussagen zu *algebraisieren.*

Bei den formalen Sprachsysytemen, die wir dabei kennenlernen werden, werden wir uns auf die Prädikatenlogik erster Stufe beschränken und besonderes Augenmerk auf den Spezialfall der Aussagenlogik legen, die in vielen Bereichen der Informatik eine wichtige Rolle spielt.

Dass der Leibniz'sche Anspruch schon in der Mathematik prinzipiell unerfüllbar ist, hat Gödel mit seinem Unvollständigkeitssatz bewiesen. Wir empfehlen als Lektüre das Spektrum der Wissenschaft Biographie 1/2002.

8.2 Beispiele

Aussagen im *aristotelischen Sinne* sind sprachliche Gebilde, denen man eindeutig einen Wahrheitswert zuweisen kann (die Aussage ist wahr oder falsch, ein drittes gibt es nicht, *tertium non datur*).

Beispiele für solche Aussagen sind:

> Sylvester 2004 war ein Freitag.

und

> Sylvester 2004 war vorlesungsfrei.

> Sylvester 2004 war Ostern und Sylvester 2004 war vorlesungsfrei.

Die letzte Aussage ist zusammengesetzt und lässt sich „zerlegen“ in zwei aristotelische Aussagen, die miteinander durch „und“ verbunden sind.

Betrachten wir hingegen folgendes Konstrukt:

> Der Satz im unteren Kasten ist falsch.
> Der Satz im oberen Kasten ist wahr.

Umgangssprachlich betrachtet sehen die Gebilde wie Aussagen aus. Bei der Zuweisung von Wahrheitswerten stoßen wir allerdings auf Probleme. Wenn der Satz im oberen Kasten wahr ist, muss der im unteren Kasten falsch sein. Also muss der im oberen Kasten falsch sein. Wenn der Satz im oberen Kasten hingegen falsch ist, ist der im unteren wahr, also der im oberen wahr.

Man kann dieses Beispiel auch in den (selbstbezüglichen) Satz „Dieser Satz ist falsch“ verpacken.

Man kann also den einzelnen Aussagen keinen sinnvollen Wahrheitswert zuweisen. Man kann nun durch Vorgeben einer *Syntax*, d.i. eine Konstruktionsvorschrift für Aussagen, ausschließen, dass solche Gebilde formuliert werden.

Betrachten wir ein anderes Beispiel, das (solche Probleme umgeht und) sich leicht formalisieren lässt.

Beispiel 8.2.1 *Ein Student, der eine Klausur nicht bestanden hat, fragt einen erfolgreicheren Kommilitonen um Rat. Der sagt, es sei ganz einfach, er würde drei Regeln beachten.*

a) Wenn ich die Vorlesung blau mache, arbeite ich den Stoff nach und

b) wenn ich in die Vorlesung gehe und den Stoff nacharbeite, spare ich mir das Tutorium und

c) wenn ich ins Tutorium gehe oder die Vorlesung blau mache, arbeite ich den Stoff nicht nach.

Formalisieren wir dies nun, indem wir mit v symbolisieren, dass er in die Vorlesung geht, mit n Nacharbeiten und mit t den Besuch des Tutoriums.

a) $\neg v \rightarrow n$

b) $v \wedge n \rightarrow \neg t$

c) $t \vee \neg v \rightarrow \neg n$.

Untersuchen wir die möglichen Handlungsweisen an Hand einer Wahrheitstafel. Wir tragen dabei für v ein w (wie wahr) ein, wenn ein Vorlesungsbesuch stattfindet und ein f (wie falsch) sonst.

n	v	t	a	b	c	*Ergebnis*
f	f	f	f	w	w	*falsch*
f	f	w	f	w	w	*falsch*
f	w	f	w	w	w	*wahr*
f	w	w	w	w	w	*wahr*
w	f	f	w	w	f	*falsch*
w	f	w	w	w	f	*falsch*
w	w	f	w	w	w	*wahr*
w	w	w	w	f	f	*falsch*

Ein genauer Blick auf die Tabelle lässt uns erkennen, dass der gute Mann stets die Vorlesung besucht, aber nie alle drei Veranstaltungen.

Versuchen wir dies herzuleiten, indem wir die drei Bedingungen vereinfachen. Mit c) gilt sicher auch $\neg v \rightarrow \neg n$, mit a) schließen wir, dass v stets wahr sein muss. Dann wird aus dem System einfach

a) v

b) $n \rightarrow \neg t$

c) $t \rightarrow \neg n$.

Wir können aus diesem „Ausdruck“ den Folgerungspfeil eliminieren, indem wir schreiben:

$$v \wedge (\neg(n \wedge t)) \wedge (\neg(t \wedge n)) \quad = \quad v \wedge (\neg(n \wedge t))$$

Also geht er immer in die Vorlesung und wenn er nacharbeitet geht er nicht ins Tutorium und umgekehrt.

In diesem Beispiel haben wir ein System von logisch verknüpften Aussagen. Wir erhalten den Wahrheitswert der Gesamtaussage, indem wir den einzelnen Aussagevariablen Wahrheitswerte zuweisen. Deswegen nennen wir die Logik solcher Formeln *Aussagenlogik*.

Im nächsten Beispiel wollen wir die Struktur eines mathematischen Beweises genauer betrachten. Dafür führen wir zunächst *Quantoren* ein, den *Allquantor* $\forall$, gesprochen „für alle“ und den *Existenzquantor* $\exists$, gesprochen „es gibt ein“.

Beispiel 8.2.2 *Wir betrachten das Axiomensystem der Äquivalenzrelation „$\sim$ “.*

(E1) $\forall x : x \sim x$.

(E2) $\forall x, y : x \sim y \Rightarrow y \sim x$.

(E3) $\forall x, y, z : x \sim y$ *und* $y \sim z \Rightarrow x \sim z$.

Wir wollen nun den Beweis von Proposition 1.5.2, die Einteilung in Äquivalenzklassen, möglichst stark formalisieren und in eine „Berechnung“ umformen

Satz 8.2.3 $\forall x, y : (\exists u : (x \sim u \text{ und } y \sim u) \Rightarrow \forall z : (x \sim z \Leftrightarrow y \sim z))$.

Diese Aussage können wir nun nicht mehr durch Einsetzen aller möglichen Belegungen nachweisen, es ist auch gar nicht ganz klar, wo alle x, y „sich befinden“.

Statt dessen leiten wir solche Aussagen aus den vorgegebenen mittels logischer Schlussregeln ab.

Beweis. *Seien x, y beliebig aber fest vorgegeben und existiere ein u mit*

$$\begin{aligned} & x \sim u \text{ und } y \sim u && (8.1)\\ \overset{(E2)}{\Rightarrow} \quad & u \sim x \text{ und } y \sim u && (8.2)\\ \Rightarrow \quad & y \sim u \text{ und } u \sim x && (8.3)\\ \overset{(E3)}{\Rightarrow} \quad & y \sim x. && (8.4) \end{aligned}$$

Analog erhalten wir

$$x \sim y. \tag{8.5}$$

Sei nun z beliebig. Dann erhalten wir falls

$$\begin{aligned} & x \sim z && (8.6)\\ \overset{y\sim x,(E3)}{\Rightarrow} \quad & y \sim z && (8.7) \end{aligned}$$

und wiederum analog $y \sim z \Rightarrow x \sim z$. □

Zunächst einmal analysieren wir den formalen Aufbau der *Ausdrücke* und stellen fest, dass diese aufgebaut sind aus

- Quantoren (Generalisator und Partikularisator),
- Junktoren (logische Verknüpfungen wie „und“ und Folgerungspfeil),
- Variablen,
- einem zweistelligen *Prädikat* oder *Attribut*, das für jedes geordnete Paar zweier Elemente entweder gilt oder nicht gilt.

Wir haben in jedem Schritt aus dem bisher Bekannten (Axiomen, Prämissen) eine neue Aussage *gefolgert*, was wir durch den Implikationspfeil angedeutet haben. Wann dürfen wir einen solchen Pfeil schreiben? Was ist eine „richtige logische Schlussfolgerung“? Wir werden diese Schlussregeln später etwas genauer analysieren und formalisieren.

Im letzten, etwas längeren Beispiel wollen wir nun weitere auftretende Symbole diskutieren und das obige Resultat anwenden.

Beispiel 8.2.4 *Wir betrachten die Axiome der Gruppentheorie bzgl. einem* Individuenbereich *G. Seien $\circ : G \times G \to G$ und $^{-1} : G \to G$ Abbildungen mit*

(G1) $\forall x, y, z : (x \circ y) \circ z = x \circ (y \circ z)$

(G2) $\forall x : x \circ 1 = x$

(G3) $\forall x : x \circ x^{-1} = 1.$

Hier treten zusätzlich noch eine „Konstante“ 1*, eine ein*stellige Funktion x^{-1} *und eine zweistellige Funktion* $x \circ y$ *auf.*

Wir zeigen nun zunächst, dass jedes Rechtsinverse auch Linksinverses ist.

Proposition 8.2.5 $\forall x, y : (x \circ y = 1 \Rightarrow y \circ x = 1).$

Beweis. $y \circ x \overset{(G2)}{=} (y \circ x) \circ 1 \overset{(G3)}{=} (y \circ x) \circ (y \circ y^{-1}) \overset{(G1)}{=} y \circ (x \circ (y \circ y^{-1})) \overset{(G1)}{=} y \circ ((x \circ y) \circ y^{-1})) = y \circ (1 \circ y^{-1}) \overset{(G1)}{=} (y \circ 1) \circ y^{-1} \overset{(G2)}{=} y \circ y^{-1} \overset{(G3)}{=} 1.$ □

Als nächstes beweisen wir die Eindeutigkeit des Rechtsinversen.

Korollar 8.2.6 $\forall x, y : (x \circ y = 1 \Rightarrow y = x^{-1}).$

Beweis. $x^{-1} \overset{(G2)}{=} x^{-1} \circ 1 = x^{-1} \circ (x \circ y) \overset{(G1)}{=} (x^{-1} \circ x) \circ y \overset{8.2.5}{=} 1 \circ y \overset{(G3)}{=} (y \circ y^{-1}) \circ y \overset{(G1)}{=} y \circ (y^{-1} \circ y) \overset{8.2.5}{=} y \circ 1 \overset{(G2)}{=} y.$ □

Sei nun $U \subseteq G$*, eine Teilmenge von* G*, die die Gruppenaxiome erfüllt (eine Untergruppe), wobei die Funktionen Einschränkung der Funktionen von* G *sind. Insbesondere ist also die Verknüpfung zweier Elemente aus* U *wieder in* U *und ebenso das Inverse eines Elementes in* U*. Wir behaupten*

Satz 8.2.7 *Durch* $a \sim b \Leftrightarrow a \circ b^{-1} \in U$ *ist auf* G *eine Äquivalenzrelation erklärt.*

Beweis. *Wir haben* $(E1), (E2), (E3)$ *zu verifizieren.*

E1: $\forall a \in U : a \circ a^{-1} = 1 \in U \Rightarrow \forall a : a \sim a.$

E2: $a \sim b \Leftrightarrow a \circ b^{-1} \in U \Rightarrow (a \circ b^{-1})^{-1} \in U$*. Nun ist wegen* $(a \circ b^{-1}) \circ (b \circ a^{-1}) \overset{(G1)}{=} a \circ (b^{-1} \circ (b \circ a^{-1})) \overset{(G1)}{=} a \circ ((b^{-1} \circ b) \circ a^{-1}) \overset{8.2.5}{=} a \circ (1 \circ a^{-1}) \overset{(G1)}{=} (a \circ 1) \circ a^{-1}) \overset{(G2),(G3)}{=} 1$ *und Korollar 8.2.6* $b \circ a^{-1} = (a \circ b^{-1})^{-1} \in U$ *und somit* $b \sim a$*.*

E3: $a \sim b$ *und* $b \sim c \Rightarrow (a \circ b^{-1} \in U$ *und* $b \circ c^{-1} \in U) \Rightarrow (a \circ b^{-1}) \circ (b \circ c^{-1}) \overset{(G1)}{=} a \circ (b^{-1} \circ (b \circ c^{-1})) \overset{(G1)}{=} a \circ ((b^{-1} \circ b) \circ c^{-1}) \overset{(G3),(8.2.5)(G1),(G2)}{=} a \circ c^{-1} \in U.$

Mit Satz 8.2.3 schließen wir also, dass G *in* Linksnebenklassen *von* U *zerfällt.* □

Was erlaubt es uns eigentlich, Satz 8.2.3 auf die speziellere Situation von Satz 8.2.7 anzuwenden? Was heißt es, dass ein Satz gültig ist (aus der Theorie *folgt*)? Was impliziert dies für die Situation in einer bestimmten Gruppe?

Betrachten wir die multiplikative Gruppe von $\mathbb{Z}/37\mathbb{Z}$ also $(\{1, 2, \ldots, 36\}, \cdot)$ mit der Multiplikation Modulo 37 gerechnet. Dann ist $\{1, 6, 31, 36\}$ eine Untergruppe. Diese liefert eine Zerlegung in 9 Nebenklassen, die z.B. repräsentiert werden durch $\{1, 2, 3, 4, 5, 8, 9, 10, 15\}$. Die angegebene Gruppe stellt ein *Modell* für die Gruppentheorie dar. Hieraus schließen wir, dass Sätze der Gruppentheorie für das Modell gültig sind. Umgekehrt ist es auch sinnvoll zu sagen, dass eine Aussage in der Gruppentheorie gültig ist, wenn Sie in all ihren Modellen gilt.

8.3 Programm

Also werden wir im Rahmen dieser Vorlesung folgende Problematiken diskutieren:

Syntax Wie kann man formal korrekt geformte „Sätze“ formulieren. Wir werden hier die Syntax der Aussagenlogik und die Syntax der Prädikatenlogik erster Stufe kennenlernen.

Semantik In der Semantik interpretieren wir diese „Sätze“. Dabei werden wir „zutreffende Interpretationen“ als *Modell* bezeichnen.

Folgerungsbeziehung Wenn wir eine Menge von vorgegebenen Sätzen als Axiomensystem für eine Theorie bezeichnen, so folgt ein Satz aus der Theorie, wenn er in jedem Modell der Theorie gültig ist. Die Folgerungsbeziehung ist also ein semantischer Begriff.

Beweiskalküle Man gibt formale Regeln an, wie man aus syntaktisch korrekten Zeichenketten andere Zeichenketten ableiten kann. Eine solche Ableitung werden wir Beweis nennen. Dieser Beweisbegriff spielt sich also auf der rein syntaktischen Ebene ab.

Vollständigkeit: Ein Kalkül ist vollständig, wenn man alle Sätze ableiten kann, die folgen.

Entscheidbarkeit Hier geht es darum, ob man die Existenz der Ableitungen maschinell in endlicher Zeit entscheiden kann, also ob man existierende Beweise mittels eines algorithmischen Konzeptes in endlicher Zeit finden, oder in endlicher Zeit feststellen kann, dass es eine solche

Ableitung nicht gibt. Allerdings werden wir diesen Begriff nicht exakt behandeln können, da uns die Zeit fehlt, Maschinenmodelle zu diskutieren.

Kapitel 9

Syntax

Wie wir gesehen haben, erleichtert eine Formalisierung von Aussagen ihre Auswertung. Wir definieren deshalb die Sprache der *Ausdrücke*. Dafür einigen wir uns auf ein *Alphabet*, d.i. der Symbolvorrat, aus dem unsere Ausdrücke zusammengesetzt werden. Von den Zeichenketten, die aus diesen Symbolen gebildet werden können, wollen wir aber nur „wohlgeformte“ zulassen:

9.1 Die Alphabete

Definition 9.1.1 *Das* Alphabet einer Sprache erster Stufe *umfasst folgende Symbole:*

a) *die Junktoren* $\wedge$ *und* $\neg$*,*

b) *Klammern* $), ($*,*

c) *den Generalisator* $\forall$*,*

d) *das Gleichheitszeichen* $\equiv$*,*

e) *für jede Zahl* $n \in \mathbb{N}$ *höchstens abzählbar viele* n*-stellige* Funktionssymbole*. Dabei heißen die nullstelligen Funktionssymbole auch* Individuenvariablen *oder manchmal auch* Konstanten*. Zusätzlich gelte, dass es entweder keine oder abzählbar viele Individuenvariablen gibt.*

f) *Für jedes* $n \in \mathbb{N}$ *höchstens abzählbar viele* Relationssymbole *oder* Prädikatssymbole*. Die nullstelligen Relationssymbole heißen auch* Aussagenvariablen*.*

Fortan stehe Σ für die unter a)−d) genannten technischen *Zeichen und S für die unter e) − f) benannte* Symbolmenge. *Die Menge $\Sigma_S = \Sigma \cup S$ ist unser* Alphabet. *Die Menge aller endlichen Zeichenreihen* (Wörter)*, die man aus diesem Zeichenvorrat durch Aneinanderreihung gewinnen kann, nennen wir* Sprache *und bezeichnen sie mit Σ_S^*. Auch jede Teilmenge von Σ_S^* wollen wir Sprache nennen. Die Symbolmenge S bestimmt eine Sprache erster Stufe.*

Wir wollen von den Zeichen voraussetzen, dass man

a) von jedem Symbol entscheiden kann, ob es ein Funktionssymbol, ein Relationssymbol oder ein technisches Zeichen ist,

b) in den ersten beiden Fällen die Stelligkeit bestimmen kann

c) und jede Zeichenreihe eindeutig und effektiv in die Symbole zerlegen lässt, aus denen sie zusammengesetzt ist.

Wir haben oben als Beispiele $S_G = \{1,\ ^{-1}, \circ, v_0, v_1, \ldots\}$ für die Sprache der Gruppentheorie (mit Individuenvariablen 1 und v_i) und $S_E = \{\sim, v_0, v_1, \ldots\}$ für die Sprache der Äquivalenzrelationen.

Definition 9.1.2 *Das* Alphabet der Sprache der Aussagenlogik *umfasst folgende Symbole:*

a) Die Junktoren $\wedge$ und $\neg$,

b) Klammern $), ($,

c) abzählbar viele nullstellige Relationssymbole (Aussagenvariablen). Die Menge der Aussagenvariablen bezeichnen wir mit $\mathcal{A} = \{A_0, A_1, \ldots\}$.

Wir haben es also hier mit einem einfachen Spezialfall der obigen Definition zu tuen, bei dem zusätzlich der Generalisator entfallen ist (vgl. Bemerkung nach Definition 9.3.2).

9.2 Terme

Als nächstes wollen wir die einfachsten Bausteine unserer Sprache definieren, die Terme. Im Falle der Aussagenlogik ist die folgende Definition leer und deshalb nur im allgemeinen Fall sinnvoll:

Definition 9.2.1 *(induktiver Aufbau der Terme) Die* S-Terme *über der Symbolmenge* S *sind genau die Wörter in* Σ_S^**, die durch folgenden induktiven Prozess gegeben sind:*

a) Alle Individuenvariablen aus S *sind* S*-Terme.*

b) Sind die Wörter $t_1, \ldots, t_n$ S*-Terme und ist* f *ein* n*-stelliges Funktionssymbol aus* S*, so ist* $ft_1 \ldots t_n$ *ein Term.*

Die Menge der S*-Terme bezeichen wir mit* T^S*.*

Bemerkung 9.2.2 *Der erste Fall in der Definition ist im zweiten enthalten und nur aus Gründen der Übersichtlichkeit extra aufgeführt.*

Beispiel 9.2.3 *Betrachten wir das Alphabet der Gruppentheorie, so sind* $v_1, \circ v_{50} v_{13}$ *und* $\circ v_1 \circ v_2 v_3$ *Terme, nicht aber* $v_1 \circ v_2, \circ v_1 v_2 \circ v_1 \circ 1$*.*

Terme sind also stets mittels Funktionssymbolen aufgebaut. Da diese in der Aussagenlogik nicht auftreten, gibt es dort keine Terme.

9.3 Ausdrücke und Formeln

Aus diesen Termen setzen wir nun wie folgt die Ausdrücke oder Formeln zusammen.

Definition 9.3.1 *(induktive Definition)* S-Ausdrücke *oder* S-Formeln *sind genau die Wörter in* Σ_S^**, die durch folgenden induktiven Prozess gegeben sind.*

a) Für je zwei S*-Terme* t_1, t_2 *ist* $t_1 \equiv t_2$ *ein* S*-Ausdruck.*

b) Sind $t_1, \ldots, t_n$ S*-Terme und ist* R *ein* n*-stelliges Relationssymbol aus* S*, so ist* $Rt_1 \ldots t_n$ *ein* S*-Ausdruck.*

c) Ist φ *ein* S*-Ausdruck, so ist* $\neg\varphi$ *ein* S*-Ausdruck. Sind* φ *und* ψ S*-Ausdrücke, so ist* $(\varphi \wedge \psi)$ *ein* S*-Ausdruck.*

d) Ist φ *ein* S*-Ausdruck und* x *eine Individuenvariable, so ist* $\forall x\varphi$ *ein* S*-Ausdruck.*

Wir nennen $\neg\varphi$ *die* Negation von φ *und* $(\varphi \wedge \psi)$ *die* Konjunktion von φ und ψ. *Ausdrücke, die mit Regel a) oder b) gebildet werden, heißen* atomare Ausdrücke. *Ein Ausdruck* F*, der als zusammenhängende Zeichenkette in einem Ausdruck* G *auftritt, heißt* Teilausdruck. *Falls zusätzlich* $F \neq G$ *ist, so nennen wir* F *einen* echten Teilausdruck von G.

Wiederum wollen wir den Spezialfall der Aussagenlogik extra notieren. Hier sind die Aussagenvariablen die einzigen atomaren Ausdrücke.

Definition 9.3.2 *Eine* atomare Formel *(oder auch* Aussagenvariable*) hat die Form* A_i *mit* $i = 0, 1, 2, \ldots$. *Die Menge der* Formeln *(oder* Ausdrücke*) wird dann durch folgenden rekursiven Prozess definiert.*

a) Alle atomaren Formeln sind Formeln.

b) Sind F *und* G *Formeln, so sind auch* $(F \wedge G)$ *und* $\neg F$ *Formeln.*

Eine Formel F*, die als zusammenhängende Zeichenkette in einer Formel* G *auftritt heißt* Teilformel*, falls zusätzlich* $F \neq G$ *ist, so nennen wir* F *eine* echte Teilformel von G.

Wir sehen, dass wir bei der Aussagenlogik den Generalisator nicht benötigen, da nur über Individuenvariablen quantifiziert werden darf, die in der Sprache der Aussagenlogik nicht vorkommen. Ebenso entfällt das Gleichheitszeichen, da keine Terme auftreten.

Wir vereinbaren folgende abkürzenden Schreibweisen. Anstatt $\neg(\neg\varphi \wedge \neg\psi)$ schreiben wir $(\varphi \vee \psi)$ und für $\neg(\neg\psi \wedge \varphi)$ auch $(\varphi \to \psi)$ und $(\varphi \leftrightarrow \psi)$ für $((\varphi \to \psi) \wedge (\psi \to \varphi))$ jeweils für S-Ausdrücke oder Formeln φ und ψ. $(\varphi \vee \psi)$ nennen wir eine *Disjunktion* und $(\varphi \to \psi)$ eine *Implikation.* Anstatt $\neg\forall x\varphi$ schreiben wir auch $\exists x\neg\varphi$ und nennen $\exists$ den *Partikularisator.* Außerdem werden wir häufig Klammern weglassen, wenn sie für das Verständnis unwesentlich sind. Bei iterierten Konjunktionen und Disjunktionen vereinbaren wir Linksklammerung, also $\varphi \wedge \psi \wedge \chi = ((\varphi \wedge \psi) \wedge \chi)$. Man beachte, dass die dadurch entstehenden Zeichenketten im strengen Sinne keine Ausdrücke, Formeln oder Terme mehr sind. Allerdings erleichtert die abgekürzte Schreibweise häufig das Verständnis.

Im Falle quantorenfreier Ausdrücke (insbesondere in der Aussagenlogik) benutzen wir zusätzlich als Abkürzungen für Ketten von Disjunktionen oder Konjunktionen:

$$(\bigvee_{i=1}^{n} F_i) := ((\ldots((F_1 \vee F_2) \vee F_3) \vee \ldots) \vee F_n)\,,$$

$$(\bigwedge_{i=1}^{n} F_i) := ((\ldots((F_1 \wedge F_2) \wedge F_3) \wedge \ldots) \wedge F_n)\,.$$

9.4 Beispiel

Beispiel 9.4.1 (Aussagenlogik) *Betrachten wir die Formel $\neg(\neg a \wedge b)$, so hat diese die Teilformeln $(\neg a \wedge b), \neg a, a, b$. Hingegen ist $\neg(\wedge a \wedge b)\neg a$ ebensowenig eine Teilformel wie $a \wedge b$.*

Beispiel 9.4.2 (Prädikatenlogik) *Wir betrachten $S = \{x, y, z\} \cup \mathbb{N}_0 \cup \{+, \cdot, <\}$, wobei $+$ und $\cdot$ zweistellige Funktionssymbole und $<$ ein zweistelliges Relationssymbol und $\mathbb{N}_0$ die Menge der natürlichen Zahlen als Individuenvariablen ist und drei zusätzlichen Individuen x, y, z ausgezeichnet sind. Die atomaren Formeln dieser Sprache haben die Gestalt*

$$s \equiv t \text{ oder } < st$$

mit Termen s und t, wobei z.B. $+(3) \cdot (5)(7)$ ein Term ist. Hier haben wir die natürlichen Zahlen geklammert damit eine Unterscheidung z.B. zwischen (57) und (5)(7) möglich ist, was wir gefordert hatten. Folgende Beispiele sind Formeln:

$$< (0)(0), \quad \forall x \neg(< \cdot x(+y(1)) + \cdot xx \cdot yz \wedge \forall y \neg < \cdot yy + xz)$$

oder in lesbarer Schreibweise mit Infixnotation

$$0 < 0, \quad \forall x\,((x \cdot (y+1) < x \cdot x + y \cdot z) \rightarrow (\exists y\,(y \cdot y < x + z))\,.$$

Mit obigen Definitionen erhalten wir, wie bereits gesagt die Sprache der Aussagen und Formeln. Die Zugehörigkeit einer Zeichenkette zu der Sprache kann man, wegen der vorausgesetzten Eigenschaften der Symbole, leicht und effizient testen, indem man ihren Aufbau zurückverfolgt.

9.5 Induktion im Term- und Ausdruckskalkül

Der induktive Aufbau der Zeichenketten Z (Terme oder Ausdrücke), erlaubt es, Behauptungen über die Elemente von Z durch *Induktion über den Aufbau* zu beweisen. Dafür müssen wir bei den Termen beweisen, dass die Behauptung erstens für Individuenvariable wahr ist. Wenn sie für die Terme $t_1, \ldots t_n$

wahr ist und f eine n-stellige Funktion ist, so haben wir die Gültigkeit für $ft_1 \dots t_n$ zu zeigen. Bei der Induktion über den Aufbau der Ausdrücke müssen wir analog zeigen, dass die Behauptung

a) für alle Ausdrücke der Gestalt $t_1 \equiv t_2$ mit Termen t_1, t_2 wahr ist und

b) für alle Ausdrücke der Gestalt $Rt_1 \dots t_n$ für Terme $t_1, \dots, t_n$ und n-stellige Relationen R wahr ist.

c) Ist die Behauptung für die Ausdrücke φ und ψ wahr, so ist sie auch für $(\varphi \wedge \psi)$ und $\neg\varphi$ wahr.

d) gilt die Behauptung für den Ausdruck φ, so auch für $\forall x\varphi$.

Wir demonstrieren diese Vorgehensweise an dem Beweis des folgenden Satzes.

Proposition 9.5.1 *Für alle Symbolmengen S ist das* leere Wort ϵ *(die Zeichenkette der Länge 0) kein Term und kein Ausdruck.*

Beweis. Sei E die Eigenschaft, die für eine Zeichenreihe genau dann zutrifft, wenn die Zeichenreihe nicht leer ist. Dann gilt E für alle Individuenvariablen. Ist f ein n-stelliges Funktionssymbol, so ist $ft_1 \dots t_n$ nicht leer. Somit gilt E für alle Terme. Der Beweis für die Ausdrücke ist analog. □

Proposition 9.5.2 *Sei S eine Symbolmenge.*

a) Seien t, t' S-Terme sowie $\zeta \in \Sigma_S^*$. Ferner gelte $t = t'\zeta$. Dann ist ζ ist das leere Wort, maW. ein Term kann nicht echtes Anfangsstück eines Terms sein.

b) Seien α, α' S-Ausdrücke sowie $\zeta \in \Sigma_S^*$. Ferner gelte $\alpha = \alpha'\zeta$. Dann ist ζ ist das leere Wort, maW. ein Ausdruck kann nicht echtes Anfangsstück eines Ausdrucks sein.

c) Ausdrücke lassen sich effizient erkennen.

Beweis. Als Übung. □

Kapitel 10

Semantik

Bisher haben die Zeichenketten noch keine Bedeutung. Dies ändert sich, wenn wir einen Grundbereich G für die Individuenvariablen, für die Relationssymbole Relationen und für die Funktionssymbole Funktionen festlegen, also die Zeichenketten interpretieren. Dabei ist eine n-stellige Relation $\Re \subseteq G^n$ eine Teilmenge von G^n und eine n-stellige Funktion f eine Funktion $f : G^n \to G$. Bei Relationen sagen wir anstatt $(g_1, \ldots, g_n) \in \Re$ oft einfacher $\Re$ trifft auf $(g_1, \ldots, g_n)$ zu, oder $Ra_1 \ldots a_n$ ist wahr.

10.1 Interpretationen

Wir definieren also

Definition 10.1.1 *Eine* S-Interpretation *ist ein Paar* $\mathcal{G} = (G, \Im)$ *mit den folgenden Eigenschaften:*

a) G *ist eine Menge, der sog.* Grundbereich *oder* Träger *von* $\mathcal{G}$*.*

b) $\Im$ *ist eine auf* S *definierte Abbildung, die*

   (a) für jedes $n \in \mathbb{N}$ *jedem* n*-stelligen Relationssymbol* R *aus* S *eine* n*-stellige Relation* $\Im(R) \subseteq G^n$ *zuordnet und*

   (b) für jedes $n \in \mathbb{N}$ *jedem* n*-stelligen Funktionssymbol* f *aus* S *eine* n*-stellige Funktion* $\Im(f) : G^n \to G$ *zuordnet.*

Bemerkung 10.1.2 *Es gibt zwei nullstellige Relationen, nämlich* $\{\emptyset\}$ *und* $\emptyset$*. Erstere interpretieren wir als „wahr“ und letztere als „falsch“. Nullstellige*

Funktionen sind konstant, Individuenvariablen werden also durch Elemente aus dem Grundbereich interpretiert.

Den Spezialfall der Aussagenlogik erhalten wir, indem wir $G = \{\}$ wählen, da keine Individuen benötigt werden.

Definition 10.1.3 (Semantik der Aussagenlogik) *Die Elemente w, f heißen* Wahrheitswerte*, wir interpretieren w als* wahr *und f als* falsch*. Eine* Interpretation *ist eine Funktion $\Im : \mathcal{A} \to \{w, f\}$, die jeder atomaren Formel (oder Aussagenvariablen) einen Wahrheitswert (nullstellige Relation) zuordnet.*

10.2 Modell– und Folgerungsbeziehung

Wir werden im Folgenden allgemeiner den Ausdrücken in einer Interpretation Wahrheitswerte zuweisen. Dafür analysieren wir den Aufbau der Ausdrücke. Da logische Junktoren beim Aufbau der Ausdrücke eine zentrale Rolle spielen, müssen wir die Verknüpfung von Wahrheitswerten normieren. Dies soll so geschehen, dass die Normierungen sich mit dem umgangssprachlichen Begriff möglichst decken und eindeutig defniert sind.

Wir legen deshalb folgende Funktionstafeln (Wahrheitstafeln) fest, indem wir den Konjunktor als Abbildung $\wedge : \{w, f\} \times \{w, f\} \to \{w, f\}$ und den Negator als $\neg : \{w, f\} \to \{w, f\}$ festlegen:

φ	ψ	$(\varphi \wedge \psi)$
w	w	w
w	f	f
f	w	f
f	f	f

φ	$\neg\varphi$
w	f
f	w

Man rechnet nach, dass für unsere abkürzenden Schreibweisen gilt:

φ	ψ	$(\varphi \vee \psi)$	$(\varphi \to \psi)$	$(\varphi \leftrightarrow \psi)$
w	w	w	w	w
w	f	w	f	f
f	w	w	w	f
f	f	f	w	w

Bemerkung 10.2.1 *Auf Grund der Definition interpretieren wir* $\vee$ *semantisch als logisches* oder, $\wedge$ *als logisches* und, *sowie* $\neg$ *als* Negation *und* $\rightarrow$ *als* Implikation. *Wir haben uns bei der Definition der Ausdrücke auf die Junktoren* $\wedge$ *und* $\neg$ *beschränkt, um bei der Induktion über den Aufbau der Ausdrücke, Fälle zu sparen. Wir können diese Symbole weiter als Makros betrachten oder als Zeichen zulassen. Wir werden im nächsten Kapitel feststellen, dass solch ein größerer Symbolvorrat nützlich ist, um Ausdrücke, deren Wahrheitswert bei jeder Belegung übereinstimmt, zu normieren.*

Bevor wir in diesem Sinne die Modellbeziehung definieren können, erweitern wir die Interpretationen auf Terme.

Definition 10.2.2 *Ist* $\mathcal{G} = (G, \Im)$ *eine Interpretation, so setzen wir* $\Im$ *auf die Menge aller S-Terme fort, indem wir setzen*

$$\Im(ft_1 \dots t_n) := \Im(f)\Im(t_1) \dots \Im(t_n).$$

Nun können wir induktiv die Modellbeziehung definieren.

Definition 10.2.3 (Definition der Modellbeziehung) *Für alle* $\mathcal{G} = (G, \Im)$ *setzen wir:*

$$\begin{array}{lll}
\Im \models t_1 \equiv t_2 & :\Leftrightarrow & \Im(t_1) = \Im(t_2), \\
\Im \models \Re t_1 \dots t_n & :\Leftrightarrow & \Im(\Re)\Im(t_1) \dots \Im(t_n), \\
\Im \models \neg\varphi & :\Leftrightarrow & \textit{nicht } \Im \models \varphi, \\
\Im \models (\varphi \wedge \psi) & :\Leftrightarrow & \Im \models \varphi \textit{ und } \Im \models \psi, \\
\Im \models \forall x\varphi & :\Leftrightarrow & \textit{für alle } g \in G : \Im^g_x \models \varphi.
\end{array}$$

Dabei ist $\Im^g_x$ *die Abbildung, welche die Variable* x *auf* g *abbildet und ansonsten mit* $\Im$ *übereinstimmt. Gilt* $\Im \models \varphi$*, so sagen wir* $\Im$ ist Modell von φ, $\Im$ erfüllt φ *oder auch* φ gilt unter $\Im$.

Ist schließlich Φ *eine Menge von S-Ausdrücken, so sagen wir, dass* $\Im$ *ein Modell von* Φ *ist, falls* $\Im \models \varphi$ *für alle* $\varphi \in \Phi$*. Anders ausgedrückt, die Menge der Modelle von* Φ *ist der Durchschnitt aller Modellmengen von* $\varphi \in \Phi$.

Wir haben schon im Laufe der letzten Definition den festen Individuenbereich $\mathcal{G}$ nicht mehr erwähnt. Wenn wir im folgenden von Interpretationen sprechen, so gehört implizit stets ein Individuenbereich dazu.

Die letzte Definition ist offensichtlich verträglich mit der Normierung der Junktoren. Mit Hilfe des Modellbegriffs können wir nun definieren, wann ein Ausdruck aus einer Menge von Ausdrücken *folgt*.

Definition 10.2.4 *Sei Φ eine Menge von S-Ausdrücken und φ ein S-Ausdruck. Dann sagen wir φ* folgt aus *Φ und schreiben $\Phi \models \varphi$ genau dann, wenn jede Interpretation, die Modell von Φ ist, auch Modell von φ ist, oder anders gesagt, φ in allen Modellen von Φ gilt, bzw. die Menge der Modelle von Φ eine Teilmenge der Modelle von φ ist.*

Betrachten wir die Modell- und Folgerungsbeziehung wieder im Spezialfall der Aussagenlogik. Unter einer Interpretation $\Im$ wird zunächst nur jeder Aussagevariablen ein Wahrheitswert zugewiesen, wir haben also für jedes $A_i \in \mathcal{A}$ entweder $\Im \models A_i$ oder $\Im \models \neg A_i$ aber nicht beides. Auf Grund der Definition der Modellbeziehung wird nun induktiv jeder Formel ein Wahrheitswert unter der jeweiligen Interpretation zugewiesen.

Beispiel 10.2.5 *Sei $B : \mathcal{A} \to \{w, f\}$ eine Interpretation der Sprache der Aussagenlogik. Dann ist*

a) Für jede atomare Formel $A_i \in \mathcal{A}$ ist $B(A_i) = w$ oder $B(A_i) = f$.

b) $B((F \wedge G)) = \begin{cases} w & \text{falls } B(F) = B(G) = w \\ f & \text{sonst.} \end{cases}$

c) $B(\neg F) = \begin{cases} w & \text{falls } B(F) = f \\ f & \text{sonst.} \end{cases}$

Betrachten wir also z.B. die Formel

$$(A_1 \wedge \neg(A_2 \wedge A_3))$$

unter einer Interpretation B mit $B(A_1) = B(A_2) = B(A_3) = w$, so erhalten wir unter Berücksichtigung der Normierung der Junktoren

$$\begin{aligned} B((A_1 \wedge \neg(A_2 \wedge A_3))) &= B(A_1) \wedge B(\neg(A_2 \wedge A_3)) \\ &= w \wedge \neg B(A_2 \wedge A_3) \\ &= w \wedge \neg(B(A_2) \wedge B(A_3)) \\ &= w \wedge \neg(w \wedge w) \\ &= w \wedge \neg w \\ &= w \wedge f = f. \end{aligned}$$

Mit Hilfe der Folgerungsbeziehung definieren wir nun die semantischen Begriffe der *Allgemeingültigkeit* und *Erfüllbarkeit*.

Definition 10.2.6 *Ein S-Ausdruck φ heißt*

a) allgemeingültig*, wenn $\emptyset \models \varphi$, wir schreiben dafür kürzer $\models \varphi$,*

b) erfüllbar*, wenn es eine Interpretation gibt, die Modell von φ ist. Eine Menge von Ausdrücken Φ heißt* erfüllbar*, wenn es eine Interpretation gibt, die Modell aller Ausdrücke aus Φ ist.*

Auch hier betrachten wir wieder den Spezialfall der Aussagenlogik.

Beispiel 10.2.7 *Eine Formel F ist*

a) allgemeingültig oder eine Tautologie*, wenn für jede Belegung B der Variablen in F gilt $B(F) = w$,*

b) erfüllbar, wenn es eine Belegung B mit $B(F) = w$ gibt.

Zum Beispiel ist $(v_0 \wedge \neg v_0)$ nicht erfüllbar, $(v_0 \vee \neg v_0)$ eine Tautologie und $(v_0 \wedge v_1)$ erfüllbar.

Wir zeigen folgendes kleine Lemma:

Lemma 10.2.8 *$\Phi \models \varphi$ genau dann, wenn $\Phi \cup \{\neg\varphi\}$ nicht erfüllbar ist. Insbesondere ist φ genau dann allgemeingültig, wenn $\neg\varphi$ nicht erfüllbar ist.*

Beweis. $\Phi \models \varphi :\Leftrightarrow$ jede Interpretation, die Modell von Φ ist, ist auch Modell von $\varphi \Leftrightarrow$ es gibt keine Interpretation, die Modell von Φ aber nicht von φ ist $\Leftrightarrow \Phi \cup \{\neg\varphi\}$ ist nicht erfüllbar. □

Bemerkung 10.2.9 *Man beachte, dass in dem letzten Lemma eine Menge von Ausdrücken vereinigt wurde. Per definitionem muss ein Modell alle Ausdrücke gleichzeitig erfüllen, wir können, falls Φ endlich ist, also auch die Konjunktion all dieser Aussagen betrachten. Insbesondere notieren wir, dass wir uns bei dem Nachweis von Folgerungsbeziehungen auf nicht erfüllbare Aussagen oder nach Negation auf die Untersuchung von allgemeingültigen Aussagen beschränken können.*

Als Spezialfall der letzten Bemerkung notieren wir:

Proposition 10.2.10 *Seien φ und ψ S-Ausdrücke. Dann gilt $\psi \models \varphi$ genau dann, wenn $\varphi \vee \neg\psi$ allgemeingültig ist.*

Beweis. $\psi \models \varphi :\Leftrightarrow$ jedes Modell von ψ ist Modell von $\varphi \Leftrightarrow$ jede Interpretation, die nicht Modell von φ ist, kann kein Modell von ψ sein $\Leftrightarrow$ jede Interpretation ist entweder Modell von $\neg\psi$ oder Modell von $\varphi \Leftrightarrow$ jede Interpretation ist Modell von $\varphi \vee \neg\psi :\Leftrightarrow (\varphi \vee \neg\psi)$ ist allgemeingültig. □

Man beachte, dass $\psi \to \varphi$ von uns gerade als Abkürzung von $\varphi \vee \neg\psi$ definiert wurde.

In unseren (metasprachlichen) Argumentationen verwenden wir den Begriff genau dann, wenn und das Symbol „$\Leftrightarrow$“. Was meinen wir eigentlich damit, wenn wir es genau nehmen?

Definition 10.2.11 *Zwei Ausdrücke φ und ψ heißem* semantisch äquivalent*, wenn für alle Interpretationen $\Im$ von φ und ψ gilt $\Im(\varphi) = \Im(\psi)$, also $\varphi \models \psi$ und $\psi \models \varphi$. Wir schreiben dafür auch $\varphi \cong \psi$.*

Bemerkung 10.2.12 *Beachte, dass äquivalente Ausdrücke nicht unbedingt die gleichen oder isomorphe Symbolmenge haben müssen. So sind z.B. alle allgemeingültigen Ausdrücke semantisch äquivalent. Im folgenden Kapitel wollen wir die semantische Äquivalenz von aussagenlogischen Formeln eingehender untersuchen. Dort kann man sich nämlich, indem man nur „wesentliche“ Variablen betrachtet, das sind solche a, bei denen die Formeln, die durch Einsetzen von $a = w$ und $a = f$ entstehen nicht semantisch äquivalent sind, und Identifikation von f mit* 0 *und w mit* 1 *auf die Betrachtung von* booleschen Funktionen $b : \{0,1\}^n \to \{0,1\}$ *zurückziehen.*

Kapitel 11

Normalformen und boolesche Algebra

Offensichtlich ist die Relation der semantischen Äquivalenz reflexiv, symmetrisch und transitiv also eine *Äquivalenzrelation*. Durch Rechnen mit „semantischen Äquivalenzen“ wollen im Folgenden typische Repräsentanten bestimmen. Dafür wollen wir uns zunächst um die Rechenregeln kümmern wie sie von George Boole (1815–1864) in seiner „Algebra der Logik“ eingeführt wurde. Dafür betrachten wir in diesem Kapitel nur quantorenfreie Ausdrücke.

Man verifiziert, z.B. an Hand der Wahrheitstafeln, folgende Proposition.

Proposition 11.0.1 *Seien φ, ψ, ρ Ausdrücke. Bezeichnen wir mit* 1 *einen (beliebigen) allgemeingültigen S-Ausdruck und mit* 0 *einen nicht erfüllbaren Ausdruck, so gelten folgende semantische Äquivalenzen:*

a) $(\varphi \wedge \psi) \cong (\psi \wedge \varphi)$ *und* $(\varphi \vee \psi) \cong (\psi \vee \varphi)$ *(Kommutativität),*

b) $((\varphi \wedge \psi) \wedge \rho) \cong (\varphi \wedge (\psi \wedge \rho))$ *und* $((\varphi \vee \psi) \vee \rho) = (\varphi \vee (\psi \vee \rho))$ *(Assoziativität)*

c) $(\varphi \wedge (\psi \vee \rho)) \cong ((\varphi \wedge \psi) \vee (\varphi \wedge \rho))$ *und* $(\varphi \vee (\psi \wedge \rho)) \cong ((\varphi \vee \psi) \wedge (\varphi \vee \rho))$ *(Distributivität),*

d) $(\varphi \wedge \neg\varphi) \cong 0$, $(\varphi \vee \neg\varphi) \cong 1$, $(\varphi \wedge 1) \cong \varphi$, $(\varphi \vee 0) \cong \varphi$.

Nehmen wir diese Beobachtungen als Axiome eines Rechenbereichs.

11.1 Die boolesche Algebra

Definition 11.1.1 *Eine* boolesche Algebra $(A, 0, 1, \wedge, \vee, \neg)$ *ist eine Menge A mit zwei ausgezeichneten Symbolen 0 und 1, zwei Verknüpfungen $\wedge : A \times A \to A$ und $\vee : A \times A \to A$, sowie einer Abbildung $\neg : A \to A$, so dass folgende Axiome gelten:*

B1: $\forall a, b \in A : (a \wedge b) = (b \wedge a)$ *und* $(a \vee b) = (b \vee a)$ *(Kommutativität),*

B2: $\forall a, b, c \in A : ((a \wedge b) \wedge c) = (a \wedge (b \wedge c))$ *und* $((a \vee b) \vee c) = (a \vee (b \vee c))$ *(Assoziativität)*

B3: $\forall a, b, c \in A : (a \wedge (b \vee c)) = ((a \wedge b) \vee (a \wedge c))$ *und* $(a \vee (b \wedge c)) = ((a \vee b) \wedge (a \vee c))$ *(Distributivität),*

B4: $\forall a \in A : (a \wedge \neg a) = 0$, $(a \vee \neg a) = 1$, $(a \wedge 1) = a$, $(a \vee 0) = a$.

Bemerkung 11.1.2 *Offensichtlich können wir mit der Syntax der Aussagenlogik angewendet auf die Variablen einer booleschen Algebra* Formeln *definieren, denen in der booleschen Algebra ein eindeutiges Element zugeordnet ist. Wir sprechen dann auch von einer Formel der booleschen Algebra.*

Satz 11.1.3 *Sei $(A, 0, 1, \wedge, \vee, \neg)$ eine boolesche Agebra.*

a) *Dann gilt:* $(a \wedge a) = (a \vee a) = a$, $a \wedge 0 = 0$ *und* $a \vee 1 = 1$.

b) *Ist* $(b \wedge a) = 0$ *und* $(b \vee a) = 1$, *so ist* $b = \neg a$,

c) $\neg\neg a = a$,

d) $\neg(a \wedge b) = \neg a \vee \neg b$ *und* $\neg(a \vee b) = \neg a \wedge \neg b$ *(De Morgan).*

Beweis.

a) $(a \wedge a) = ((a \wedge a) \vee 0) = ((a \wedge a) \vee (a \wedge \neg a)) \overset{B3}{=} (a \wedge (a \vee \neg a)) = (a \wedge 1) = a$. Analog berechnet man $(a \vee a)$. Weiterhin ist $(a \wedge 0) = (a \wedge (a \wedge \neg a)) = ((a \wedge a) \wedge \neg a) = (a \wedge \neg a) = 0$ und ähnlich $a \vee 1 = a$.

b) $b = (b \vee (a \wedge \neg a)) = ((b \vee a) \wedge (b \vee \neg a)) = (b \vee \neg a) = ((\neg a \vee a) \wedge (b \vee \neg a)) = (\neg a \vee (a \wedge b)) = \neg a$.

c) selber (benutze b))

d) Auf Grund des bereits Gezeigten folgt die erste Aussage, wenn wir beweisen, dass $((\neg a \vee \neg b) \wedge (a \wedge b)) = 0$ und $((\neg a \vee \neg b) \vee (a \wedge b)) = 1$. $((\neg a \vee \neg b) \wedge (a \wedge b)) = (\neg a \wedge a \wedge b) \vee (\neg b \wedge a \wedge b) = (0 \wedge b) \vee (0 \wedge a) = 0$. $((\neg a \vee \neg b) \vee (a \wedge b)) = (\neg a \vee a \vee b) \wedge (\neg b \vee a \vee b) = 1$. Rest analog.

□

Bemerkung 11.1.4 *Die Axiome* $B1, B2$ *zusammen mit dem* Axiom der Adjunktivität

$$\textbf{V3:} \forall a, b \in A : (a \wedge (a \vee b)) = a = (a \vee (a \wedge b))$$

bilden die Axiome der Verbandstheorie. *Die boolesche Algebra definiert einen Verband, da auf Grund der Distributivität zunächst*

$$a \wedge (a \vee b) = (a \wedge a) \vee (a \wedge b) = a \vee (a \wedge b)$$

und ferner nach de Morgan

$$\neg(a \wedge (a \vee b)) = \neg a \vee \neg(a \vee b) = \neg a \vee (\neg a \wedge \neg b) = \neg a \wedge (\neg(a \wedge b))$$

und somit

$$(a \vee \neg(a \wedge (a \vee b)) = a \vee (\neg a \vee \neg(a \vee b)) = 1 \vee \neg(a \vee b) = 1$$

sowie

$$(a \wedge \neg(a \wedge (a \vee b)) = a \wedge (\neg a \wedge \neg(a \wedge b)) = 0 \wedge \neg(a \wedge b) = 0.$$

Also folgt aus Satz 11.1.3 b) auch $(a \wedge (a \vee b)) = a$.

11.2 Normalformen

Ziel dieses Abschnitts ist es, standardisierte Vertreter semantisch äquivalenter, quantorenfreier Ausdrücke zu bestimmen. (In diesem Kapitel sind $\wedge$ und $\vee$ gleichberechtigt. Die Definition der Ausdrücke beinhaltet dann auch den Übergang von den Ausdrücken φ und ψ zu $\varphi \vee \psi$.) Vor Allem interessieren uns dabei Formeln der Aussagenlogik. Da sich einige Sachverhalte aber auch auf quantorenfreie Ausdrücke übertragen lassen, indem wir die atomaren Teilaussagen wie Aussagevariablen behandeln, werden wir allgemeiner vorgehen.

Wir bringen die Ausdrücke in Normalform.

Definition 11.2.1 *Sei φ ein quantorenfreier Ausdruck. Dann sagen wir φ hat* Negationsnormalform*, wenn das Symbol $\neg$ nur vor atomaren Ausdrücken steht, also nie vor einer Klammer. Einen atomaren Ausdruck mit oder ohne Negation nennen wir* Literal*. Also ist ein quantorenfreier Ausdruck in Negationsnormalform, wenn er nur aus Literalen und den Symbolen $\vee, \wedge, ()$ besteht.*

Ein quantorenfreier Ausdruck (in Negationsnormalform) heißt in konjunktiver Normalform *oder in* KNF, CNF*, wenn er eine Konjunktion von Disjunktionen von Literalen ist.*

Ein quantorenfreier Ausdruck in Negationsnormalform heißt in disjunktiver Normalform *oder in* DNF*, wenn er eine Disjunktion von Konjunktionen von Literalen ist.*

Beispiel 11.2.2 *$(\neg a \vee b) \wedge (\neg c \vee b)$ ist in KNF, $(\neg a \wedge \neg c) \vee b$ ist in DNF.*

Anstelle die Existenz semantisch äquivalenter Formeln mit Hilfe der Definition zu beweisen, nutzen wir aus, dass die Äquivalenzklassen der semantischen Äquivalenz eine boolesche Algebra bilden.

Satz 11.2.3 *Für jeden quantorenfreien Ausdruck φ gibt es einen Ausdruck φ^N in Negationsnormalform, einen Ausdruck φ^K in KNF und einen Ausdruck φ^D in DNF, so dass in jeder booleschen Algebra, deren Grundmenge die atomaren Ausdrücke von φ enthält, $\varphi = \varphi^N = \varphi^K = \varphi^D$ gilt.*

Beweis. Wir führen Induktion über den Aufbau von φ, ist φ atomar, so ist nichts zu zeigen. Ist nun $\varphi = (\varphi_1 \vee \varphi_2)$, so gibt es nach Induktionsvoraussetzung φ_1^N, φ_1^K und φ_1^D für φ_1 und φ_2^N, φ_2^K und φ_2^D für φ_2. Wir setzen $\varphi^N = (\varphi_1^N \vee \varphi_2^N)$ und $\varphi^D = (\varphi_1^D \vee \varphi_2^D)$. Für φ^K müssen wir noch das Distributivgesetz anwenden. Bezeichnen dabei C_i die Disjunktionen, aus deren Konjunktion φ_1^K nach Induktionsvoraussetzung besteht und D_j analog die von φ_2^K:

$$\varphi_1^K = \bigwedge_{i=1}^{k} C_i,\ \varphi_2^K = \bigwedge_{i=1}^{l} D_j,\ \varphi^K = \bigwedge_{i=1}^{k} \bigwedge_{j=1}^{l} (C_i \vee D_j).$$

Der Fall $\varphi = (\varphi_1 \wedge \varphi_2)$ läuft analog. Sei also $\varphi = \neg\varphi_1$. Ist φ_1 atomar, so ist nichts zu zeigen. Ist $\varphi_1 = \neg\varphi_2$ so folgt mit Satz 11.1.3 $\varphi = \varphi_2$ und nach Induktionsvoraussetzung gibt es Normalformen für φ. Schließlich bleibt der Fall, dass $\varphi_1 = (\varphi_3 \vee \varphi_4)$ oder $\varphi_1 = (\varphi_3 \wedge \varphi_4)$. Dann folgt aber mit den de Morganschen Regeln $\varphi = (\neg\varphi_3 \wedge \neg\varphi_4)$ bzw. $\varphi = (\neg\varphi_3 \vee \neg\varphi_4)$. Da es nach Induktionsvoraussetzung Normalformenen für $\neg\varphi_3$ und $\neg\varphi_4$ gibt, folgt die Behauptung wie oben. □

Bemerkung 11.2.4 *Man beachte, dass der Beweis der letzten Aussage konstruktiv ist. Insbesondere können wir durch eine rekursive Prozedur jede Formel in KNF bzw. DNF bringen.*

Beispiel 11.2.5 $F = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3)$ *ist in KNF. Wir berechnen*

$$F = (x_1 \vee ((x_2 \vee x_3) \wedge (\neg x_2 \vee x_3)) = (x_1 \vee ((x_2 \wedge \neg x_2) \vee x_3)) = (x_1 \vee x_3)$$

ist sowohl in KNF als auch in DNF.

Wie wir am letzten Beispiel gesehen haben, ist unser Ziel, eine eindeutige Darstellung zu definieren, noch nicht erreicht. Wir wollen deshalb im Folgenden die Formen weiter spezialisieren, ausgehend von folgender Idee. Bei einem gegebenen quantorenfreien Ausdruck φ auf den atomaren Ausdrücken $x_1, \ldots, x_n$ kann man die 2^n möglichen Wahrheitswertbelegungen $B : \{x_1, \ldots, x_n\} \to \{w, f\}$ der Ausdrücke (unabhängig davon, ob Sie unter einer Interpretation $\Im$ tatsächlich angenommen wird,) einteilen in solche mit $B(\varphi) = w$ und solche mit $B(\varphi) = f$. Offensichtlich ist es ein Leichtes, eine Konjunktion der atomaren Ausdrücke hinzuschreiben, die bei genau einer Wahrheitswertbelegung erfüllt ist. Bilden wir die Disjunktion über alle diese Konjunktionen zu erfüllenden Belegungen, so erhalten wir eine Darstellung von φ in DNF. Verfahren wir spiegelverkehrt mit den nicht erfüllenden, so erhalten wir eine kanonische Darstellung in KNF.

Definition 11.2.6 *Sei φ ein quantorenfreier Ausdruck mit atomaren Ausdrücken $x_1, \ldots, x_n$ und B eine Wahrheitswertbelegung. Mit P_B bezeichnen wir dann die Konjunktion*

$$\bigwedge_{i=1}^{n} L_i^B, \qquad L_i^B = \begin{cases} x_i & \text{falls } B(x_i) = w \\ \neg x_i & \text{falls } B(x_i) = f. \end{cases}$$

Falls $B(\varphi) = w$ ist, so nennen wir P_B einen Minterm*. Mit der De Morganschen Regel können wir $\neg P_B$ zu einer Disjunktion m_B umformen. Ist $B(\varphi) = f$, so nennen wir m_B einen* Maxterm*.*

Eine boolesche Formel F in DNF (KNF) heißt in vollständiger DNF *(bzw.* KNF*), wenn jede Variable in jeder Konjunktion (bzw. Disjunktion) genau einmal vorkommt.*

Bemerkung 11.2.7 *Offensichtlich ist die Formel*

$$\bigvee_{\{B|B(F)=w\}} P_B$$

in der Aussagenlogik semantisch äquivalent zu F und in vollständiger DNF, gleiches gilt für die Formel

$$\bigwedge_{\{B|B(F)=f\}} m_B,$$

die in vollständiger KNF ist.

Im Folgenden wollen wir zeigen, dass vollständige Normalformen auch in jeder booleschen Algebra existieren und eindeutig sind. Unsere bisherige Argumentation behandelte nur boolesche Algebren, die durch semantische Äquivalenz auf den atomaren Teilausdrücken quantorenfreier Ausdrücke definiert sind. Wir werden nun zeigen, dass jede boolesche Algebra sich als Struktur semantischer Äquivalenz von Formeln der Aussagenlogik deuten lässt.

Satz 11.2.8 *Ist $(A, 1, 0, \wedge, \vee, \neg)$ eine boolesche Algebra und F eine Formel darin, so gibt es eine Formel F^{K^*} in vollständiger KNF, bzw. F^{D^*} in vollständiger DNF mit $F = F^{K^*} = F^{D^*}$.*

Beweis. Wegen der De Morganschen Regeln können wir den Fall DNF durch Negation auf KNF zurückführen, es genügt also, letzteren Fall zu betrachten

Wegen Satz 11.2.3 können wir ferner annehmen, dass F in KNF gegeben ist. Ferner können wir annehmen, dass in jeder Disjunktion jede Variable höchstens einmal vorkommt, da ein doppeltes Vorkommen entweder redundant ist, weil sie entweder nur positiv oder nur negiert vorkommt (Assoziativität, Kommutativität und Satz 11.1.3 a), oder die Disjunktion zu 1 macht. Im letzteren Fall können wir die Disjunktion streichen (Definition 11.1.1 Axiom d). Sei also $F = \bigwedge_{i=1}^{k} C_i$ und die KNF so gewählt, dass jede Variable in jeder Disjunktion höchstens einmal vorkommt und die Potenzialfunktion

$$\sum_{i=1}^{k} \left(2^{n-|C_i|} - 1\right)$$

möglichst klein ist. Angenommen es gäbe nun eine Disjunktion C_{i_0}, in der Variable x_j nicht vorkommt. Dann ist $C_{i_0} = (C_{i_0} \wedge 1) = (C_{i_0} \wedge (x_j \vee \neg x_j)) = ((C_{i_0} \vee x_j) \wedge (C_{i_0} \vee \neg x_j))$ und somit

$$F = (\bigwedge_{i \neq i_0}^{k} C_i \wedge (C_{i_0} \vee x_j) \wedge (C_{i_0} \vee \neg x_j)).$$

Für diese Darstellung erhalten wir aber als Potenzial

$$\sum_{i\neq i_0}^{k}\left(2^{n-|C_i|}-1\right)+2(2^{n-|C_{i_0}|-1}-1)=\left(\sum_{i=1}^{k}\left(2^{n-|C_i|}-1\right)\right)-1$$

im Widerspruch zur Wahl der KNF. □

Bemerkung 11.2.9 *Auch der Beweis des letzten Satzes ist algorithmisch. Insbesondere können wir jede Formel in KNF algorithmisch in vollständige KNF bringen.*

Nun können wir beweisen, dass Gleichheit in der booleschen Algebra und semantische Äquivalenz in der Aussagenlogik identisch sind.

Korollar 11.2.10 *Sind F und G zwei aussagenlogische Formeln und ist $F \cong G$, so ist in jeder Booleschen Algebra $F = G$.*

Beweis. Wir können annehmen, dass F und G die gleichen Variablen haben, ansonsten bilden wir Konjunktionen mit Tautologien $(y \vee \neg y)$. Bringen wir F und G in vollständige KNF F^{K^*} bzw. G^{K^*}, so müssen diese übereinstimmen, da die Wahrheitstafeln von F und G übereinstimmen. Also haben wir in jeder booleschen Algebra $F = F^{K^*} = G^{K^*} = G$. □

Bemerkung 11.2.11 *In der letzten Aussage müssen wir uns auf die Aussagenlogik beschränken. Bei beliebigen quantorenfreien Ausdrücken ist die Aussage des Korollars falsch. Betrachten wir hierzu den Ausdruck*

$$(f \equiv g) \wedge \neg(g \equiv h) \wedge (f \equiv h).$$

Dieser Ausdruck ist nicht erfüllbar, aber nicht die Null.

11.3 Primimplikanten und Primklauseln

In diesem Kapitel werden wir den Resolutionskalkül vorstellen, seine Korrektheit beweisen, und zeigen dass dieser Kalkül vollständig ist.

Definition 11.3.1 *Sind F und G aussagenlogische Formeln mit $F \models G$ und F zusätzlich eine Konjunktion von Literalen $\bigwedge_{i\in I} l_i$, so nennen wir F einen* Implikanten *von G. Ein Implikant heißt* Primimplikant*, wenn er inklusionsminimal ist, d.h. für alle $j \in I$ ist $\bigwedge_{\substack{i\in I\\ i\neq j}} l_i$ kein Implikant von G. Ist G eine Disjunktion von Literalen, so nennen wir G eine* semantische Klausel für F*. Auch hier nennen wir inklusionsminimale Vertreter* Primklausel*.*

Was sind nun die Primklauseln einer nicht-erfüllbaren Formel? Offensichtlich gibt es genau eine, nämlich die leere Klausel. Ebenso sind die Primimplikanten einer allgemeingültigen Formel leer. Wenn wir also ein Verfahren zur Berechnung von Primimplikanten oder Primklauseln angeben können, so haben wir sowohl die Vollständigkeit als auch die Entscheidbarkeit der Aussagenlogik bewiesen (vgl. Bemerkung 10.2.9).

Satz 11.3.2 (Resolution) *Sei F eine aussagenlogische Formel und A_j ein Literal. Sind $C \vee A_j$ und $C \vee \neg A_j$ semantische Klauseln der Formel F, so ist auch C eine semantische Klausel von F.*

Beweis. Nach Proposition 10.2.10 gilt $F \models C \vee A_j$ genau dann, wenn $\neg F \vee (C \vee A_j)$ allgemeingültig ist. Analog ist nach Voraussetzung $(\neg F \vee (C \vee \neg A_j))$ allgemeingültig und somit auch $((\neg F \vee (C \vee A_j)) \wedge (\neg F \vee (C \vee \neg A_j))) = ((\neg F \vee C) \vee (A \wedge \neg A_j)) = ((\neg F \vee C) \vee 0)$. Dies bedeutet aber wiederum nach Proposition 10.2.10 $F \models C$. □

Offensichtlich ist nun auch mit einer semantischen Klausel C von F, die die Variable A_j nicht enthält auch $C \vee A_j$ und $C \vee \neg A_j$ eine semantische Klausel.

Wir folgern hieraus:

Satz 11.3.3 *Eine semantische Klausel C von F ist Primklausel von F genau dann, wenn es keine semantische Klausel C' von F gibt, die sich von C in genau einem Literal unterscheidet, d.h. $C = (D \vee A_j)$, $C' = (D \vee \neg A_j)$ mit einem Literal A_j und einer Disjunktion D.*

Beweis. Ist C eine Primklausel, so kann es wegen Satz 11.3.2 eine semantische Klausel wie C' nicht geben. Ist hingegen C nicht prim, so gibt es eine echte Teilmenge der Literale, die Klausel ist, und hieraus kann man C' konstruieren. □

Die folgenden beiden Sätze benutzen wir nun als Abkürzungen $PC(F)$ für die Menge der Primklauseln und $PI(F)$ für die Menge der Primimplikanten.

Satz 11.3.4 *Sei F eine Formel. Dann gilt*

$$F \cong \bigwedge_{C \in PC(F)} C.$$

Beweis. Wie eben schließen wir zunächst, dass $F \models \bigwedge_{C \in PC(F)} C$. Angenommen nun $\bigwedge_{C \in PC(F)} C \not\models F$. Dann gibt es eine Belegung B der Variablen, so dass $(\bigwedge_{C \in PC(F)} C)(B) = w$ und $F(B) = f$. Der zugehörige Maxterm m_B muss nun eine Primklausel enthalten, die aber die Konjunktion verletzte. Widerspruch. □

Völlig analog erhalten wir das Gegenstück.

Satz 11.3.5 *Sei F eine Formel. Dann gilt*

$$F \cong \bigvee_{C \in PI(F)} C.$$

Wir können nun aus dem bisher Gezeigten die Korrektheit, Vollständigkeit und Entscheidbarkeit des Resolutionskalküls in der Aussagenlogik schließen. Wir hatten festgestellt, dass wir Folgendes algorithmisch leisten können:

- Wir können jede aussagenlogische Formel F in KNF bringen.
- Wir können jede Formel in vollständige KNF bringen und die Menge aller Maxterme hinschreiben.
- Ist $\mathcal{C}$ eine Menge von Disjunktionen, C eine Disjunktion, A_j ein Literal mit $C \vee A_j \in \mathcal{C}$ und $C \vee \neg A_j \in \mathcal{C}$, so können wir von $\mathcal{C}$ übergehen zu $\mathcal{C} \cup \{C\}$.

Satz 11.3.6 *Dieser aussagenlogische „Kalkül“ ist korrekt und „vollständig“, d.h.*

a) Jede Disjunktion in $\mathcal{C}$, die aus einer Formel F in KNF hergeleitet werden kann, ist eine semantische Klausel von F.

b) $F \models G$ genau dann, wenn man aus $(\neg G \wedge F)$ mit dem obigen Verfahren eine Menge von Disjunktionen produzieren, welche die leere Disjunktion enthält.

Beweis. Für die erste Aussage betrachten wir eine solche Disjunktion G und führen Induktion über die Anzahl der Resolutionsschritte. Ist G eine Disjunktion in F, so ist G eine semantische Klausel von F, da in jedem Modell von F alle Disjunktionen von F erfüllt sein müssen. Ist nun G aus F abgeleitet worden, so ist G im letzten Ableitungsschritt durch Resolution entstanden

aus $C_1 := (G \vee A_j)$ und $C_2 := (G \vee \neg A_j)$, wobei diese beiden Disjunktionen aus F hergeleitet sind. Da in jedem Resolutionsschritt die Länge einer Disjunktion echt kleiner wird, sind C_1 und C_2 in weniger Schritten abgeleitet worden, also nach Induktionsvoraussetzung semantische Klauseln von F. Also ist nach Satz 11.3.2 auch G eine semantische Klausel von F.

Gelte nun $F \models G$. Nach Proposition 10.2.10 ist dann $(G \vee \neg F) = \neg(\neg G \wedge F)$ eine Tautologie also $(\neg G \wedge F)$ nicht erfüllbar. Die leere Disjunktion ist also eine Klausel dieser Formel, nämlich die einzige Primklausel. Wir können nun induktiv alle Klauseln mit $0 \leq k \leq n$ Variablen berechnen. Nach Satz 11.3.3 berechnen wir dann auch die leere Klausel, haben also die Implikation bewiesen. □

Also kann man mit dem Kalkül nur korrekte Schlüsse ziehen und alle Folgerungsbeziehungen beweisen.

Literaturverzeichnis

[1] M. Aigner, *Diskrete Mathematik*, Vieweg, vierte ed., 2001.

[2] A. Brandstädt, *Graphen und Algorithmen*, Teubner, 1994.

[3] R. Diestel, *Graphentheorie*, Springer, second ed., 2000.

[4] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics: a foundation for computer science*, Addison-Wesley, second ed., 1994.

[5] R. P. Grimaldi, *Discrete and Combinatorial Mathematics*, Addison-Wesley, 1994.

[6] F. Harary, *Graph Theory*, Addison Wesley, 1972.

[7] H. Hermes, *Einführung in die mathematische Logik*, Teubner, vierte ed., 1972.

[8] T. Ihringer, *Diskrete Mathematik*, Heldermann Verlag, korrigierte und erweiterte ed., 2002.

[9] L. Lovász, *Combinatorial problems and exercises*, North-Holland, second ed., 1993.

[10] J. Matoušek and J. Nešetril, *Invitation to Discrete Mathematics*, Oxford University Press, 1998.

[11] ——, *Diskrete Mathematik - Eine Entdeckungsreise*, Springer, 2002.

[12] B. Russel, *A critical exposition of the philosophy of Leibniz*, Cambridge University Press, Cambridge, 1900.

[13] U. Schöning, *Logik für Informatiker*, BI, 1989.

[14] J. Truss, *Discrete Mathematics for Computer Scientists*, Addison-Wesley, 1991.

[15] J. van Lint and R. Wilson, *A Course in Combinatorics*, Cambridge University Press, 1992.